

Министерство образования и науки Российской Федерации

**ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)**

**Кафедра компьютерных систем в управлении и
проектировании (КСУП)**

Е.Ф. Жигалова

ДИСКРЕТНАЯ МАТЕМАТИКА

**Учебное методическое пособие для проведения практических
занятий и лабораторных работ**

ТОМСК – 2014

Рецензент:

доктор технических наук - Погребной Владимир Кириллович.,
профессор кафедры "Информатика и проектирование систем", ТПУ

Жигалова Е.Ф. Дискретная математика. Учебное методическое пособие для проведения практических занятий и лабораторных работ. – Томск: 2014г.

В пособии на общей теоретической основе изложены основные понятия и определения теории графов и переключательных функций в иллюстрациях и примерах. Рассмотрены основные постановки задач и методы их решения. Особое внимание уделено вопросам построения алгоритмов приближённого решения оптимизационных задач.

Пособие рассчитано на студентов технических университетов, изучающих курсы дискретной математики.

© Жигалова Елена Фёдоровна 2014

© Томский государственный университет
систем управления и радиоэлектроники

ОГЛАВЛЕНИЕ

Томск – 2014.....	1
ТЕМА 1. Основные понятия и определения теории графов.....	7
1.1. ОПРЕДЕЛЕНИЕ ГРАФА.....	7
В общем случае множество U можно представить в виде.....	8
1.2. Классы графов.....	10
1.3. СПОСОБЫ ЗАДАНИЯ ГРАФОВ.....	12
1.4. Числовые характеристики вершин графа.....	15
1.5. МАРШРУТЫ, ЦЕПИ И ЦИКЛЫ.....	15
1.6. ОПРЕДЕЛЕНИЕ ЧИСЛА МАРШРУТОВ ДЛИНЫ "L" НА ГРАФЕ.....	18
Тема 2. Операции на графах.....	22
2.1. Удаление вершины.....	22
2.2. Удаление ребра.....	22
2.3. Замыкание (отождествление) вершин.....	23
2.4. Стягивание вершин графа по ребру.....	24
2.5. Симметрическая разность графов.....	25
2.6. Пересечение графов.....	25
2.7. Объединение графов.....	25
ТЕМА 3. Части графа.....	28
3.1. ПОДГРАФ.....	28
3.2. Частичный граф.....	29
Тема 4. Взвешенные графы. Метрика графа.....	32
4.1. Основные понятия и определения.....	32
4.2. Способ нахождения матрицы метрики для графа общего вида	34
4.3. Алгоритм построения матрицы метрики графа.....	35
Если такие элементы имеются, то переходим к шагу 4; в противном случае – к шагу 7.	36
2. Возможна ли ситуация, когда $R_k = R_{k+1}$ при $k = 1$?.....	37
3. Для графа, заданного матрицей смежности (рисунок 4.2), определить его радиус и диаметр.....	37
Тема 5. Структурный анализ графов.....	39
5.1. Связность графа.....	39
5.2. Скелет графа.....	39
5.3. Покрывающие множества вершин и рёбер графа G	40
5.4. Максимальные полные и максимальные пустые подграфы.....	40
5.5. Клика графа.....	40

<u>5.6. Алгоритм нахождения всех максимальных пустых и всех максимальных полных подграфов (максимальных клик) в графе общего вида.....</u>	<u>41</u>
<u>Тема 6. Раскраска графов.....</u>	<u>47</u>
<u>6.1. Правильная раскраска. Хроматическое число</u>	<u>47</u>
<u>6.2. Алгоритм определения хроматического числа $\chi(G)$ с использованием метода Дж. Магу.....</u>	<u>48</u>
3. Сколько существует правильных раскрасок вершин куба, представленного на рисунке 6.1, в минимальное число цветов?.....	52
4. Разработать алгоритм правильной раскраски с минимальным количеством цветов для графа, представленного на рисунке 6.2, отличный от алгоритма, основанного на методе Магу,	52
.....	53
<u>Тема 7. Маршруты специального вида.....</u>	<u>53</u>
<u>7.1. Алгоритм Флери построения эйлерова цикла.....</u>	<u>54</u>
<u>7.2. Обходы графа вида гамильтоновой цепи, гамильтонова цикла, гамильтонова пути и контура.</u>	<u>55</u>
<u>7.3. Пример решения задачи коммивояжера с помощью алгоритма Литтла.....</u>	<u>61</u>
<u> А.....</u>	<u>62</u>
<u> А.....</u>	<u>62</u>
<u> А.....</u>	<u>64</u>
<u> А.....</u>	<u>65</u>
<u> А.....</u>	<u>65</u>
<u> А.....</u>	<u>65</u>
<u> А.....</u>	<u>66</u>
<u> А.....</u>	<u>67</u>
<u> А.....</u>	<u>67</u>
1. Определить имеют ли пятиугольник и пятигранник-пирамида с петлями в некоторых вершинах эйлеров цикл (эйлерову цепь)? ..	68
2. Имеют ли графы на рисунке 7.2 гамильтоновы циклы, цепи?.....	68
<u>Тема 8. Двудольные графы</u>	<u>70</u>
<u>8.1. Основные положения.....</u>	<u>70</u>
Очевидно, что граф является бихроматическим тогда и только тогда, когда он не содержит петель, а его скелет есть двудольный граф.....	71
<u>8.2. Венгерский алгоритм нахождения максимального паросочетания в двудольном графе.....</u>	<u>71</u>
Проиллюстрируем венгерский алгоритм на следующем примере.	72
Ясно, что при наличии у графа L с заданным паросочетанием K хотя бы одной тонкой чередующейся цепи Q можно вместо K построить новое паросочетание K' , содержащее на одно ребро больше. Для этого надо все слабые ребра цепи Q сделать сильными, а сильные – слабыми, не трогая рёбер вне Q . Иначе говоря, надо удалить из множества V все те рёбра, которые принадлежат цепи Q , и к остатку добавить рёбра Q , принадлежащие UV . В свою очередь, если у графа L с паросочетанием K' опять есть	

тонкая чередующаяся цепь, то аналогично предыдущему можно получить паросочетание K'' , имеющее уже на два ребра больше исходного.....	73
В нашем конкретном примере всё решение выглядит следующим образом.	74
2. Вычислить по формуле Кёнига-Холла число $\pi(L)$ для графов $K_1=(V_1, E_1)$ и $K_2=(V_2, E_2)$, представленных списком рёбер и вершин:.....	78
$V_1=\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$; $V_2=\{1, 2, 3, 4, 5, 6, 7\}$;.....	78
$E_1 =\{(1,8), (1,9), (2,8), (2,9), (2,12), (3,11), (4,8), (4,10), (5,11), (4, 12)\}$;	78
Тема 9. Компоненты связности.....	79
9.1. Основные определения.....	79
9.2. Способ определения числа компонент связности.....	80
9.3. Алгоритм вычисления числа компонент связности графа	83
Тема 10. Оптимальные потоки в транспортных/информационных сетях ..	85
10.1. СЕТЬ (транспортная, информационная).....	85
10.2. Поток в сети.....	85
10.3. Задача о максимальном потоке.....	86
10.4. Алгоритм Форда — Фалкерсона.....	88
10.5. Пример решения задачи о максимальном потоке с помощью алгоритма Форда-Фалкерсона.....	89
Раздел 2. Переключательные функции и их разложения.....	99
2.1. Переключательные функции и способы задания.....	99
2.2. Булевы функции (бф).....	107
2.3. Аналитическое представление булевых функций	109
2.3.1. Булева алгебра функций и эквивалентные преобразования в ней	114
2.3.2. Представление переключательных функций в виде многочленов.....	117
2.4. Функционально полные системы	119
2.5. Примеры	122
2.6. ПЕРЕЧЕНЬ ЗАДАНИЙ НА ЛАБОРАТОРНЫЕ РАБОТЫ.....	124
ти полученного решения.....	125
Лабораторная работа №4.....	125
4. Провести анализ полученных решений.....	125
1. Обходы графа. Изучить содержательную постановку и математическую модель задачи коммивояжёра;.....	125
2. Разработать алгоритм для вычисления числа рёбер в максимальном паросочетании в заданном двудольном графе с помощью теоремы Кёнига-Холла.	126
Требования к оформлению отчетов по лабораторным работам.....	126
2. Отчет должен содержать:	126
Список литературы.....	127

ТЕМА 1. Основные понятия и определения теории графов

1.1. ОПРЕДЕЛЕНИЕ ГРАФА

Теория графов оперирует формальными моделями объектов, имеет дело со свойствами самих графов независимо от того, какова природа объектов, описываемых графами. Использование аппарата теории графов для разработки алгоритмов конструкторского и технологического проектирования схем ЭВА приводит к повышению эффективности и качества создаваемых объектов.

Понятие графа опирается на понятие множества. Графом можно назвать объект, состоящий из двух множеств: множества точек и множества линий, которые соединяют между собой все или часть этих точек.

Множество точек графа обозначается $X = \{x_1, x_2, \dots, x_n\}$, и называется множеством вершин. Суммарное число n всех вершин графа называется мощностью множества X данного графа и обозначается $|X| = n$.

Множество линий, соединяющих любые пары вершин (x_i, x_j) , принадлежащих множеству X , называется множеством рёбер или дуг (если линии имеют направление) и обозначается:

$U = \{u_1, u_2, \dots, u_m\}$, где u_i - ребро (дуга) графа.

Суммарное число m всех рёбер графа называется мощностью множества рёбер графа и обозначается $|U| = m$.

Таким образом, графом можно считать математический объект, который обозначается $G = (X, U)$ и состоит из множества вершин X и множества ребер или дуг U , находящихся между собой в некотором отношении.

В общем случае множество U можно представить в виде

$$U = \bar{U} \cup \overset{\circ}{U} \cup \vec{U}.$$

\bar{U} - подмножество неориентированных линий, для которых не существует порядок соединения вершин. Подмножество \bar{U} называется подмножеством ребер. Причем каждое ребро $u_i \in \bar{U}$ определяется неупорядоченной парой вершин x, y , которые оно соединяет и записывается: $u_i = (x, y)$ или $u_i = (y, x)$.

\vec{U} - подмножество ориентированных линий, для которых существует порядок соединения вершин. Подмножество \vec{U} называется подмножеством дуг. Причем каждая дуга $u_i \in \vec{U}$ определяется упорядоченной парой (кортежем длины два) вершин x_k, y_l , которые u_i соединяет и записывается: $u_i = (x_k, y_l)$.

Заметим, что $u_i = (x_k, y_l)$ и $u_j = (y_l, x_k)$ - это различные дуги в графе G ;

$\overset{\circ}{U}$ - подмножество линий, каждая из которых выходит и входит в одну и ту же соответствующую этой линии вершину. Называется $\overset{\circ}{U}$ подмножеством петель.

Каждая петля u_i принадлежит множеству $\overset{\circ}{U}$ и может определяться упорядоченной парой, например вида $u_i = (x_k, x_k)$.

Граф состоит из вершин, которые на плоскости изображаются нумерованными кружками или точками, и ребер, изображаемых линиями (со стрелками или без стрелок), которые соединяют некоторые из этих вершин. Однонаправленное соединение ребром двух вершин называется дугой. Двухнаправленные или ненаправленные рёбра называются звеньями. Рёбра, соединяющие вершину саму с собой, называются петлями.

Рёбра, подходящие к вершине x , называются *инцидентными* данной вершине. Соответственно говорят, что вершина x *инцидентна* рёбрам, подходящим к ней.

Количество рёбер, инцидентных вершине x , называется *степенью* вершины $s(x)$.

Вершина x называется *изолированной*, если её степень $s(x)$ равна нулю.

Если степени всех вершин равны k , то граф называется *регулярным* степени k .

Граф является конечным, если он имеет конечное число вершин и рёбер.

Бесконечные графы обладают следующими характеристиками:

1. Вершинами графа служат натуральные числа, причём вершины p и q соединены звеном в том и только том случае, если оба числа p и q простые и $|p - q| \leq 2$. Множество вершин этого графа счётно, а является ли множество рёбер счётным или только конечным – неизвестно до сих пор (проблема близнецов в теории чисел).

2. Вершинами являются числа $1, 2, \dots, n$, а каждое действительное число x , удовлетворяющее условию $i < x < i + 1$, служит дугой из вершины i в вершину $i + 1$. Граф содержит конечное множество вершин и континуум рёбер (дуг).

3. Вершинами служат все действительные числа, и при фиксированном $\delta > 0$ вершины x и y соединены ребром (звеном или петлёй) тогда и только тогда, когда $|x - y| < \delta$. Каждому значению δ отвечает свой граф, у которого множества вершин и рёбер оба континуальны.

1.2. КЛАССЫ ГРАФОВ

Класс орграфов (ориентированных графов). Это граф $G=(X,U)$, у которого $\bar{U}=\emptyset$.

Класс неорграфов (неориентированных графов). Это граф $G=(X,U)$, у которого $\vec{U}=\emptyset$.

Класс смешанных графов. Это граф $G=(X,U)$, у которого

$$\bar{U} \subset U, \vec{U} \subset U \text{ и } \bar{U} \cup \vec{U} \subseteq U.$$

Класс мультиграфов. Мультиграф - это граф $G=(X,U)$, у которого имеются параллельные (кратные) рёбра, т.е.

$$\exists x,y \in X \mid x u_k y, x u_m y, \dots, x u_p y, u_k, u_m, \dots, u_p \in U.$$

Для некоторых классов графов естественно определяется понятие *полного графа*, как такого, который содержит все рёбра, возможные при принадлежности графа данному классу и при неизменном множестве вершин. Например, в случае *p-графа* полнота означает, что при каждой вершине имеется ровно p петель (если граф при вершинах содержит петли), а каждая пара различных вершин связана ровно p -рёбрами (среди них могут быть как звенья, так и дуги).

Граф общего вида, в котором две различные вершины всегда смежны, называется *полным*.

Особо важную роль играют так называемые *обыкновенные* графы. Граф этого класса характеризуется следующими четырьмя свойствами:

- 1) он конечен;
- 2) он является неориентированным, т.е. не содержит дуг;
- 3) он не содержит петель;
- 4) он не содержит "параллельных" ("кратных") рёбер; иначе говоря, никакие две его вершины не могут соединяться более чем одним ребром (звеном).

Определение:

Обыкновенный граф – это неориентированный униграф без петель.
Униграф - это граф, в котором смежные вершины связаны только одним неориентированным ребром.

Дополнение графа. Дополнением графа $L = (X, U)$ называется граф $\bar{L} = (X, U^*)$ с тем же множеством вершин X и с множеством $U^* \doteq \{\tilde{xy} / x, y \in X \ \& \ x \neq y\} \setminus U$ рёбер. Иначе говоря, это такой граф, в котором две различные вершины смежны тогда и только тогда, когда они не смежны в исходном графе L .

Рассмотрим ещё одно важное в теории графов понятие- *скелет графа*.

Скелет графа общего вида. В случае, когда при исследовании графа $L=(X,U;P)$ общего вида требуется не полная информация о нём, а лишь знание того, какие пары его различных вершин смежны и какие нет, то прибегают к носителю такой информации - *скелету* графа L , который обозначим как $\bar{L} = (X, \bar{U})$. Граф \bar{L} относится к классу обыкновенных графов с множеством вершин тем же, что и в графе L и новым множеством рёбер \bar{U} , определённым следующим образом:

1. если в графе L есть петли, то они удаляются;
2. если в графе L есть дуги, то производится дезориентация дуг;
3. если в графе L есть кратные рёбра, то они заменяются одним эквивалентным ребром-звеном.
4. Оставшиеся рёбра образуют множество рёбер \bar{U} .

Таким образом, множество рёбер \bar{U} состоит из рёбер, полученных из множества U после выполнения описанных выше процедур 1,2,3.

Деревья.

Граф без циклов называется *ациклическим*, или *лесом*.

Деревом называется связный ациклический граф.

1.3. СПОСОБЫ ЗАДАНИЯ ГРАФОВ

1.3.1 Геометрический

Основой геометрического способа задания графа является рисунок, дающий визуальное изображение графа. Изображение графа в виде рисунка наглядно раскрывает содержательный смысл представляемого объекта. В этом способе вершины графа изображаются точками (кружками), а рёбра – линиями (со стрелками или без стрелок), концы которых подходят к вершинам графа.

Такое изображение графа ещё называют *диаграммой графа*.

1.3.2. Описание графа через предикат (инцидентор) P

Говорят, что задан граф $G=(X,U,P)$, если дано множество вершин X , множество ребер U и трёхместный предикат (инцидентор) P , определяющий, какую пару вершин x_i, x_j , принадлежащих множеству вершин X , соединяет ребро $u_k=(x_i,x_j)$. Инцидентор P удовлетворяет двум условиям:

А. Инцидентор P определён на всех таких упорядоченных тройках элементов x, u, y , для которых $x, y \in X$ и $u \in U$;

Б. $\forall u \exists x, y \{P(x, u, y) \& \forall x', y' [P(x', u, y') \Rightarrow (x = x' \& y = y') \vee (x = y' \& y = x')]\}$.

1.3.3. Матричный

1.3.3.1. Большинство задач автоматизации конструирования схем удобно решать при использовании матричного способа представления

графов. Квадратная таблица $R = \|r_{ij}\|_{n \times n}$ называется *матрицей смежности*, если её строки и столбцы соответствуют вершинам графа, а элементы r_{ij} образуются по правилу:

- $r_{ij} = 1$, если вершина x_i соединена с вершиной x_j ребром, т.е. x_i смежна x_j ;

- $r_{ij} = 0$, в противном случае.

Заметим, что для мультиграфа и смешанного графа задают:

- $r_{ij} = p$, если вершина x_i соединена с вершиной x_j p – числом рёбер;

- $r_{ij} = 0$ в противном случае.

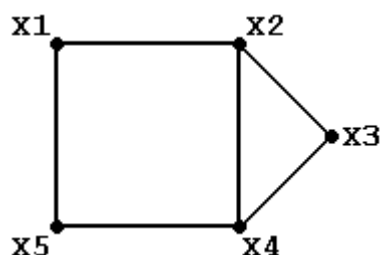


Рисунок 1.

Очевидно, что для неорграфов $r_{ij} = r_{ji}$ и для их задания можно использовать треугольную матрицу R . Так для графа на рисунке 1 матрица смежности R имеет вид:

$R =$

	X1	X2	X3	X4	X5
X1	0	1	0	0	1
X2	1	0	1	1	0
X3	0	1	0	1	0

X4	0	1	1	0	1
X5	1	0	0	1	0

Треугольная матрица R' для этого же графа запишется

$R' =$

	X1	X2	X3	X4	X5
X1	0	1	0	0	1
X2		0	1	1	0
X3			0	1	0
X4				0	1
X5					0

Строки и столбцы матрицы R соответствуют вершинам графа. На пересечении i -й строки и j -го столбца ставится элемент r_{ij} , соответствующий числу рёбер, соединяющих вершины x_i и x_j . Строки и столбцы матрицы R можно нумеровать числами натурального ряда, соответствующими индексам помеченных вершин. Петлям в графе соответствуют элементы r_{ii} главной диагонали матрицы R . Преимущество использования матриц смежности - это простота выполнения преобразований и операций над графами как для конструктора, так и для ЭВМ.

1.3.3.2. Граф можно задавать также матрицей *инциденций* B , строки которой соответствуют вершинам графа, столбцы - ребрам. Элементы b_{ij} матрицы B для неорграфа могут принимать значения (0 или 1):

$b_{ij} = 1$, если ребро u_k инцидентно вершине x_i ;

$b_{ij} = 0$, если ребро u_k не инцидентно вершине x_i .

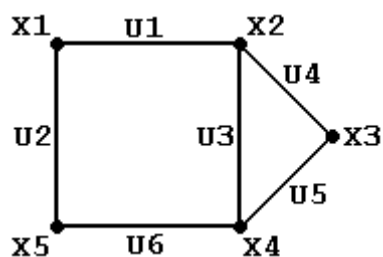


Рисунок 2.

Для графа, представленного на рисунке 2, матрица инциденций B имеет вид:

	U1	U2	U3	U4	U5	U6
X1	1	1	0	0	0	0
X2	1	0	1	1	0	0
X3	0	0	0	1	1	0
X4	0	0	1	0	1	1
X5	0	1	0	0	0	1

1.4. ЧИСЛОВЫЕ ХАРАКТЕРИСТИКИ ВЕРШИН ГРАФА

Каждая вершина x_i графа $G = (X, U)$ имеет числовую характеристику $s(x)$, которая называется степенью, или валентностью вершины. Степень вершины $s(x_i)$ это целое, положительное число, равное количеству ребер, инцидентных вершине x_i .

Для ориентированных графов различают "*полустепени исхода*" и "*полустепени захода*". Это тоже числовые характеристики вершин, соответственно равные количеству дуг, исходящих из вершины и входящих в вершину.

1.5. МАРШРУТЫ, ЦЕПИ И ЦИКЛЫ

В этой главе будут рассмотрены такие свойства графов, которые не меняются при произвольной ориентации звеньев графа, переориентации или дезориентации дуг (всех или некоторых). Мы рассмотрим такие свойства графов общего вида $L = (X, U; P)$, которые полностью определяются предикатом $\tilde{P}(x, u, y) \Leftrightarrow P(x, u, y) \vee P(y, u, x)$, называемом полуинцидентором (неоринцидентором) графа L .

О неорграфе $\tilde{L} = (X, U; \tilde{P})$ можно сказать, что он получен из L посредством дезориентации его дуг. В свою очередь L можно получить из \tilde{L} ориентацией звеньев.

Определение:

Конечная последовательность M элементов графа L :

$$M = x_0 u_1 x_1 u_2 x_2 \dots x_{k-1} u_k x_k \quad (k \geq 0),$$

для которой истинно высказывание:

$$P'(x_0, u_1, x_1) \& P'(x_1, u_2, x_2) \& \dots \& P'(x_{k-1}, u_k, x_k),$$

называется маршрутом из вершины x_0 в вершину x_k , или маршрутом, соединяющим вершину x_0 с вершиной x_k ; в случае $x_0 = x_k$ имеем циклический маршрут при вершине x_0 , или цикл. Число k носит название длины маршрута M . Иными словами, длина маршрута равняется числу ребер, входящих в него. Заметим, что маршрут - это не просто часть графа, ибо порядок его обхода играет существенную роль; так, маршрут $M1$:

$$M1 = x_k u_k x_{k-1} \dots x_2 u_2 x_1 u_1 x_0 \quad \text{при } k \geq 0$$

не совпадает с написанным выше маршрутом M , хотя и состоит из тех же самых элементов и с тем же отношением инцидентности.

Маршрут M :

$M = x_0 u_1 x_1 u_2 x_2 \dots x_{k-1} u_k x_k$ называется *цепью*, если все рёбра u_1, u_2, \dots, u_k попарно различны. Цепь в случае, если $x_0 = x_k$ при $k \geq 1$ называется *циклом*.

Цепь называется простой, если все ее вершины x_p, x_k, \dots, x_t попарно различны. В случае, когда $x_p = x_t$ имеем *простой цикл*, который, будучи цепью, в то же время не есть простая цепь.

Цепь, в которой начальная и конечная вершины не совпадают, но есть повторяющиеся вершины, называется циклической.

Гамильтоновой цепью называется простая цепь, содержащая все вершины графа.

Гамильтоновым циклом графа называется простой цикл, содержащий все вершины графа. *Граф, содержащий гамильтонов цикл*, называется гамильтоновым графом.

Эйлеровой цепью графа называется цепь, содержащая все рёбра графа. Каждое ребро входит в эйлеров цикл ровно один раз. Граф, содержащий эйлерову цепь, называется полуэйлеровым графом.

Эйлеровым циклом графа называется цепь, содержащая все рёбра графа, и каждое ребро входит в цикл ровно один раз. Граф, содержащий эйлеров цикл, называется эйлеровым.

ЛЕММА

Всякий маршрут (в частности, всякая цепь) графа содержит хотя бы одну простую цепь, соединяющую ту же пару вершин. Всякий циклический маршрут нечетной длины содержит простой цикл нечетной длины. Всякий цикл содержит простой цикл.

СЛЕДСТВИЕ

Всякий кратчайший маршрут между двумя заданными вершинами графа есть простая цепь. Всякий цикл наименьшей длины при заданной вершине является простым.

1.6. ОПРЕДЕЛЕНИЕ ЧИСЛА МАРШРУТОВ ДЛИНЫ "L" НА ГРАФЕ

Маршрутом $\mu_{i,j}$ в графе $G=(X,U)$ называется конечная последовательность вершин и рёбер вида –

$$\mu_{0,1} = (x_0, u_1, x_1, u_2, x_2, \dots, x_{l-1}, u_l, x_l),$$

где x_0, x_l - соответственно начальная и конечная вершины маршрута $\mu_{0,1}$.

Очевидно, - в конечном графе $G=(X,U)$ можно выделить только конечное число маршрутов. Длина маршрута $\mu_{i,j}$ равна числу рёбер, которые в него входят.

Часто требуется знать, - сколько маршрутов заданной длины в графе G связывает вершину x_i с вершиной x_j .

Для определения маршрутов длины q в графе $G=(X,U)$ его матрицу смежности R возводят в степень, равную q . Тогда для каждого значения степени $q=1,2,\dots,k$ значение элемента $(r_{i,j})^q$ матрицы R^q определяет количество маршрутов $\mu_{i,j}$ длиной, равной значению степени q .

ПРИМЕР: Для графа $G=(X,U)$, представленного на рисунке 3, определить количество маршрутов длины, равной 2.

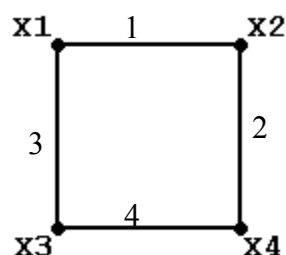


Рисунок 1.3.

Матрица смежности R графа G имеет вид:

$$R =$$

	X1	X2	X3	X4
X1	0	1	1	0
X2	1	0	0	1
X3	1	0	0	1
X4	0	1	1	0

Возведем матрицу R в квадрат:

$$R^2 =$$

	X1	X2	X3	X4
X1	2	0	0	2
X2	0	2	2	0
X3	0	2	2	0
X4	2	0	0	2

Значение каждого элемента r_{ij} матрицы R^2 равно числу маршрутов длины 2, ведущих из вершины x_i в вершину x_j .

Например, $r_{3,2}=2$ означает, что в графе два маршрута длины 2, которые ведут из вершины x_3 в вершину x_2 . Запишем их:

$$\mu_{3,2} = x_3, 3, x_1, 1, x_2; \quad \mu_{3,2} = x_3, 4, x_4, 2, x_2.$$

Вопросы и упражнения для самопроверки

1. Какую характеристику должны иметь степени вершин в эйлеровом графе?

2. Какую характеристику должны иметь степени вершин в полуэйлеровом графе?

3. Дан граф $G = (X, U)$ и граф $\bar{G} = (X, U^*)$. Дайте интерпретацию выражению $U^* \doteq \{x\tilde{y} / x, y \in X \ \& \ x \neq y\} \setminus U$.

4. Доказать, что всякий замкнутый маршрут нечётной длины содержит простой цикл. Верно ли аналогичное утверждение для маршрутов чётной длины?

5. Показать, что в любом обыкновенном графе, имеющем не менее двух вершин, найдутся две вершины с одинаковыми степенями.

6. Граф G задан матрицей смежности R с элементами $r_{1,3} = 1, r_{1,5} = 1, r_{2,3} = 1, r_{2,4} = 1, r_{3,1} = 1, r_{3,2} = 1, r_{3,4} = 1, r_{3,5} = 1, r_{4,2} = 1, r_{4,3} = 1, r_{5,1} = 1, r_{5,3} = 1$

Определить является ли граф G эйлеровым?

7. Граф G задан матрицей смежности R с элементами $r_{13} = 1, r_{15} = 1, r_{23} = 1, r_{24} = 1, r_{31} = 1, r_{32} = 1, r_{34} = 1, r_{35} = 1, r_{42} = 1, r_{43} = 1, r_{51} = 1, r_{53} = 1$.

Для каждой пары вершин данного графа G определить: - количество маршрутов длины $d = 3$, соединяющих их.

8. На рисунке 1.4 изображены графы G_1 — G_9 . Определить для каждого графа его класс, вершинные характеристики. Сравнить графы G_1 — G_2 ; G_4 — G_5 ; G_8 — G_9 .

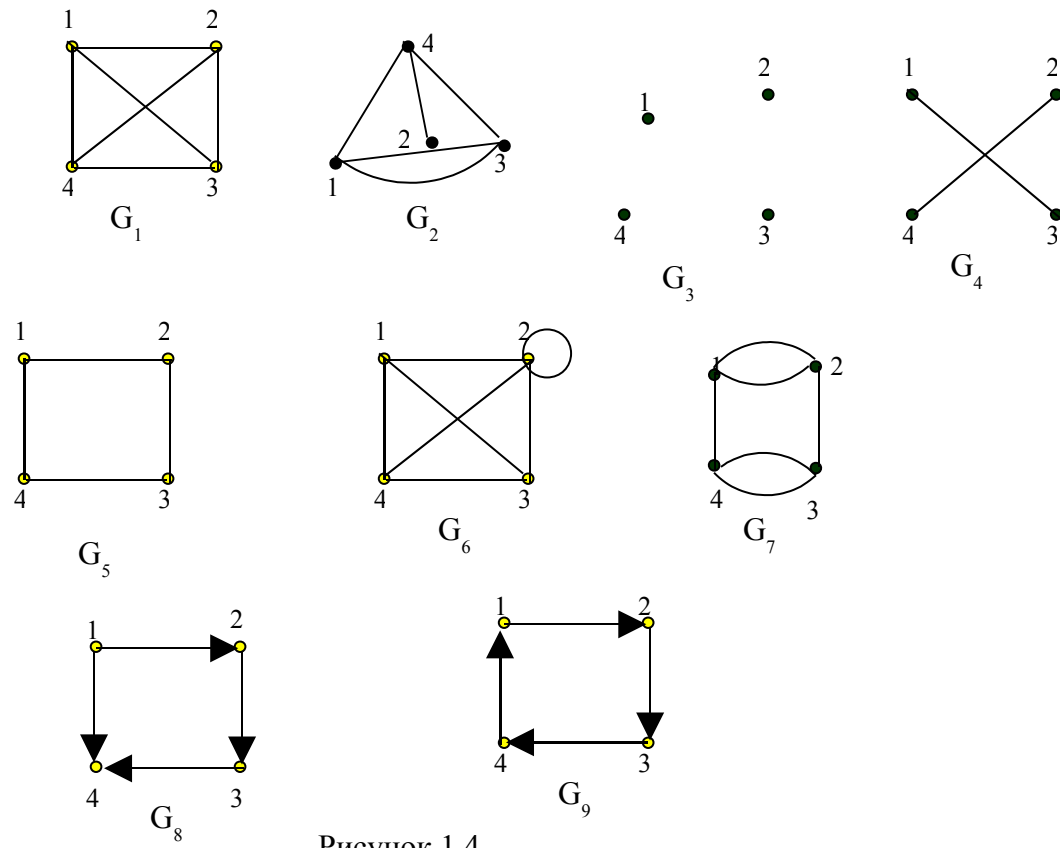


Рисунок 1.4

Тема 2. Операции на графах

2.1. УДАЛЕНИЕ ВЕРШИНЫ

При удалении вершины удаляются все инцидентные ей рёбра.

Пример.

В графе $G = (X, U)$ на рисунке 2.1 удалить вершину x_1 .

Результат выполнения данной операции — Граф G' (рисунок 2.2).

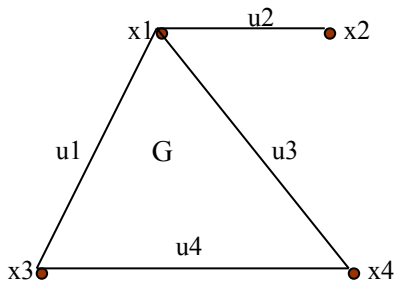


Рисунок 2.1

\Rightarrow

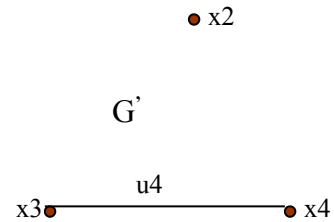


Рисунок 2.2

2.2. УДАЛЕНИЕ РЕБРА

При удалении ребра инцидентные ему вершины (концевые) не удаляются!

Пример.

В графе $G = (X, U)$, представленном на рисунке 2.1, удалить ребро u_2 .

Граф G'' (рисунок 2.3) – результат выполнения данной операции.

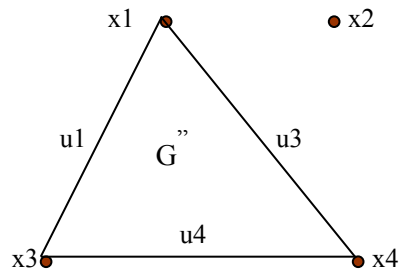


Рисунок 2.3

Если из графа требуется удалить некоторое множество вершин и рёбер, то эта процедура сводится к последовательному удалению каждой вершины отдельно и отдельно каждого ребра.

2.3. ЗАМЫКАНИЕ (ОТОЖДЕСТВЛЕНИЕ) ВЕРШИН

Для любой заданной пары вершин V_i, V_j операция замыкания сводится к отождествлению этих вершин в новую вершину V_k , при этом все рёбра, инцидентные вершинам V_i и V_j становятся инцидентными вершине V_k .

ПРИМЕР: На графе $G=(X,U)$, представленном на рисунке 4а), выполнить операцию «замыкания» вершин x_1 и x_2 .

На рисунке 4б) представлен граф G'' , полученный из графа G , после «замыкания» вершин x_1 и x_2 , где вершина $x_k=(x_1+x_2)$.

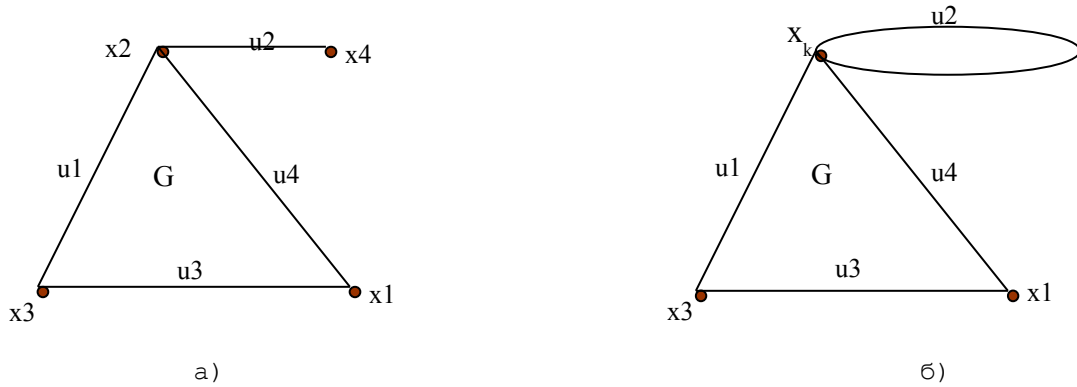


Рисунок 2.4

2.4. СТЫГИВАНИЕ ВЕРШИН ГРАФА ПО РЕБРУ

Операция стягивания вершин x_i и x_j графа $G(X,U)$ по инцидентному им ребру u_k включает операцию удаления ребра u_k и операцию отождествления вершин x_i, x_j .

ПРИМЕР: На рисунке 2.5б) представлен граф G^\wedge , полученный из графа G (рисунок 2.5а) операцией стягивания вершин x_4, x_2 по ребру u_2 .

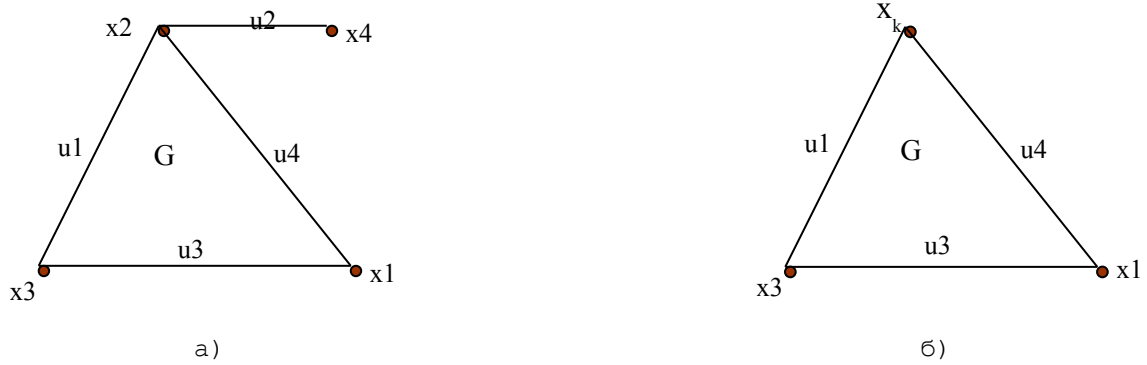


Рисунок 2.5

2.5. СИММЕТРИЧЕСКАЯ РАЗНОСТЬ ГРАФОВ

Пусть $G=(V, X)$ и $H=(U, Y)$ – два графа.

Через $G \oplus H$ будет обозначаться граф, называемый симметрической разностью графов G и H с множеством вершин $W = V \cup U$ и множеством рёбер $Z = X \oplus Y$, состоящим из тех и только тех рёбер, которые входят ровно в одно из множеств X или Y .

2.6. ПЕРЕСЕЧЕНИЕ ГРАФОВ

Пересечение графов $G=(V, X)$ и $H=(U, Y)$ есть граф $G \cap H$, вершинами которого являются вершины, присутствующие одновременно и в графе G и в графе H , а множество рёбер состоит только из рёбер, присутствующих одновременно и в графе G и в графе H .

2.7. ОБЪЕДИНЕНИЕ ГРАФОВ

Объединением графов $G=(V, X)$ и $H=(U, Y)$ называется граф $E=(V \cup U, X \cup Y)$.

Примеры операций симметрической разности, объединения, пересечения над графами G и H (рисунок 2.6).

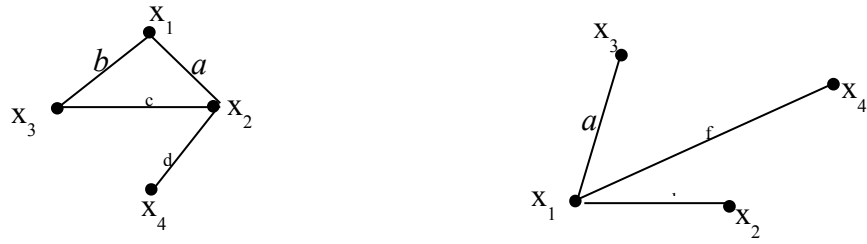


Рисунок 2.6

Симметрическая разность $G \oplus H$ (Н графов G и H :

$G \oplus H$

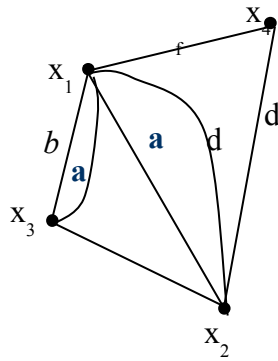
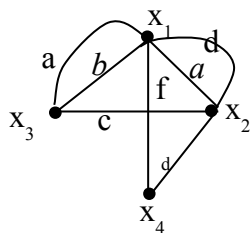


Рисунок 2.7.

Объединение $G \cup H$ графов G и H :

$G \cup H$



—

Рисунок 2.8.

Пересечение $G \cap H$ графов G и H (рисунки 2.10, 2.9):

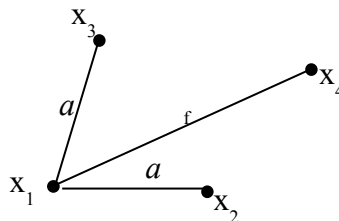
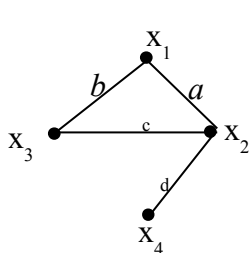
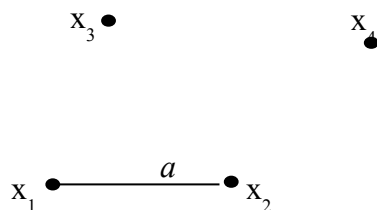
Рисунок 2.9
 $G \cap H$ 

Рисунок 2.10

Упражнения для самопроверки

1. Для неорграфов $G(X,U)$, $H(Z,V)$, $L(Y,W)$ выполнить следующие операции:

$G \oplus L$; $G \cup H$; $H \cap G$; $H \cap L$; $G \cap L$, если $U = \{(x_1, x_2), (x_1, x_5), (x_1, x_3), (x_1, x_4), (x_4, x_5), (x_3, x_2), (x_3, x_4), (x_3, x_5)\}$; $V = \{(z_1, z_2), (z_1, z_3), (z_1, z_5), (z_3, z_4), (z_4, z_5)\}$; $W = \{(y_1, y_3), (y_1, y_5), (y_2, y_3), (y_5, y_3), (y_4, y_3), (y_2, y_4)\}$.

2. Определить какой из графов G , H , L является эйлеровым, полуэйлеровым, гамильтоновым.

3. Даны неорграфы $G(X,U)$, $H(X,V)$, $L(X,W)$. Пусть $X = \{x_1, x_2, x_3, x_4, x_5\}$, $U = \{(x_3, x_4), (x_4, x_5)\}$, $V = \{(x_2, x_1)\}$, $W = \{(x_2, x_1), (x_3, x_4), (x_4, x_5)\}$.

Определить: - результатом, какой операции на графах G и H является граф L .

4. Даны неорграфы $G_1(X,U)$, $G_2(X,V)$, $G_3(X,W)$. Пусть $X=\{x_1,x_2,x_3\}$ в графах G_1 и G_2 , а в графе G_3 — $X=\{x_1,x_2\}$; $U=\{(x_3, x_4), (x_4, x_5), (x_2,x_2)\}$, $V=\{(x_2,x_1),(x_1,x_1)\}$, $W=\{(x_2,x_1), (x_3,x_4), (x_4,x_5), (x_1,x_1), (x_2,x_2)\}$.

Определить: - результатом, какой операции на графах G_1 и G_2 является граф G_3 .

5. Даны неорграфы $G(X,U)$, $H(X,V)$, $L(X,W)$. Пусть $X=\{x_1,x_2, x_3, x_4,x_5\}$, $U=\{(x_3,x_4), (x_2,x_5), (x_2,x_3), (x_1,x_4)\}$, $V=\{(x_1,x_5), (x_5,x_3), (x_3,x_4)\}$, $W=\{(x_2,x_1)$.

Определить: - результатом, какой операции на графах G и H является граф L .

ТЕМА 3. Части графа

3.1. ПОДГРАФ

Граф $G^{\wedge} = (X'', U'')$ является *подграфом* графа $G = (X, U)$, если X'' является подмножеством X ($X'' \subset X$) и U'' является подмножеством U ($U'' \subset U$)

Пример: Дан граф G (рисунок 3.1).

Граф G_1 (рисунок 3.2) является подграфом графа G .

Граф G_2 (рисунок 3.3) является подграфом графа G .

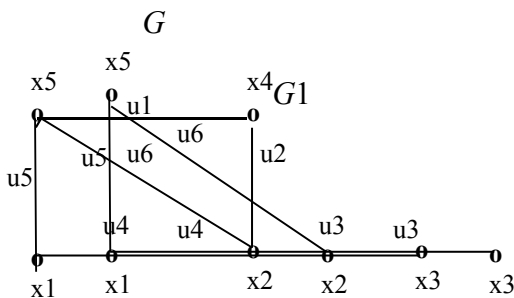


Рисунок 3.1

Рисунок 3.2

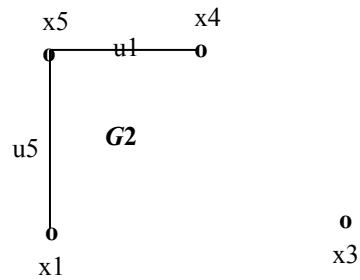


Рисунок 3.3

3.2 ЧАСТИЧНЫЙ ГРАФ.

Граф $L^{\wedge} = (X'', Y'')$ является *частичным графом* (суграфом) графа $L(X, U)$, если:

$X'' = X$; U'' является подмножеством U , т.е. $U'' \subseteq U$.

Примеры:

1. Граф $G3$ (рисунок 3.4) является *частичным графом* графа G (рисунок 3.1).

2. Граф $L1$ (рисунок 3.6) является *частичным графом* графа L (рисунок 3.5)

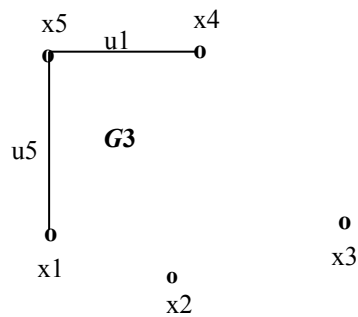


Рисунок 3.4

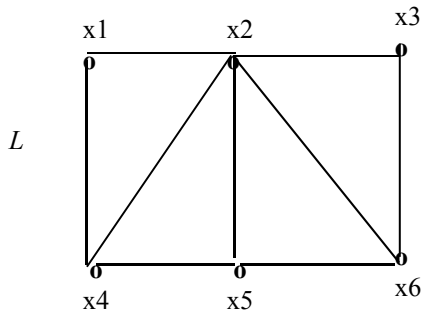


Рисунок 3.5

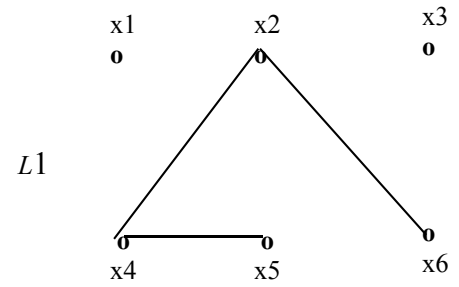


Рисунок 3.6

Упражнения для самопроверки

1. Определить класс графа, заданного матрицей смежности (рисунок 3.7).

	1	2	3	4	5	6	7
1	0	0	0	0	0	3	1
2	0	0	0	0	0	0	0
3	0	0	0	2	0	0	1
4	0	0	2	0	3	2	0
5	0	0	0	2	0	1	1
6	3	0	0	2	0	0	1
7	1	0	1	0	1	1	0

Рисунок 3.7

2. Построить одновершинные, дву-вершинные, три-вершинные подграфы для графа, заданного матрицей смежности (рисунок 3.8).

	1	2	3	4	5	6	7
1	0	0	0	0	0	1	1
2	0	0	0	0	0	0	0
3	0	0	0	1	0	0	1
4	0	0	1	0	1	1	0
5	0	0	0	1	0	1	1
6	1	0	0	1	1	0	1
7	1	0	1	0	1	1	0

Рисунок 3.8

3. Построить частичные графы для графа, заданного матрицей смежности (рисунок 3.8).

ТЕМА 4. Взвешенные графы. Метрика графа

4.1. ОСНОВНЫЕ ПОНЯТИЯ И ОПРЕДЕЛЕНИЯ

Задавая на вершинах и рёбрах графа $L=(X,U)$ функции p :

$$X \rightarrow M_p, q: U \rightarrow M_q,$$

где M_p и M_q – произвольные множества, получим взвешенный граф $L(p,q) = (X, U, p, q)$.

На множествах X и U можно задавать и более чем по одной функции или, напротив, задать функцию только на рёбрах.

К взвешенным графам принадлежат электрические схемы, сети коммуникаций, информационные и логические сети, графы автоматов, сетевые графики работ и многое другое. Ограничимся здесь отдельным вопросом, в котором наличие весов является идеей чистой теории графов: длины рёбер. Пусть $L(q) = (X, U; q)$ – обыкновенный граф с весовой функцией q , относящей каждому ребру $u \in U$ действительное число $q(u)$ (0 в качестве длины. Если M – маршрут на графе L , то сумма $q(M) = \sum_{u \in M} q(u)$

по всем его рёбрам называется его q -длиной, а просто «длина» понимается как количество рёбер маршрута (каждое ребро графа надо считать столько раз, сколько оно встречается в маршруте). Число

$$\rho(x,y) = \min \{q(M) / M(M(x,y))\} \quad (4.1),$$

где $M(x,y)$ – множество всех простых цепей из x в y , называется *расстоянием* между вершинами $x, y \in X$ взвешенного графа $L(q)$; если

$x = y$, то M – цепь нулевой длины и её длина $q(M)=0$, а если вершины x и y отделены в графе, то $\rho(x,y) = +\infty$.

Термин «расстояние» оправдан тем, что функция ρ , определённая посредством выражения (4.1), удовлетворяет трём аксиомам метрики (аксиомы Фрише):

$$\forall x, y \in X [\rho(x, y) = 0 \Rightarrow x = y], \quad (4.2)$$

$$\forall x, y \in X [\rho(x, y) = \rho(y, x)] \quad (4.3)$$

$$\forall x, y \in X [\rho(x, y) + \rho(y, z) = \rho(x, z)], \quad (4.4)$$

т.е. ρ является метрикой на множестве вершин X .

В частном случае, когда все $q(u)=1$ и, значит q - длина всякой цепи совпадает с её обычной длиной, метрика $\rho = \rho_L^1$ графа $L[1]$ называется *естественной метрикой* обыкновенного графа $L=(X, U)$.

Вершина $x_0 \in X$ графа $L = [q]$ называется *центральной*, если

$$\forall x \in X [\max_{y \in X} \rho(x, y) \geq \max_{y \in X} \rho(x_0, y)] .$$

Вершина $x_0 \in X$ графа $G=[q]$ называется *периферийной*, если

$$\forall x \in X [\max_{y \in X} \rho(x, y) \leq \max_{y \in X} \rho(x_0, y)] .$$

В силу того, что множество X конечно, а величина $+$ (допускается как возможное значение функции ρ), вершины каждого из двух указанных типов всегда существуют. Величина

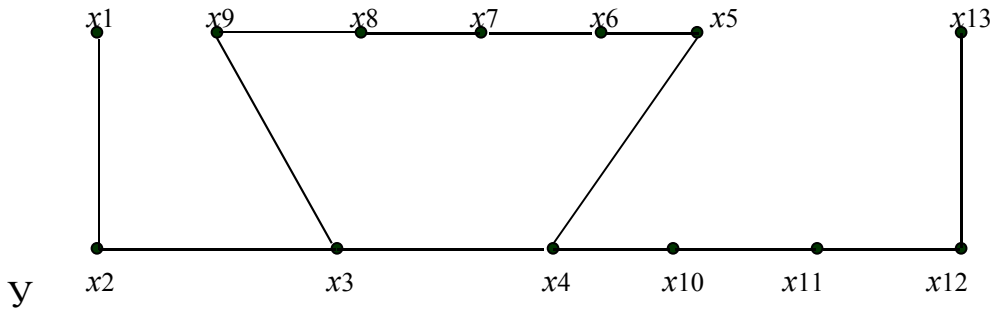
$$r(G) = \min_{x \in X} \max_{y \in X} \rho(x, y)$$

носит название *радиуса*, а величина

$$d(G) = \max_{x, y \in X} \rho(x, y)$$

называется *диаметром* графа $L(X, U)$. У несвязного графа $\max_{x, y \in X} \rho(x, y) = +\infty$, для любой пары вершин $x, y \in X$, поэтому каждая его вершина x является одновременно и центральной, и периферийной, а радиус и диаметр бесконечны.

ПРИМЕР: Дан граф $L=(X, U)$ (рисунок 4.1) с естественной метрикой ρ .



данного

Рисунок 4.1

графа вершины x_4 и x_{10} — центральные, вершины x_1, x_7, x_8, x_{13} (периферийные, радиус $r(L)=4$, диаметр $d(L)=7$.

4.2. СПОСОБ НАХОЖДЕНИЯ МАТРИЦЫ МЕТРИКИ ДЛЯ ГРАФА ОБЩЕГО ВИДА

Для нахождения метрики $\rho = \rho_L^1$ графа $L = (X, U)$ достаточно знать его матрицу смежности R над булевой алгеброй $B = (0, 1)$, где элемент матрицы $r_{ij} = 1$, если вершины x_i и x_j — смежны и $r_{ij} = 0$, в противном случае.

Все дальнейшие действия над элементами матрицы R выполняются по правилам алгебры логики Буля:

$$1 + 1 = 1; 0 + 0 = 0; 1 + 0 = 1; 0 * 0 = 0; 1 * 0 = 0.$$

Сопоставляя уже известные нам способы для установления существования в графе маршрутов длины l , можно утверждать, что при возведении в степень матрицы $S = R + E$, где E - единичная матрица той же размерности, что и размерность матрицы R , на некотором шаге возведения в степень получим: $S^k = S^{k+1}$, т.е. устойчивую матрицу S в степени " k ".

Значения степеней p матрицы S^p : $p = \{k, k-1, k-2, \dots, 1\}$, равны длинам простых кратчайших цепей, связывающих вершины x_i и x_j .

Таким образом, последовательно возводя в степень $p = \{1, 2, 3, \dots, k\}$ матрицу S , до получения устойчивой матрицы S^k можно определить расстояния между всеми вершинами графа $L=(X,U)$, построив матрицу метрики графа L .

4.3. АЛГОРИТМ ПОСТРОЕНИЯ МАТРИЦЫ МЕТРИКИ ГРАФА

Введём обозначения, которые будут использоваться в алгоритме построения матрицы метрики (матрицы отклонений):

Граф $L=(X,U)$;

R - матрица смежности заданного графа L ;

E – единичная матрица;

M - матрица метрики;

Матрица смежности R графа L с элементами логического типа:

$$r_{ij} = \begin{cases} 1, & \text{если вершины } x_i, x_j \text{ – смежны;} \\ 0, & \text{в противном случае} \end{cases}$$

S — матрица $S = R + E$.

4.3.1. Описание алгоритма

Значения элементов $m_{i,j}$ матрицы M определяются за несколько итераций по результатам последовательного возведения матрицы $S=(E+R)$ в степень $p = \overline{1,k}$ до получения устойчивой матрицы S^k , где k - степень устойчивой матрицы S^k . Матрица S^k называется *устойчивой*, если

$$S^k = S^{k+1}.$$

Шаг 1. Задаём матрицу метрики $M = (m_{i,j})_{n \times n}$. Размерность матрицы M равна размерности матрицы R . Все элементы $m_{i,j}$ матрицы M не определены.

Шаг 2. Начальное значение степени k матрицы S равно «1»: - $k=1$.

$\forall m_{i,j}$ присваиваем значение « 0 », на основании 1 – ой аксиомы Фрише.

Шаг 3. Всем элементам m_{ij} , значения которых не определены, присвоить значение степени k , если соответствующие им элементы матрицы $S^k \neq 0$. (Значения элементов m_{ij} определяются только один раз).

Шаг 4. Проверяем, в матрице M имеются элементы m_{ij} , значения которых ещё не определены?

Если такие элементы имеются, то переходим к шагу 4; в противном случае – к шагу 7.

Шаг 5. Повышаем степень k матрицы R : $k = k+1$.

Шаг 6. Проверяем, является ли матрица R^{k-1} устойчивой.

Если матрица R^{k-1} - неустойчивая, то переходим к шагу 3.

Иначе – переходим к шагу 7.

Шаг 7. Всем элементам m_{ij} матрицы M , значения которых остались неопределенными, присваиваем значение ∞ (бесконечность).

Шаг 8. Матрица метрики $M=(m_{ij})$ построена. Конец алгоритма.

Примечание: Элементом m_{ij} значения присваиваются только один раз. Следовательно, если значение элемента m_{ij} уже определено, то оно больше не меняется.

Радиус графа определяется по матрице метрики следующим способом: в каждой строке матрицы M выделяется значение максимального элемента.

Наименьшее из выделенных значений и есть величина радиуса графа.

Диаметр графа определяется по матрице метрики следующим способом: в каждой строке матрицы M выделяется значение максимального элемента. Наибольшее из выделенных значений и есть величина диаметра графа.

Вопросы и упражнения для самопроверки

1. В каких случаях при построении матрицы метрики M в ней останутся неопределённые элементы после достижения устойчивой матрицы смежности графа: — $R^k = R^{k+1}$?

2. Возможна ли ситуация, когда $R^k = R^{k+1}$ при $k = 1$?

3. Для графа, заданного матрицей смежности (рисунок 4.2), определить его радиус и диаметр.

	1	2	3	4	5	6	7
1	0	1	0	0	0	1	1
2	1	0	0	0	0	0	1
3	0	0	0	1	0	0	1
4	0	0	1	0	1	1	0
5	0	0	0	1	0	1	1
6	1	0	0	1	1	0	1
7	1	1	1	0	1	1	0

Рисунок 4.2

4. Для графа, заданного матрицей смежности (рисунок 4.3), выполнить следующее:

- 4.1. Построить метрику графа.
- 4.2. Вычислить радиус, диаметр данного графа.
- 4.3. Найти все периферийные и центральные вершины графа.

	1	2	3	4	5	6	7
1	0	0	0	0	0	1	1
2	0	0	0	0	0	0	0
3	0	0	0	2	0	0	1
4	0	0	2	0	3	2	0
5	0	0	0	3	0	1	1
6	3	0	0	2	1	0	1
7	1	0	1	0	1	1	0

Рисунок 4.3

ТЕМА 5. Структурный анализ графов

Задача структурного анализа графов является одной из центральных задач теории графов и имеет широкое применение при решении фундаментальных теоретических проблем в программировании и прикладных задачах при анализе объектов математических моделей.

Данная задача связана с базовыми понятиями: - связность графа; компонента связности графа, число компонент связности, полный граф, максимальные полные подграфы (максимальная клика) и максимальные пустые подграфы, скелет графа. Рассмотрим эти понятия.

5.1. СВЯЗНОСТЬ ГРАФА

Говорят, что две вершины в графе $G=(X,U)$ *связаны*, если существует (простая) цепь соединяющая их. Отношение связности вершин является эквивалентностью. Классы эквивалентности по отношению связности называются *компонентами связности*. Число компонент связности графа G будем обозначать $k(G)$. Граф G является связным тогда и только тогда, когда $k(G) = 1$. Если $k(G) > 1$, то G — *несвязный* граф. Обозначим число вершин и число рёбер графа G соответственно через $p(G)$ и $q(G)$. Граф, состоящий только из изолированных вершин (в котором $k(G) = p(G)$ и $q(G) = 0$), называется вполне несвязным.

5.2. СКЕЛЕТ ГРАФА

Скелет G^{\wedge} графа G получается путём отбрасывания в графе G всех петель и заменой кратных рёбер одним эквивалентным, а также дезориентацией имеющихся дуг.

5.3. ПОКРЫВАЮЩИЕ МНОЖЕСТВА ВЕРШИН И РЕБЕР ГРАФА G

Говорят, что в графе G вершина *покрывает* инцидентные ей рёбра, а ребро *покрывает* инцидентные ему вершины.

Множество вершин графа G , покрывающее все рёбра, называется *вершинным покрытием*. Наименьшее число вершин во всех вершинных покрытиях называется числом вершинного покрытия $\alpha_0(G)$.

Множество таких рёбер, которые покрывают все вершины, называется *рёберным покрытием*. Наименьшее число рёбер во всех рёберных покрытиях называется числом рёберного покрытия $\alpha_1(G)$.

5.4. МАКСИМАЛЬНЫЕ ПОЛНЫЕ И МАКСИМАЛЬНЫЕ ПУСТЫЕ ПОДГРАФЫ

Определения

1). Подграф L' называется *максимальным пустым подграфом* графа $L=(X,U)$, если L' не является собственным подграфом никакого большего максимального пустого подграфа данного графа L .

2). Подграф L'' называется *максимальным полным подграфом* графа $L=(X,U)$, если L'' не является подграфом никакого большего максимального полного подграфа графа L .

5.5. КЛИКА ГРАФА.

Подмножество $X_k \subset X$ вершин графа $G=(X,U)$ называется *кликой*, если $G_k=(X_k, \Gamma_k)$ — полный подграф некоторого графа без петель $G=(X, \Gamma)$, соответствующего графу $G=(X,U)$.

Клика $X_{k_{\max}}$ называется *максимальной*, если $G_{k_{\max}}$ строго не содержится ни в каком другом полном подграфе графа G .

5.6. АЛГОРИТМ НАХОЖДЕНИЯ ВСЕХ МАКСИМАЛЬНЫХ ПУСТЫХ И ВСЕХ МАКСИМАЛЬНЫХ ПОЛНЫХ ПОДГРАФОВ (МАКСИМАЛЬНЫХ КЛИК) В ГРАФЕ ОБЩЕГО ВИДА

Задача выявления всех максимальных полных и максимальных пустых подграфов в заданном графе $L = (X, U; P)$ общего вида легко сводится к случаю обыкновенных графов. Поэтому, для практического выявления всех максимальных полных и максимальных пустых подграфов в произвольном графе достаточно уметь выявлять только максимальные полные и максимальные пустые подграфы обыкновенных графов.

Приведём алгоритм выявления всех максимальных пустых подграфов в заданном графе общего вида, основанный на работах Х.Магу и Дж.Уэйсмана:

Шаг 1. Для графа $L = (X, U; P)$ общего вида построить его скелет $\bar{L} = (X, \bar{U})$.

Шаг 2. Построить матрицу инцидентий A графа $\bar{L} = (X, \bar{U})$, элементы которой a_{ij} принимают значения 0 либо 1 ($i = \overline{1, n}$; $j = \overline{1, m}$, где $n = |X|$ - число вершин в \bar{L} , $m = |\bar{U}|$ - число рёбер в $\bar{L} = (X, \bar{U})$).

Шаг 3. Дополним систему логическими переменными $x_1, x_2, \dots, x_i, \dots, x_n$, которые принимают значения 0 и 1, и подчиним её условиям:

$$x_i^2 = x_i; \quad x_i + 1 = 1; \quad x_i + x_i = x_i, \quad 1+1=1, \text{ т.е. } 2=1, \quad (i=1, 2, \dots, n),$$

а также законам коммутативности, ассоциативности и дистрибутивности.

Шаг 4. Из матрицы инцидентий $A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ a_{21} & a_{22} & \dots & a_{2m} \\ \cdot & \cdot & \cdot & \cdot \\ a_{n1} & a_{n2} & \dots & a_{nm} \end{pmatrix}$

графа $\bar{L} = (X, \bar{U})$, где n, m соответственно равны числу вершин и рёбер графа, образуем матрицу

$$Ax = \begin{pmatrix} a_{11}x_1 & a_{12}x_1 & \dots & a_{1m}x_1 \\ a_{21}x_2 & a_{22}x_2 & \dots & a_{2m}x_2 \\ \cdot & \cdot & \cdot & \cdot \\ a_{n1}x_n & a_{n2}x_n & \dots & a_{nm}x_n \end{pmatrix}$$

и составим произведение

$$\Pi_L = \Pi_L(x_1, x_2, \dots, x_n) = \prod_{j=1}^m \sum_{i=1}^n a_{ij} x_i$$

Очевидно, что j -й сомножитель произведения Π_L есть сумма двух слагаемых, соответствующих тем двум вершинам, которые в графе соединены j -м ребром.

Шаг 5. Преобразовать произведение Π_L к бескобочному виду и привести всю сумму к минимальной форме, пользуясь дистрибутивным, ассоциативным, коммутативным законами и применяя закон поглощения:

а) $a+a \cdot b = a$; б) $(a+b)(a+c) \dots (a+p) = a+b \cdot c \dots p$, где a, b, c, \dots, p - логические переменные, принимающие значения $0; 1$, выполняя при этом условия, описанные в п.3.

В результате выполненных преобразований выражение Π_L будет иметь минимальную форму и представлять сумму произведений переменных из множества $x_1, x_2, \dots, x_i, \dots, x_n$, т.е. многочлен. Обозначим его \sum_L .

Шаг 6. Для каждого слагаемого многочлена \sum_L выделим переменные которые в него не входят, но входят в множество $\{x_1, x_2, \dots, x_i, \dots, x_n\}$. Эти переменные порождают максимальные пустые подграфы данного графа L , так как соответствующие им вершины графа L образуют максимальные пустые подграфы.

ПРИМЕР. В графе $L = (X, U; P)$, представленном на рисунке 5.1, выделить все максимальные пустые подграфы.

Матрица смежности В графа L содержит элементы (b_{ij}) , $i = \overline{1,5}; j = \overline{1,5}$

равные: $b_{11} = b_{22} = b_{33} = b_{55} = 0$; $b_{44} = 1$; $b_{12} = 2$; $b_{13} = 1$; $b_{14} = 0$; $b_{15} = 0$;

$b_{21} = 2$; $b_{23} = 0$; $b_{24} = 2$; $b_{25} = 0$; $b_{31} = 1$; $b_{32} = 0$; $b_{34} = 0$; $b_{35} = 3$;

$b_{41} = 0$; $b_{42} = 2$; $b_{43} = 0$; $b_{45} = 1$; $b_{51} = 0$; $b_{52} = 0$; $b_{53} = 3$; $b_{54} = 1$;

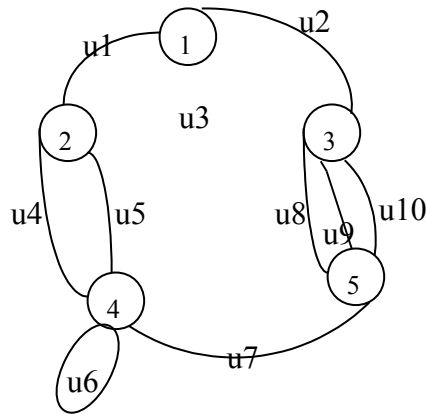


Рисунок 5.1

Шаг 1. Строим скелет $\overline{L} = (X, \overline{U})$ (рисунок 5.2) графа L .

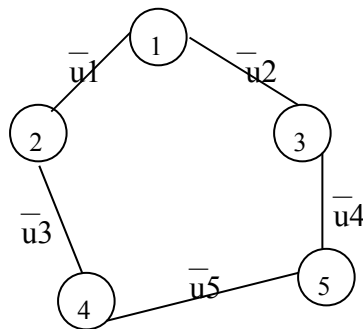


Рисунок 5.2

Шаг 2. Для графа \bar{L} определим его матрицу инциденций A :

$$A = \begin{array}{c|ccccc} & \bar{u1} & \bar{u2} & \bar{u3} & \bar{u4} & \bar{u5} \\ \hline 1 & 1 & 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 1 & 0 & 0 \\ 3 & 0 & 1 & 0 & 1 & 0 \\ 4 & 0 & 0 & 1 & 0 & 1 \\ 5 & 0 & 0 & 0 & 1 & 1 \end{array}$$

Шаг 3. Введём новые логические переменные x_1, x_2, x_3, x_4, x_5 (по числу вершин в графе \bar{L}) и из матрицы A образуем матрицу Ax :

$$Ax = \begin{array}{c|ccccc} & \bar{u1} & \bar{u2} & \bar{u3} & \bar{u4} & \bar{u5} \\ \hline 1 & x_1 & x_1 & 0 & 0 & 0 \\ 2 & x_2 & 0 & x_2 & 0 & 0 \\ 3 & 0 & x_3 & 0 & x_3 & 0 \\ 4 & 0 & 0 & x_4 & 0 & x_4 \\ 5 & 0 & 0 & 0 & x_5 & x_5 \end{array}$$

Шаг 4. Составим произведение Π_L

$$\Pi_L = (x_1 + x_2)(x_1 + x_3)(x_2 + x_4)(x_3 + x_5)(x_4 + x_5)$$

Шаг 5. Преобразуем выражения Π_L к минимальной форме:

$$\Pi_L = (x_1 + x_2)(x_1 + x_3)(x_2 + x_4)(x_3 + x_5)(x_4 + x_5) =$$

(перемножаем скобки первую со второй и третью с пятой)

$$= (x_1 + x_2x_3)(x_4 + x_2x_5)(x_3 + x_5) =$$

(перемножаем скобки первую со второй)

$$= (x_1x_4 + x_1x_2x_5 + x_2x_3x_4 + x_2x_3x_5)(x_3 + x_5) =$$

(перемножаем скобки первую со второй)

$$= x_1x_3x_4 + x_1x_4x_5 + x_1x_2x_3x_5 + x_1x_2x_5 + x_2x_3x_4 + x_2x_3x_4x_5 + x_2x_3x_5 + x_2x_3x_5 =$$

(применяем законы указанные в п.п. 3,5 данного пособия)

$$= x_1 x_3 x_4 + x_1 x_4 x_5 + x_1 x_2 x_5 + x_2 x_3 x_4 + x_2 x_3 x_5 .$$

Преобразование выражения Π_L закончено. Получена минимальная форма - полином $\sum L$.

Шаг 6. Выделим для каждого слагаемого полинома

$$\sum L = x_1 x_3 x_4 + x_1 x_4 x_5 + x_1 x_2 x_5 + x_2 x_3 x_4 + x_2 x_3 x_5$$

его дополнение до множества переменных $\{x_1, x_2, x_3, x_4, x_5\}$:

$$(x_2 x_5); (x_2, x_3); (x_3, x_4); (x_1, x_5); (x_1, x_4)$$

полученные дополнения порождают максимальные пустые подграфы графа \bar{L} , и заданного графа L .

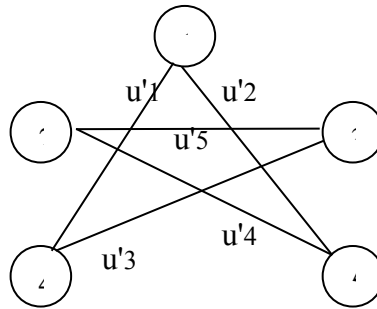
Алгоритм Х.Магу и Дж. Уэйсмана может быть применён и для выявления в графе $L=(X,U; P)$ общего вида всех максимальных полных подграфов. Для этого необходимо построить для заданного графа $L=(X,U;P)$ его скелет – граф $\bar{L}=(X,\bar{U})$, а для графа $\bar{L}=(X,\bar{U})$ построить дополнительный граф $L^*=(X,U^*)$ (определение дополнительного графа дано в теме 1 данного методического пособия). Получить дополнительный граф легко, если исходный граф задать матрицей смежности его вершин, в которой всем элементам, равным нулю присвоить значение “1”, а всем элементам, значения которых не равны нулю присвоить значение “0”.

Далее для полученного графа $L^*=(X,U^*)$ с помощью алгоритма Х.Магу и Дж. Уэйсмана (рассмотренного выше) выявляем все максимальные пустые подграфы. Эти подграфы являются максимальными полными подграфами для графов $\bar{L}=(X,\bar{U})$ и $L=(X,U;P)$.

ПРИМЕР. В графе $L=(X,U;P)$, представленном на рисунке 1, выделить все максимальные полные подграфы .

Шаг 1. Строим скелет $\bar{L} = (X, \bar{U})$ (рисунок 5.2) графа L .

Шаг 2. Для графа $\bar{L} = (X, \bar{U})$ строим его дополнительный граф $L^* = (X, U^*)$ (рисунок 5.3), в котором с помощью алгоритма Х.Магудж.Уэйсмана выявляем максимальные пустые подграфы.



Приведём
решения данной задачи -

Рисунок 5.3

окончательный результат
полином

$$\sum L = x_1 x_2 x_3 + x_1 x_2 x_4 + x_1 x_3 x_5 + x_2 x_4 x_5 + x_3 x_4 x_5$$

и дополнения для его слагаемых: (x_4, x_5) ; (x_3, x_5) ; (x_4, x_2) ; (x_1, x_3) ; (x_1, x_2) которые порождают все максимальные пустые подграфы графа $L^* = (X, U^*)$ и максимальные полные подграфы графа $\bar{L} = (X, \bar{U})$ и заданного графа $L = (X, U; P)$.

Упражнения для самопроверки

1. Граф G задан матрицей смежности $R_{5 \times 5}$ с элементами:

$$r_{13} = 2, r_{15} = 1,$$

$$r_{23} = 1, r_{24} = 1, r_{31} = 1, r_{32} = 2, r_{34} = 2, r_{35} = 1, r_{42} = 1, r_{43} = 2, r_{51} = 1, r_{53} = 1.$$

Неуказанные элементы $r_{ij} = 0$.

Найти все максимальные пустые подграфы, с помощью алгоритма Х.Магу -Дж.Уэйсмана. Привести их геометрическое представление.

2. Построить скелет графа и дополнительный граф для графов, представленных на рисунках 5.4, 5.5.

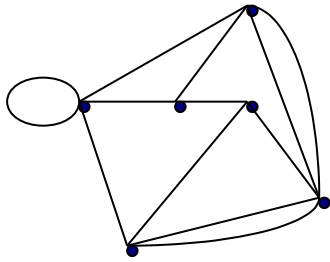


Рисунок 5.4

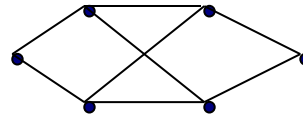


Рисунок 5.5

3. Найти все максимальные полные подграфы в графе, представленном на рисунке 5.4.

4. Найти (аналитическим способом) число вершинного покрытия $\alpha_0(G)$ для графа, представленного на рисунке 5.4.

5. Найти (аналитическим способом) число рёберного покрытия $\alpha_1(G)$ для графа, представленного на рисунке 5.5.

6. Определить дополнение \bar{G} графа G , если: а) G — пятиугольник;
б) G — треугольник.

ТЕМА 6. Раскраска графов

6.1. ПРАВИЛЬНАЯ РАСКРАСКА. ХРОМАТИЧЕСКОЕ ЧИСЛО

Под раскраской графа понимают приписывание его вершинам (рёбрам) какого либо цвета.

Если в графе $G = (X, U)$ его вершины (рёбра) раскрашены так, что смежные вершины (рёбра) окрашены в разные цвета, то такую раскраску

называют *правильной*.

Если на правильную раскраску затрачено p цветов, то граф называется p -хроматическим.

Наименьшее натуральное число p , для которого граф является p -хроматическим, называется *хроматическим числом графа* и обозначается $\chi(G)$.

Для решения задачи правильной раскраски графа $G=(X,U)$ и определения его хроматического числа можно применить алгоритм Дж. Магу, который подробно изложен в теме 5 (структурный анализ графов).

Раскраска графа с помощью алгоритма Дж. Магу выполняется в два этапа.

На первом этапе определяются подмножества вершин графа G , которые можно раскрасить одним цветом.

На втором этапе определяется хроматическое число графа $\chi(G)$.

Для определения подмножества вершин, которые можно раскрасить одним цветом, в графе $G=(X,U)$ с помощью алгоритма Дж. Магу находятся все максимальные пустые подграфы $G_1=(X_1,U_1)$, $G_2=(X_2,U_2)$, $G_3=(X_3,U_3), \dots, G_i=(X_i,U_i), \dots, G_k=(X_k,U_k)$

Очевидно, что вершины, принадлежащие одному подмножеству X_i можно раскрашивать одним цветом. Наибольшее число цветов p , необходимое для правильной раскраски вершин графа равно числу максимальных пустых подграфов данного графа.

Так как множества вершин $X_1, X_2, \dots, X_i, \dots, X_k$ максимальных пустых подграфов могут пересекаться, то число p , как правило, больше хроматического числа $\chi(G)$.

6.2. АЛГОРИТМ ОПРЕДЕЛЕНИЯ ХРОМАТИЧЕСКОГО ЧИСЛА $\chi(G)$ С ИСПОЛЬЗОВАНИЕМ МЕТОДА ДЖ. МАГУ

Шаг 1. Упорядочить все максимальные пустые подмножества $X_1, X_2, X_3, \dots, X_i, \dots, X_k$ графа G в порядке возрастания их кардинальных чисел $|X_i|$.

Шаг 2. Выбрать X_i , имеющее $\max |X_i|$.

Шаг 3. Присвоить цвет (допустим, синий) всем вершинам $x_i \in \max X_i$.

Шаг 4. Вычеркнуть из других подмножеств вершины, которым присвоен цвет.

Шаг 5. Исключить из дальнейшего рассмотрения подмножество X_i .

Шаг 6. Если семейство пустых подмножеств $X_1, X_2, X_3, \dots, X_k$ пусто, то перейти к ш.9, иначе – к ш.7.

Шаг 7. Из оставшихся подмножеств $X_1, X_2, X_3, \dots, X_k$ выбрать следующее $\max X_s$, где $s \in I : I = \{1, 2, 3, \dots, k\}$, и присвоить вершинам, входящим в него, цвет, который ещё не использовался.

Шаг 8. Для $\max X_s$ выполнить действия, описанные в шагах (4, 5, 6, 7), принимая $i=s$.

Шаг 9. Определить хроматическое число графа $\chi(G)$: — подсчитать полученное число подмножества вершин, «окрашенных» в разные цвета.

$\chi(G)$ = количество цветов, потребовавшихся для раскраски в ходе выполнения всех действий, описанных в шагах (1-8).

Пример. Найти раскраску для графа $G=(X,U)$, представленного на рисунке 6.1.

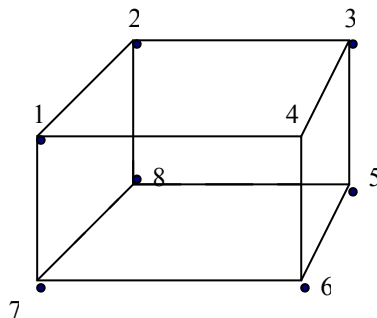


Рисунок 6.1

Решение.

1-й этап. Находим все максимальные пустые подграфы графа $G=(X,U)$ с помощью алгоритма Дж. Магу.

$$1. \text{ Составляем произведение } P_G = \prod_{j=1}^m \sum_{i=1}^n a_{ij} \cdot x_i,$$

где: a_{ij} - элемент матрицы инцидентий графа G ; $a_{ij} = (0,1)$; $x_i \in X$ -

новые образующие, подчиняющиеся условиям $x_i^2 = x_i$, $x_i + 1 = 1$, $2=1$ и

законам коммутативности, ассоциативности и дистрибутивности, на

основании которых преобразуем выражение для P_G к минимальной бесскобочной форме.

$$\begin{aligned} P_G &= (x_1 + x_2)(x_1 + x_4)(x_1 + x_7)(x_3 + x_2)(x_3 + x_4)(x_3 + x_5) \times \\ &\times (x_6 + x_4)(x_6 + x_5)(x_6 + x_7) \times (x_8 + x_2)(x_8 + x_5)(x_8 + x_7) = \\ &= (x_1 + x_2 x_4 x_7)(x_3 + x_2 x_4 x_5)(x_6 + x_4 x_5 x_7)(x_8 + x_2 x_5 x_7) = \dots = \\ &= x_1 x_3 x_6 x_8 + x_1 x_2 x_3 x_5 x_6 x_7 + x_1 x_3 x_4 x_5 x_7 x_8 + x_2 x_4 x_5 x_7 + x_1 x_2 x_4 x_5 x_6 x_8 + \\ &+ x_2 x_3 x_4 x_6 x_7 x_8. \end{aligned}$$

2. Для каждого слагаемого преобразованного выражения P_G запишем его дополнение до полной системы образующих $\{x_i\}$, $i=\overline{1,8}$

$$x_2 x_4 x_5 x_7; \quad x_4 x_8; \quad x_2 x_6; \quad x_1 x_3 x_6 x_8; \quad x_3 x_7; \quad x_1 x_5.$$

Получили полный обзор всех максимальных пустых подграфов графа G .

II-й этап.

Переходим к раскрашиванию вершин графа G

Будем кодировать цвета арабскими символами: 1,2,3,4,5,6.

Всем вершинам, принадлежащим одному максимальному пустому подграфу и ещё не раскрашенным, приписываем один цвет. Таким образом, для *правильной раскраски* вершин графа G требуется шесть цветов, т.е. $p=6$.

Граф G является 6-хроматическим.

III-й этап.

Определяем хроматическое число $\chi(G)$ графа G .

1. Упорядочим полученные множества вершин в порядке убывания их кардинальных чисел:

$$x_2 x_4 x_5 x_7; \quad x_1 x_3 x_6 x_8; \quad x_4 x_8; \quad x_2 x_6; \quad x_3 x_7; \quad x_1 x_5.$$

2. Припишем вершинам множества $x_2 x_4 x_5 x_7$ цвет «1» (цвет выбирается произвольно).

3. Удалим раскрашенные вершины из всех множеств и оставшиеся множества упорядочим в порядке убывания их мощности:

$$x_1 x_3 x_6 x_8; \quad x_8; \quad x_6; \quad x_3; \quad x_1.$$

4. Припишем вершинам множества $x_1 x_3 x_6 x_8$ цвет «2».

5. Удалим раскрашенные вершины из всех множеств:

получаем пустое множество. Это говорит о том, что все вершины графа G – раскрашены. Для раскраски потребовалось всего два цвета, т. е. хроматическое число $\chi(G)$ графа G равно двум: $\chi(G) = 2$.

Выполнить упражнения

1. Выполнить правильную раскраску рёбер графа, представленного на рисунке 6.3.

2. Вычислить хроматические числа для графов, представленных на рисунках 6.4, 6.5, используя метод Магу.

3. Сколько существует правильных раскрасок вершин куба, представленного на рисунке 6.1, в минимальное число цветов?

4. Разработать алгоритм правильной раскраски с минимальным количеством цветов для графа, представленного на рисунке 6.2, отличный от алгоритма, основанного на методе Магу,

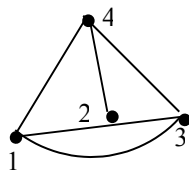


Рисунок 6.2

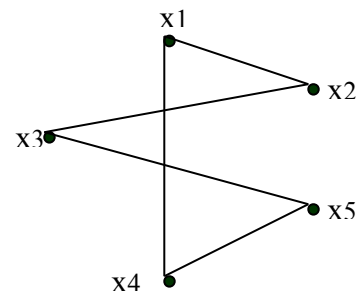


Рисунок 6.3

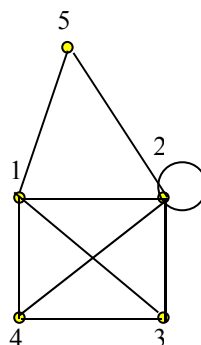


Рисунок 6.5

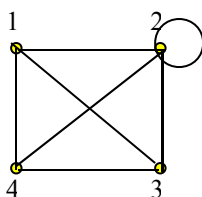


Рисунок 6.4

ТЕМА 7. Маршруты специального вида

К маршрутам специального вида относятся маршруты, относящиеся к понятию — *обходы графа*. Обход графа это — систематическое перечисление его вершин (и / или рёбер). Иными словами, обход графа — это маршрут, содержащий все рёбра или все вершины данного графа и обладающий определёнными свойствами. К обходам графа относятся гамильтоновы циклы и эйлеровы цепи и циклы. Названные обходы содержатся в графах, обладающих определённой характеристикой.

Для эйлеровых графов это сформулировано в следующей теореме:

Теорема Эйлера. 1. *Связный граф $G = (X, U)$ является эйлеровым тогда и только тогда, когда каждая вершина графа G имеет чётную степень.*

2. *Для существования в графе $G = (X, U)$ эйлеровой цепи, ведущей из x в y ($x, y \in X$), необходимо и достаточно, чтобы все отличные от x, y вершины G обладали чётными валентностями, а валентности x, y (в случае $x \neq y$) были нечётными.*

Необходимость и достаточность условий, сформулированных в теореме Эйлера, доказана. Для построения эйлерова цикла в эйлеровом графе пользуются алгоритмом, предложенным Флери.

7.1. АЛГОРИТМ ФЛЕРИ ПОСТРОЕНИЯ ЭЙЛЕРОВА ЦИКЛА

Рассматривая связный граф, все вершины которого удовлетворяют условиям теоремы Эйлера, строим эйлеровый цикл одним росчерком, придерживаясь следующих правил:

1. Отправляемся из произвольной вершины α ; каждое пройденное ребро зачёркиваем;

2. В процессе построения не прибегаем к исправлению уже построенной части траектории;

3. Никогда не идём по такому ребру, которое в рассматриваемый момент является перешейком, т.е. при удалении которого граф, образованный не зачёркнутыми рёбрами, распадается на две компоненты связности, имеющие хотя бы по одному ребру.

Легко показать, что, придерживаясь этих правил, действительно можно построить эйлеровый цикл.

Действительно, если построен маршрут до вершины x , то в оставшемся суграфе (из не зачёркнутых рёбер) имеются две вершины

нечётной валентности: x и α и в силу теоремы Эйлера имеется эйлерова цепь, начинающаяся из x .

Осталось показать, что всегда, придя в вершину x , имеем в распоряжении не пройденное ребро, не являющееся в данный момент перешейком. Действительно, если бы все инцидентные x рёбра были перешейками, то их было бы по крайней мере два; два перешейка вели бы в две не связанные друг с другом компоненты, каждая из которых содержит хотя бы одну вершину нечётной валентности. Но это невозможно, так как единственная вершина нечётной валентности, не считая x , является вершина α .

7.2. ОБХОДЫ ГРАФА ВИДА ГАМИЛЬТОНОВОЙ ЦЕПИ, ГАМИЛЬТОНОВА ЦИКЛА, ГАМИЛЬТОНОВА ПУТИ И КОНТУРА.

Гамильтоновой цепью в неориентированном графе называется цепь, проходящая через каждую его вершину один и только один раз.

Гамильтоновым циклом в неориентированном графе называется цикл, проходящий через каждую вершину один и только один раз за исключением начальной вершины, которая совпадает с конечной.

Гамильтоновым путём в ориентированном графе называется путь $S(x_0, x_1, \dots, x_n)$, проходящий через все вершины графа, притом каждая вершина проходится только один раз.

Гамильтоновым контуром называется контур $M(x_0, x_1, \dots, x_n, x_0)$ в ориентированном графе $G = (X, U)$, если он проходит через все вершины графа, притом каждая вершина проходится только один раз.

Существует несколько интерпретаций задачи о гамильтоновых циклах, имеющих ряд приложений в экономике и исследовании операций.

Рассмотрим одну из них: - **задача о бродячем торговце** (задача о ком-мивояжёре). Район, который должен посетить коммивояжер, содержит

определенное количество городов. Расстояния между ними известны, и нужно найти кратчайшую дорогу, проходящую через все пункты и возвращающуюся в исходный пункт. Решение данной задачи связано с нахождением гамильтонового цикла (контура) наименьшей длины в графе, интерпретирующем все возможные маршруты коммивояжера.

Сформулирован целый ряд достаточных условий существования гамильтоновых цепей, циклов, путей и контуров. Приведем некоторые из них без доказательства.

1. Теорема Кёнига. В полном конечном графе всегда существует гамильтонов путь.

2. Если в графе $G(X,U)$ с n вершинами для любой пары вершин x_i и $x_j \in X$ справедливо неравенство

$$m(x_i) + m(x_j) \geq n-1,$$

где $m(x_i)$, $m(x_j)$ - степени вершин x_i и x_j , то граф $G=(X,U)$ имеет гамильтонову цепь.

Несмотря на сходство в определении эйлерова и гамильтонова циклов, соответствующие теории для этих понятий имеют мало общего. Критерий существования для эйлеровых циклов установлен просто, для гамильтоновых циклов никакого общего правила неизвестно. Более того, иногда даже для конкретных графов бывает трудно решить, можно ли найти такой цикл. Дж.Литтлом для решения задачи коммивояжера был предложен простой рекуррентный алгоритм, основанный на идее метода ветвей и границ.

Рассмотрим его.

На содержательном уровне задача коммивояжера формулируется следующим образом. Для n городов, соединенных некоторыми путями, задана матрица C порядка n ($\dim C = n \times n$), для которой $c_{ij} \in C$ определяет стоимость (расстояние, время) пути из города i в город j .

Требуется указать путь, включающий посещение всех городов, который имеет минимальное значение суммарной стоимости.

Произвольное решение представляет собой сумму n слагаемых, каждое из которых определяется элементом матрицы C в соответствии с принятым порядком посещения:

$$Z = c_{i_1 i_2} + c_{i_2 i_3} + c_{i_3 i_4} + \dots + c_{i_{n-1} i_n} + c_{i_n i_1}$$

Оптимальным решением будет перестановка, минимизирующая эту сумму, т. е.

$$c_{i_1 i_2} + c_{i_2 i_3} + c_{i_3 i_4} + \dots + c_{i_{n-1} i_n} + c_{i_n i_1} \rightarrow \min_{u \in U},$$

где $u = (i_1, i_2, \dots, i_n)$ — перестановка для множества индексов

$I = \{1, 2, \dots, n\}$, $U = \{u\}$ — множество всех перестановок ($U = n!$).

В общем случае для некоторых пар i, j непосредственный переход от i к j может быть запрещен. Тогда элемент c_{ij} в матрице C полагается равным специальному значению — бесконечности: $c_{ij} = \infty$.

Если существуют какое-то конечное решение задачи, то маршрут, соответствующий оптимальному решению, не содержит дуги с бесконечным значением. Поэтому в дальнейшем будем полагать, что некоторые элементы матрицы C могут быть равными ∞ . Кроме того, положим, что все диагональные элементы матрицы C также равны ∞ .

Дж. Литтлом был предложен простой рекуррентный алгоритм решения задачи коммивояжера. В основе этого алгоритма лежит последовательное преобразование составленных определенным образом матриц, приводящее каждый раз к одной из трех стандартных возможностей.

Сначала по данным решаемой задачи составляется исходная матрица. Затем эта матрица обрабатывается по определенной схеме, что приводит к более простым вариантам, также задаваемым в виде матриц. Повторное применение стандартных приемов к каждому из этих вариантов продолжается до тех пор, пока не будет получено окончательное решение задачи. Анализ каждой из матриц приводит к одной из трех возможностей.

1. Получению решения, когда решение находится непосредственно по исходной матрице, если задача достаточно проста.

2. Исключению матрицы из дальнейшего рассмотрения, когда можно показать, что из нее не следует решение задачи.

3. Ветвлению, состоящему в том, что решаемая задача приводится к рассмотрению двух вариантов более простых задач.

На каждом шаге описываемого алгоритма задача включает n городов, причем из n шагов маршрута - k шагов могут быть уже установлены, и нужно выбрать оптимальным образом оставшиеся $n - k$ шагов. Для всех возможных маршрутов необходимо задать значение Y , представляющее собой нижнюю границу стоимости всех возможных решений задачи, включая оптимальное.

Особенность алгоритма Литтла состоит в использовании определенной эвристической схемы, позволяющей разумно построить эту нижнюю границу, стремясь сделать ее как можно больше.

Таким образом, задача характеризуется оставшимся числом $n - k$ неизвестных шагов маршрута и нижней границей Y стоимости решения задачи. Кроме того, можно считать, что известно, по крайней мере, одно решение исходной задачи. Обозначим через Z стоимость лучшего из уже найденных решений. Отметим, что при инициализации алгоритма Z может быть положено равным бесконечности, хотя более разумным подходом будет использование какого-либо приближенного алгоритма для

(достаточно быстрого) определения некоторого приближенного значения Z .

Возможны случаи:

1. Если $n-k = 2$, то осталось не более двух шагов маршрута, и решение находится сразу. Если его стоимость меньше Z , то Z принимается равным этому новому значению, а решение считается лучшим из известных решений. Поэтому в литературе текущее значение Z называют *рекордом*;

2. Если Y больше или равно Z , то происходит *отсечение* и рассматриваемая задача исключается, поскольку представленные в ней *маршруты* не могут привести к решениям, лучшим, чем уже известное;

3. Если не имеет места ни одна из перечисленных выше ситуаций, то происходит *ветвление*, и вместо рассматриваемой задачи составляются две:

а) в первой из них определенным образом выбирается переход от i к j , в результате чего нижняя граница стоимости решений может возрасти;

б) во второй запрещается переход из i в j (элемент c_{ij} полагается равным ∞), в результате чего нижняя граница стоимости решений также может возрасти.

Таким образом, получаемые задачи характеризуются возрастающей нижней границей и (или) большим числом установленных шагов маршрута. Кроме того, для каждой последующей задачи число возможных маршрутов меньше, чем для предыдущей, и на некотором этапе достигается такое состояние, когда маршрут определен полностью.

Ситуации, когда решение получается сразу или задача исключается, очевидны. Особенность алгоритма заключается в ветвлении, суть которого составляют понятия *приведения* матрицы и *выбора* дуги.

Приведение матрицы C основано на том факте, что если из любой строки или любого столбца матрицы вычитается константа, то оптимальное решение не меняется, а стоимость оптимального решения отличается от исходного в точности на количество, вычтенное из строки или столбца. Поэтому, если произведено вычитание такое, что каждый столбец и строка содержат нуль и все c_{ij} неотрицательны, то общая вычтенная сумма будет нижней границей стоимости любого решения. Матрицу стоимостей, полученную после такого вычитания, будем называть *приведенной*, т. е. такая матрица не допускает дальнейшего приведения.

Пусть C' — матрица стоимостей задачи, рассматриваемой на текущем шаге алгоритма. Рассмотрим случай 3 (ветвление). После того как фиксирован переход из i в j , нужно модифицировать матрицу C' . Обозначим матрицу новой задачи через $C' - Y$. В этой матрице нужно исключить переходы из i во все другие города, кроме j , и переходы в j из всех других городов, кроме i . Для этого положим все элементы строки i и столбца j матрицы, за исключением c_{ij} , равными ∞ .

Поскольку обход всех городов при единственном посещении каждого из них и окончания маршрута в исходной точке не может включать в себя одновременно дуги (i, j) и (j, i) , необходимо положить

$$c_{ij} = \infty.$$

Кроме того, если ранее построенный частичный обход содержал пути

$$i_1 \rightarrow i_2 \rightarrow \dots \rightarrow i_u \text{ и } j_1 \rightarrow j_2 \rightarrow \dots \rightarrow j_v, \text{ причём } i = i_u \text{ и } j = j_1,$$

то нужно запретить в последующем выбор дуги (j_v, i_1) , полагая соответствующий элемент матрицы, равным ∞ . Это позволит избежать повторного посещения городов до окончания обхода. Так как эти

запрещения могут привести к устранению ряда нулей в матрице C' , то не исключена возможность дальнейшего приведения матрицы $C' - Y$ и в результате этого получения новой большей границы для решений, связанных с этой матрицей.

Пусть переход из i в j запрещен. В этом случае также не исключена возможность дальнейшего приведения матрицы и вызванного этим возрастания нижней границы стоимости решений.

Выбор дуги (i, j) должен быть таким, чтобы максимально увеличить нижнюю границу и, возможно, исключить из рассмотрения ряд задач без дальнейшего ветвления. Чтобы достигнуть этого, просматриваются все возможные дуги (i, j) с нулевым весом и выбирается такая, чтобы сумма двух последовательных приводящих констант (по строке и по столбцу) была максимальной.

Алгоритм заканчивает работу, когда список задач, допускающих ветвление, исчерпан. Оптимальным является текущее рекордное решение.

7.3. ПРИМЕР РЕШЕНИЯ ЗАДАЧИ КОММИВОЯЖЕРА С ПОМОЩЬЮ АЛГОРИТМА ЛИТТЛА

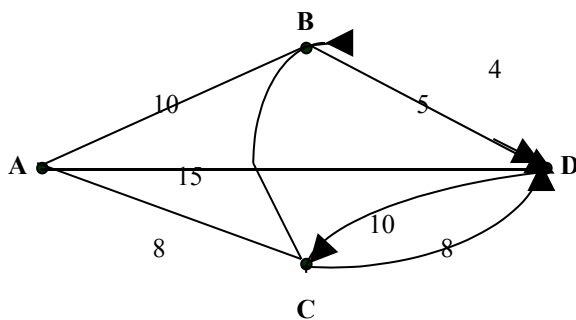


Рисунок 7.1

Требуется решить задачу коммивояжера для городов A, B, C, D , связанных сетью дорог, представленной графом на рисунке 7.1. Исходная матрица расстояний R имеет вид:

	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	$\min(i)$
<i>A</i>	∞	10	8	15	8
<i>B</i>	10	∞	9	5	5
<i>C</i>	8	9	∞	10	8
<i>D</i>	15	4	8	∞	4
					25 \Rightarrow

R =

Сумма

значений $\min(i) = 25$
(коэффициент приведения по
строкам)

Шаг 1. Приведём матрицу R к матрице R^1 , последовательно вычитая из всех элементов r_{ij} минимальное значение соответствующей строки:

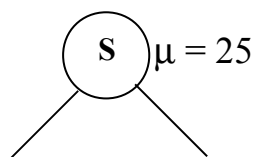
	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	
<i>A</i>	∞	2	0	7	
<i>B</i>	5	∞	4	0	
<i>C</i>	0	1	∞	2	
<i>D</i>	11	0	4	∞	
$\min(j)$	0	0	0	0	0 \Rightarrow

 $R^{(1)} =$

сумма значений $\min(j) = 0$ (коэффициент приведения по столбцам)

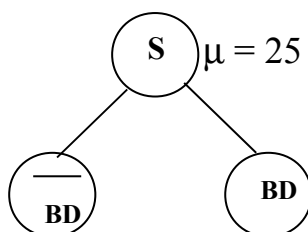
Шаг 2. Приведём матрицу $R^{(1)}$ к матрице $R^{(2)}$, последовательно вычитая из всех элементов $r_{ij} \in R^{(1)}$ минимальное значение соответствующего столбца. Так как в данном примере минимальное значение элементов в каждом столбце равно «0», то матрица $R^{(2)} = R^{(1)}$.

Шаг 3. Вычисляем сумму μ коэффициентов приведения матрицы R по строкам и столбцам: $\mu = 25 + 0 = 25$. Обозначим вершину корня дерева решений символом S и присвоим ему оценку равную μ , т.е. 25.



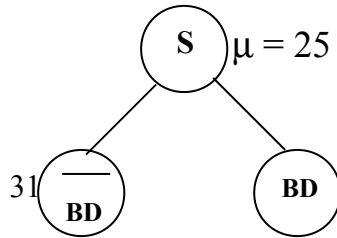
Шаг 4. В матрице $R^{(2)}$ вычислим «оценки» для каждого элемента r_{ij} , значение которого равно «0». Эта оценка складывается из суммы минимальных значений элементов в строке и столбце, на пересечении которых находится данный «нулевой» элемент. Так для элементов r_{AD} , r_{BD} , r_{CA} , r_{DB} эти оценки соответственно равны: 6, 6, 6, 5.

Шаг 5. Выбираем элемент с наибольшим значением оценки. Так как имеем несколько одинаковых наибольших значений, то выбираем любое из них, например, r_{BD} . Далее рассматриваем две альтернативы: а) маршрут BD включаем в решение, т.е. в искомый гамильтонов цикл; б) маршрут BD не включаем в решение (в гамильтонов цикл).



Если маршрут BD не включается в решение (альтернатива «а»), он обозначается как \overline{BD} . Оценка этого решения $\mu(\overline{BD})$ складывается из оценки родительской вершины (в данном случае это вершина S дерева решений) и оценки элемента r_{BD} матрицы $R^{(2)}$ (оценка нуля):

$$\text{— } \mu(\overline{BD}) = 25 + 6 = 31.$$



Для вычисления оценки альтернативы б) переходим к шагу 6.

Шаг 6. Вычёркивая из $R^{(2)}$ строку B и столбец D получаем матрицу $R^{(3)}$.

	A	B	C	D
A	∞	2	0	7
B	5	∞	4	0
C	0	1	∞	2
D	11	0	4	∞

$R^{(2)} =$



		<i>A</i>	<i>B</i>	<i>C</i>	
	<i>A</i>	∞	2	0	
	<i>C</i>	0	1	∞	
	<i>D</i>	11	0	4	

 $\mathbf{R}^{(3)} =$

По матрице $\mathbf{R}^{(3)}$ видно, что в графе имеется маршрут DB (элемент $r_{DB}=0$). Для исключения этого маршрута присвоим элементу r_{DB} значение ∞ . Выполним операцию приведения $\mathbf{R}^{(3)}$ матрицы по строкам и столбцам аналогично тому, как это выполнялось в ш.1-ш.3 и перейдём к матрице $\mathbf{R}^{(4)}$. Суммарное значение коэффициентов приведения матрицы $\mathbf{R}^{(3)}$ равно: $\mu=4+1=5$.

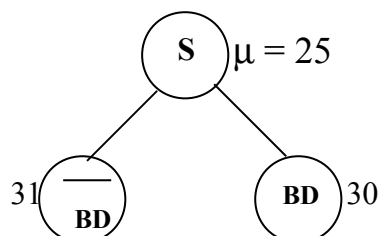
		<i>A</i>	<i>B</i>	<i>C</i>	$\min(i)$
	<i>A</i>	∞	2	0	0
	<i>C</i>	0	1	∞	0
	<i>D</i>	11	∞	4	4
$\min(j)$		0	1	0	

 $\mathbf{R}^{(3)} =$

	<i>A</i>	<i>B</i>	<i>C</i>
<i>A</i>	∞	1	0
<i>C</i>	0	0	∞
<i>D</i>	7	∞	0

 $\mathbf{R}^{(4)} =$

Шаг 7. Вычисляем значение оценки для вершины **BD**:
 $\mu_{BD} = \mu_S + \mu = 25 + 5 = 30$.



Выбираем маршрут **BD**, так как он имеет наименьшее значение оценки μ .

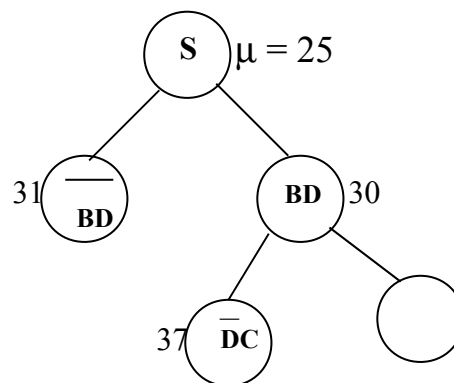
Шаг 8. В матрице $R^{(4)}$ вычислим «оценки» для каждого элемента r_{ij} , значение которого равно «0» (см. шаг 4). Так для элементов r_{DC} , r_{CB} , r_{CA} , r_{AC} эти оценки соответственно равны: 7, 1, 7, 1.

Шаг 9. Выбираем маршрут **DC** и вычисляем оценки для альтернатив **DC** и \overline{DC} (см. шаг 5). Оценка альтернативы « \overline{DC} » равна $30 + 7 = 37$.

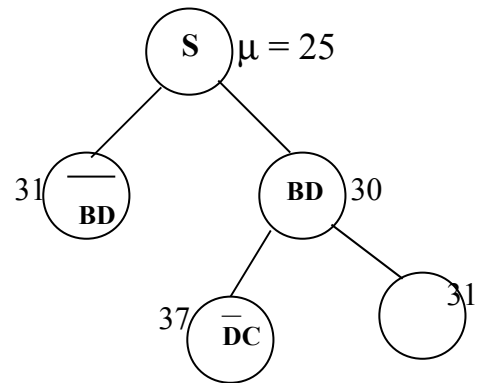
Для вычисления оценки альтернативы «**DC**» переходим к матрице $R^{(5)}$, вычеркнув из $R^{(4)}$ строку **D** и столбец **C**.

	A	B	C
A	∞	1	0
C	0	0	∞
D	7	∞	0

$R^{(4)} =$



	<i>A</i>	<i>B</i>	$\min(i)$	$R^{(5)} =$
<i>A</i>	∞	1	1	
<i>C</i>	0	0	0	
$\min(j)$	0	0		



Коэффициент приведения матрицы $R^{(5)}$ равен «1». Отсюда оценка альтернативы DC равна $30+1=31$.

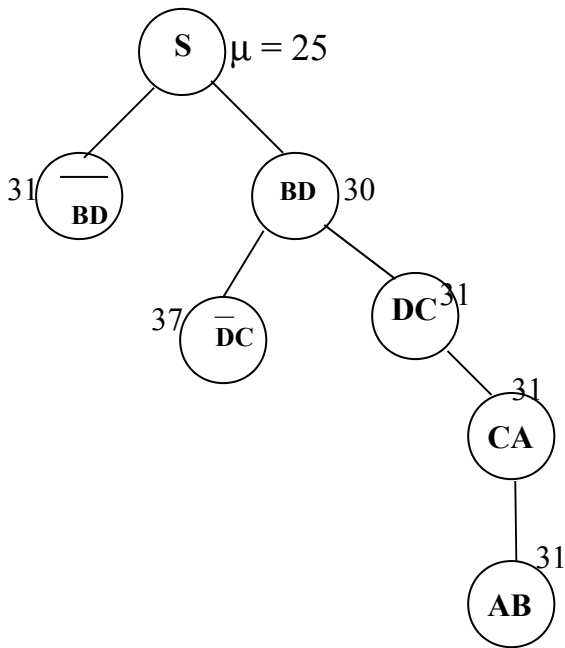
Шаг 10. Выполнив процедуру приведения матрицы $R^{(5)}$, переходим к матрице $R^{(6)}$.

	<i>A</i>	<i>B</i>
<i>A</i>	∞	0
<i>C</i>	0	0

$R^{(6)} =$

Значения элементов матрицы $R^{(6)}$ показывают, что для получения окончательного решения следует включить (безальтернативно) маршруты АВ и СА. Оценки для маршрутов АВ и СА равны «31», так как коэффициент приведения матрицы $R^{(6)}$ равен «0».

Построенное дерево решений показывает, что **наименьший**



гамильтонов цикл в данном графе (рисунок 7.1) имеет длину $l = 31$ и проходит по рёбрам (дугам) графа: BD, DC, CA, AB.

Рисунок 7.1

Упражнения

1. Определить имеют ли пятиугольник и пятигранник-пирамида с петлями в некоторых вершинах эйлеров цикл (эйлерову цепь)?
2. Имеют ли графы на рисунке 7.2 гамильтоновы циклы, цепи?

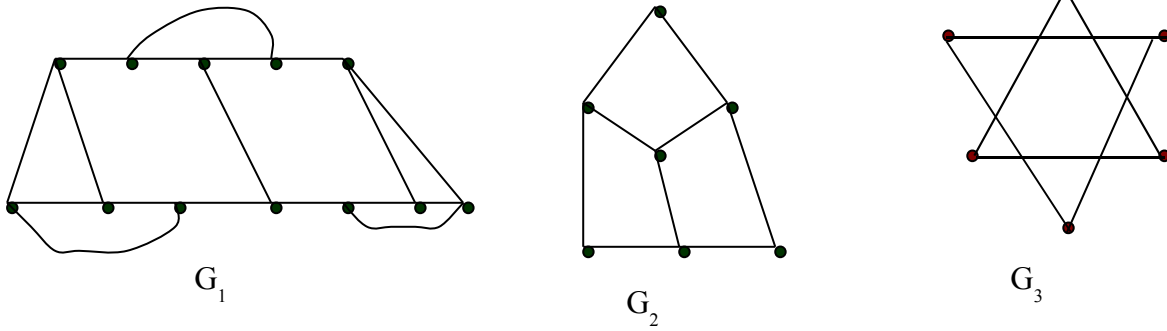


Рисунок 7.2

3. Описать алгоритм Флери для построения эйлера цикла в графе G_4 , представленного на рисунке 7.3.

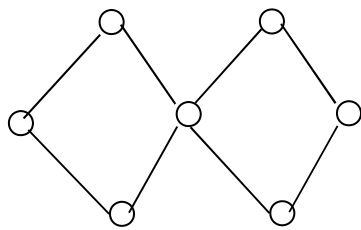


Рисунок 7.3

ТЕМА 8. Двудольные графы

8.1. ОСНОВНЫЕ ПОЛОЖЕНИЯ

Обыкновенный граф $L=(X,U)$ называется *двудольным* (или *биграфом*), если множество его вершин X можно представить в виде двух непересекающихся подмножеств X' и X'' так, чтобы никакие вершины одного и того же подмножества не были смежны, т.е.

$$X=X' \cup X''; X' \cap X'' = \emptyset \text{ и}$$

$$\forall x, y \in X [\widehat{x}\widetilde{y} \in U \Rightarrow (x \in X' \& y \in X'') \vee (x \in X'' \& y \in X')].$$

Часто **двудольный** граф записывают в виде (X', X'', U) .

Матрица смежности **двудольного** графа полностью определяется своей прямоугольной подматрицей, строки которой отвечают вершинам множества X' , а столбцы – X'' .

Двудольный граф $K_m=(X, Y, W)$, в котором $|X|=|Y|=|W|=m \geq 1$ и никакие два ребра не смежны, называется *паросочетанием*. Отображение Δ , которое каждой вершине множества X относит вершину множества Y здесь является взаимно однозначным соответствием между этими множествами.

Одной из важных в прикладном отношении задач теории графов является *задача нахождения наибольшего паросочетания*, которая формулируется следующим образом: для данного графа L найти наибольшее натуральное число $m=\pi(L)$, при котором существует паросочетание K_m , являющееся частью L . Если $L=(X', X'', W)$ – двудольный граф, то под его паросочетанием понимается часть

$K_m=(Y', Y'', W')$, удовлетворяющая условию $Y' \in X' \& Y'' \in X''$. Найти $\pi(L)$ это значит выяснить, какое наибольшее количество вершин

множества X' можно взаимно однозначно отобразить в X'' при помощи рёбер из W .

Теорема Кёнига- Холла: Все множество X' биграфа (X, X', U) можно взаимно однозначно отобразить в X'' при помощи рёбер U тогда и только тогда, когда

$$\forall A \subseteq X' (|\Delta A| \geq |A|).$$

Здесь ΔA – подмножество вершин множества X'' , смежных с вершинами из A .

Свойством, лежащим в основе определения двудольного графа, может обладать любой, не только обыкновенный граф. Именно, граф $L(X, U; P)$ называется *бихроматическим* (или *двудольным*), если множество X его вершин можно разбить на два непересекающихся подмножества

X' и X'' так, чтобы две вершины одного и того же подмножества не были бы смежны (т.е. раскрасить все вершины графа $L(X, U; P)$ двумя цветами так, чтобы смежные вершины имели разные цвета).

Очевидно, что граф является бихроматическим тогда и только тогда, когда он не содержит петель, а его скелет есть двудольный граф.

8.2. ВЕНГЕРСКИЙ АЛГОРИТМ НАХОЖДЕНИЯ МАКСИМАЛЬНОГО ПАРОСОЧЕТАНИЯ В ДВУДОЛЬНОМ ГРАФЕ

Венгерский алгоритм (Эгервари 1931, Кун 1953) предназначен для решения так называемых *задач о назначении*, которые в свою очередь в своем простейшем виде сводятся к нахождению наибольшего паросочетания у заданного двудольного графа.

Проиллюстрируем венгерский алгоритм на следующем примере.

Пусть имеется девять иностранных групп туристов T_1, \dots, T_9 , причем гости группы T_1 говорят на языке А, гости T_2 – на Б, T_3 – на В, T_4 – на А, T_5 – на Б, T_6 – на Г, T_7 – на Е, T_8 – на Г, T_9 – на Д; в свою очередь бюро «Интурист» располагает в данный момент десятью свободными переводчиками Π_1, \dots, Π_{10} , владеющими такими иностранными языками: Π_1 – языком (А,В); Π_2 – (А,Г); Π_3 – (А); Π_4 – (Б,В,Е); Π_5 – (А,В); Π_6 – (А), Π_7 – (Г,Д); Π_8 – (Б,Д); Π_9 – (Б,Г,Е); Π_{10} – (А,Б). Необходимо взаимно однозначно прикрепить переводчиков к группам, чтобы в первую очередь было обслужено, возможно большее число групп.

Построим двудольный граф $L = (X_1, X_2, U)$ (рисунок 8.1), в котором вершинам множества X_1 соответствуют переводчики $\Pi_1, \Pi_2, \dots, \Pi_{10}$, вершинам X_2 – группы туристов T_1, T_2, \dots, T_9 . Смежность вершины $\Pi_i \in X_1$ с вершиной $T_j \in X_2$ означает владение i -го переводчика языком j -ой группы. Требуется найти в L паросочетание с наибольшим количеством $\pi(L)$ ребер.

Теорема Кёнига - Холла позволяет вычислить наибольшее значение для $\pi(L)$:

$$\pi(L) = |X_1| - \max(|\Delta A| - |A|).$$

Здесь ΔA – подмножество вершин множества X_2 , смежных с вершинами из $A \subseteq X_1$. Для решения данной задачи представим граф

$L = (X_1, X_2, U)$ его матрицей смежности $B = (b_{ij})$, $i = \overline{1,10}$; $j = \overline{1,9}$ (рисунок 8.2).

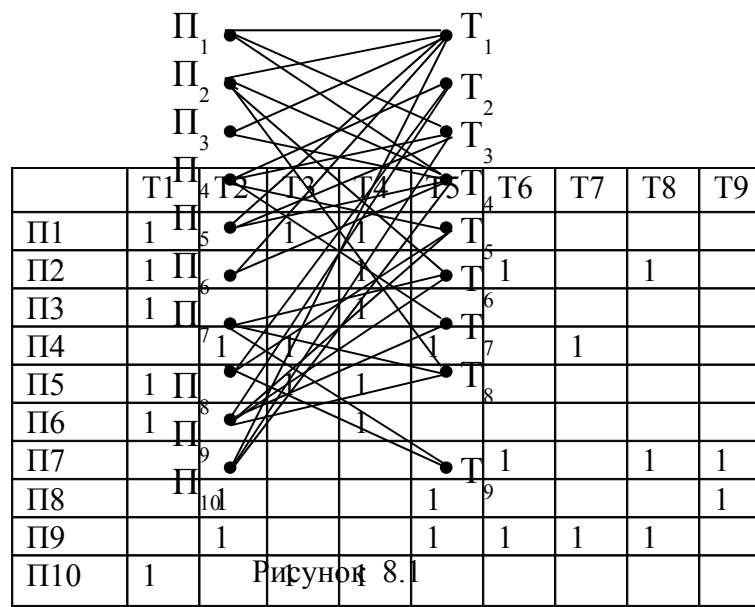


Рисунок 8.1

$$B =$$

Рисунок 8.2.

Строки матрицы B соответствуют элементам множества X_1 (переводчики), столбцы – элементам множества X_2 (туристы).

Элементы $b_{ij} = \{1, 0\}$; $b_{ij} = 1$ – если i -й переводчик может обслуживать j -ю группу туристов, $b_{ij} = 0$ – в противном случае.

Алгоритм решения состоит в следующем. Пусть некоторое паросочетание $K=(Y_1, Y_2, V)$ уже построено (в начальный момент это пустое паросочетание, $V=\emptyset$). Ребра графа L , принадлежащие V назовем «сильными», а ребра из $U \setminus V$ – «слабыми». Вершину, инцидентную *сильному* ребру будем называть *насыщенной*, а вершину, не инцидентную *сильному* ребру, – *ненасыщенной*. Под «*чередующейся цепью*» будем понимать простую цепь, в которой слабые рёбра строго чередуются с сильными; такую цепь назовем *тонкой*. Примем, что её длина отрицательна, если начальная и конечная вершины – обе ненасыщенные.

Ясно, что при наличии у графа L с заданным паросочетанием K хотя бы одной тонкой чередующейся цепи Q можно вместо K построить новое паросочетание K' , содержащее на одно ребро больше. Для этого надо все *слабые* ребра цепи Q сделать *сильными*, а *сильные* – *слабыми*, не трогая рёбер вне Q . Иначе говоря, надо удалить из множества V все те рёбра, которые принадлежат цепи Q , и к остатку добавить рёбра Q , принадлежащие $U \setminus V$.

V. В свою очередь, если у графа L с паросочетанием K' опять есть тонкая чередующаяся цепь, то аналогично предыдущему можно получить паросочетание K'' , имеющее уже на два ребра больше исходного.

Оказывается, что если на каком-то этапе тонких чередующихся цепей больше нет, то полученное паросочетание K^* – наибольшее, т.е. содержит ровно $\pi(L)$ ребер.

Процедура поиска тонкой чередующейся цепи в графе с заданным паросочетанием состоит в следующем: выбираем в X_1 любую ненасыщенную вершину и строим из нее чередующуюся цепь, отмечая штрихом пройденные рёбра и не выбирая их в дальнейшем. Попав в уже пройденную вершину или такую вершину из X_1 , которая не инцидентна ни одному еще не пройденному слабому ребру, возвращаемся на один шаг, отмечаем ребро вторым штрихом и пытаемся проложить чередующуюся цепь иначе, и т.д. В результате процесс либо оборвется в ненасыщенной вершине множества X_2 , и мы получим искомую цепь (она состоит из рёбер, отмеченных ровно одним штрихом), либо этот процесс приведёт нас в исходную вершину, и тогда надо начать аналогичный поиск с другой ненасыщенной вершины множества X_1 .

В нашем конкретном примере всё решение выглядит следующим образом.

1. Задаём произвольно паросочетание $K = \{(P_3 T_1), (P_2 T_4)\}$.
2. Фиксируем насыщенные вершины – P_3, P_2 и T_1, T_4 .
3. Строим тонкую чередующуюся цепь Q : $P_1 - T_1 - P_3 - T_4 - P_2 - T_8$, которая начинается и заканчивается в ненасыщенных вершинах P_1 и T_8 и содержит сильные рёбра $(P_3 - T_1)$ и $(P_2 - T_4)$.

4. Переименуем рёбра построенной тонкой цепи: теперь рёбра $(P_3 - T_1)$ и $(P_2 - T_4)$ – слабые, а рёбра $(P_1 - T_1)$, $(P_3 - T_4)$, $(P_2 - T_8)$ соответственно

стали сильными, вершины Π_1, T_8 – насыщенные. Новое паросочетание K' содержит уже три ребра.

5. Находим в графе новые ненасыщенные вершины T_3 и Π_7 , которые смежны насыщенным концевым вершинам построенной тонкой цепи

$$\Pi_1 - T_1 - \Pi_3 - T_4 - \Pi_2 - T_8 .$$

6. Строим новую тонкую чередующуюся цепь Q : $T_3 - \Pi_1 - T_1 - \Pi_3 - T_4 - \Pi_2 - T_8 - \Pi_7$, которая содержит сильные рёбра $(\Pi_1 - T_1), (\Pi_3 - T_4), (\Pi_2 - T_8)$.

7. Переименуем рёбра построенной тонкой чередующейся цепи: теперь рёбра $(\Pi_1 - T_1), (\Pi_3 - T_4), (\Pi_2 - T_8)$ – слабые, а рёбра $(T_3 - \Pi_1), (T_1 - \Pi_3), (T_4 - \Pi_2), (T_8 - \Pi_7)$ – сильные. Новое паросочетание K'' содержит уже четыре ребра.

8. Присоединим к построенной тонкой чередующейся цепи $T_3 - \Pi_1 - T_1 - \Pi_3 - T_4 - \Pi_2 - T_8 - \Pi_7$ ещё два ребра: $(T_3 - \Pi_4)$ и $(\Pi_7 - T_6)$ и над новой чередующейся цепью $Q = \Pi_4 - T_3 - \Pi_1 - T_1 - \Pi_3 - T_4 - \Pi_2 - T_8 - \Pi_7 - T_6$ выполним преобразование, аналогичное тому, которое описано в п.п. 4,7, увеличив таким образом паросочетание K'' ещё на одно ребро. Новое паросочетание K^v включает сильные рёбра – $(\Pi_4, T_3), (\Pi_1, T_1), (\Pi_3, T_4), (\Pi_2, T_8), (\Pi_7, T_6)$.

Увеличить длину построенной цепи Q , проходящей через вершины $\Pi_4, T_3, \Pi_1, T_1, \Pi_3, T_4, \Pi_2, T_8, \Pi_7, T_6$, нельзя, так как среди ненасыщенных вершин из множества $X \setminus V$, принадлежащих разным подмножествам (X_1, X_2) , нет инцидентных концевым вершинам цепи Q .

Увеличить построенное паросочетание K^v можно добавлением к нему цепей длины, равной 1, т.е. рёбер (Π_8, T_9) и (Π_9, T_5) или (Π_9, T_7) и (Π_8, T_9) . Тогда количество рёбер $\pi(L)$ паросочетания K^v станет равно 7.

$$Q = \Pi_4, T_3, \Pi_1, T_1, \Pi_3, T_4, \Pi_2, T_8, \Pi_7, T_6, \Pi_9, T_5, \Pi_8, T_2 .$$

	T1+	T2*	T3+	T4+	T5+	T6*	T7	T8+	T9
Π1+	1+		1-	1					

П2+	1			1-		1		1+	
П3+	1-			1 +					
П4*		1	1+		1		1		
П5	1		1	1					
П6	1			1					
П7+						1+		1-	1
П8+		1-			1+				1
П9+		1			1-	1+	1	1	
П10	1			1					

Рисунок 8.3

На рисунке 8.3 показана матрица смежности B графа L , где знаком ‘+’ помечены насыщенные вершины графа и элементы b_{ij} , соответствующие сильному ребру, т.е. ребру паросочетания Kv , значком ‘*’ помечены концевые вершины чередующейся цепи Q , знаком ‘-’ помечены элементы b_{ij} , соответствующие слабому ребру, т.е. ребру, которое не входит в паросочетания Kv .

Из матрицы видно, что увеличить построенное паросочетание нельзя так как оставшиеся ненасыщенные вершины инцидентны разным рёбрам.

Однако согласно теореме Кёнига–Холла для данного графа $\pi(L) = 8$. Действительно, если мы в построенной тонкой чередующейся цепи Q ребро $(П_4, Т_3)$ заменим ребром $(П_5, Т_3)$, то сможем построить другую чередующуюся цепь Q , которая проходит через вершины $П_5, Т_3, П_1, Т_1, П_6, Т_4, П_2, Т_6, П_7, Т_8, П_9, Т_5$.

	Т1	Т2	Т3	Т4	Т5*	Т6	Т7	Т8	Т9
П1	1+		1-	1					
П2	1			1-		1+		1	
П3	1			1					
П4		1	1		1		1		
П5*	1		1+	1					
П6	1-			1+					
П7						1-		1+	1
П8		1			1				1
П9		1			1+	1	1	1-	
П10	1			1					

Рисунок 8.4

Из матрицы B (рисунок 8.4) видно, что увеличить тонкую чередующуюся цепь Q нельзя, но в графе остались ненасыщенные вершины, инцидентные одному и тому же рёбру, которые можно добавить к к искомому паросочетанию K^v . Это рёбра $\Pi_8, T_9; \Pi_4, T_2$.

В результате получаем наибольшее паросочетание K^v (рисунок 8.5), в которое входят рёбра: $(\Pi_5, T_3), (\Pi_1, T_1), (\Pi_6, T_4), (\Pi_2, T_6), (\Pi_7, T_8), (\Pi_9, T_5), (\Pi_8, T_9), (\Pi_4, T_2)$. На рисунке 8.5 рёбра паросочетания обозначены жирными линиями.

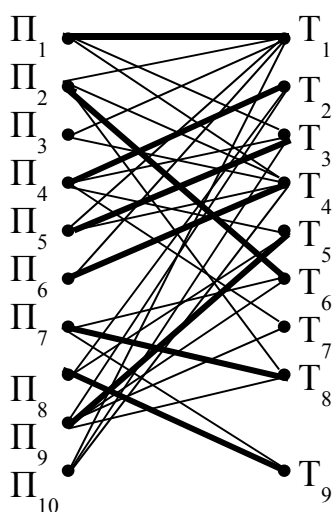
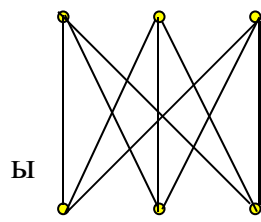


Рисунок 8.5

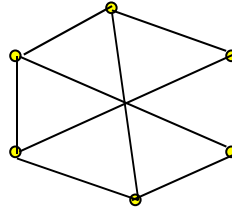
Упражнение.

1. Какой граф, из представленных на рисунке 8.6 является двудольным?

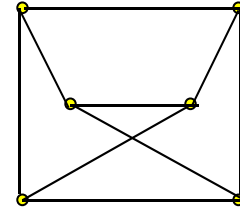


Ы

а)



б)



в)

2.
В

Рисунок 8.6

числить по формуле Кёнига-Холла число $\pi(L)$ для графов $K1=(V1, E1)$ и $K2=(V2, E2)$, представленных списком рёбер и вершин:

$$V1=\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}; V2=\{1, 2, 3, 4, 5, 6, 7\};$$

$$E1 =\{(1,8), (1,9), (2,8), (2,9), (2,12), (3,11), (4,8), (4,10), (5,11), (4, 12)\};$$

$$E2 =\{(1,5), (1,6), (2,6), (2,7), (3,5), (3,7), (4,6)\};$$

3. Разработать алгоритм для вычисления числа $\pi(L)$.

4. Построить наибольшее паросочетание в графе $K=(V, E)$, где $V=\{1,2,3,4,5,6,7,8,9,10,11,12\}$; $E=\{(1,8), (1,9), (2,8), (2,9), (2,12), (3,11), (4,8), (4,10), (5,11), (4, 12),(6,8),(6,10)\}$ /

ТЕМА 9. КОМПОНЕНТЫ СВЯЗНОСТИ

9.1. ОСНОВНЫЕ ОПРЕДЕЛЕНИЯ

Определение 1.

Граф $G(X,U)$ *связен*, если любые его две вершины можно соединить простой цепью.

Определение 2.

Подграф G' графа G называется *компонентой связности* графа G , если: 1) G' *связен*, 2) G' обладает свойством *максимальности*, т. е. если G'' — некоторый другой связный подграф графа G и $G' \subsetneq G''$, то графы G' и G'' совпадают.

Иными словами, *компонента связности* – это наибольший связный подграф данного графа.

На рисунке 9.1 показан граф $G=(X,U)$, содержащий две компоненты

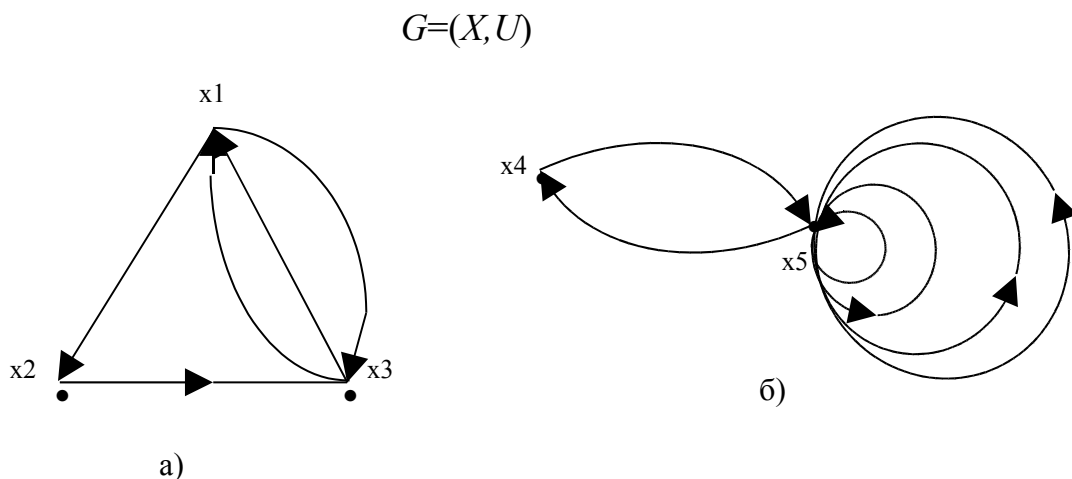


Рисунок 9.1

связности: G' с вершинами x_1, x_2, x_3 и G'' с вершинами x_4, x_5 .

9.2. СПОСОБ ОПРЕДЕЛЕНИЯ ЧИСЛА КОМПОНЕНТ СВЯЗНОСТИ

При помощи матриц смежности графов можно определить количество компонент связности. Для этого определим операцию элементарной склейки вершин в мультиграфе и выясним, как эта операция преобразует матрицу смежности.

Определение 3.

Мультиграф $G' = (X', U')$ получен из мультиграфа $G = (X, U)$ при помощи операции элементарной склейки вершин x_i и x_j из X , если

$$1) X' = (X \setminus (\{x_i\} \cup \{x_j\})) \cup \{z\}, \text{ где } z \notin X;$$

$$2) (x_m, x_l, n) \in U' \quad (m, l \neq i, j) \text{ тогда и только тогда, когда}$$

$$(x_m, x_l, n) \in U, (x_m, z, n) \in U' \quad (m \neq i, j) \text{ тогда и только тогда, когда}$$

$$\text{или } n = 2k \text{ и } (x_m, x_i, k) \in U \text{ или } n = 2k + 1 \text{ и } (x_m, x_j, k) \in U,$$

$$(z, z, n) \in U' \text{ тогда и только тогда, когда или } n = 4k \text{ и } (x_i, x_i, k) \in U,$$

$$\text{или } n = 4k + 1 \text{ и, } (x_j, x_j, k) \in U, \text{ или } n = 4k + 2 \text{ и } (x_i, x_j, k) \in U,$$

$$\text{или } n = 4k + 3 \text{ и } (x_j, x_i, k) \in U, (z, x_m, n) \in U', \text{ тогда и только}$$

G' , изображенного на рисунке 9.2, а. Очевидно, склеивание двух вершин, принадлежащих одной и той же компоненте связности, не изменяет количества компонент связности.

Обозначим $|X|$ через n . Перенумеруем множество вершин графа, полученного элементарной склейкой x_i и x_j следующим образом. Номера вершин, начиная с первого до $i-1$, сохраним. Номера вершин, начиная с $i+1$ до $j-1$, уменьшим на единицу, номера остальных вершин уменьшим на два, а вершине z присвоим номер $n-1$:

вершины $x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_{j-1}, x_{j+1}, \dots, x_n, z$

новые номера $1, \dots, i-1, i, \dots, j-2, j-1, \dots, n-2, n-1$

Обозначим через $f(k)$ ($k=1, 2, \dots, n-2$) старый номер вершины с новым

номером k . Тогда матрица $\|a'_{ik}\|$ нового графа строится по матрице

$\|a_{ik}\|$ первоначального графа по формулам:

$$a'_{lk} = a_{f(l)f(k)} \quad (l, k \leq n-2),$$

$$a'_{n-1k} = a_{if(k)} + a_{jf(k)} \quad (k \leq n-2),$$

$$a'_{l, n-1} = a_{f(l)i} + a_{f(l)j} \quad (l \leq n-2),$$

$$a'_{n-1, n-1} = a_{ii} + a_{ji} + a_{jj}$$

Например, матрицы смежности графов G и G' (см. рисунок 9.1а, б) есть соответственно

$$A = \begin{vmatrix} 0 & 2 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \end{vmatrix} \text{ и } A' = \begin{vmatrix} 0 & 1 \\ 1 & 4 \end{vmatrix}.$$

Заметим еще, что если в матрице смежности мультиграфа отличны от нуля лишь элементы, стоящие на главной диагонали, то число компонент связности равно числу вершин мультиграфа, т. е. размеру матрицы, так как такой мультиграф содержит только петли. На основании этого сформулируем алгоритм подсчета числа компонент связности по матрице мультиграфа.

9.3. АЛГОРИТМ ВЫЧИСЛЕНИЯ ЧИСЛА КОМПОНЕНТ СВЯЗНОСТИ ГРАФА

Шаг.1. Найти ненулевой элемент матрицы смежности, не стоящий на главной диагонали. Если он существует, перейти шагу.2, если нет, то перейти к шагу.3.

Шаг.2. Произвести над матрицей операцию, отвечающую склейке

вершин x_i и x_j перейти к шагу.1.

Шаг.3. Подсчитать количество p строк матрицы, содержащих ненулевые элементы на главной диагонали. Результат: количество компонент связности мультиграфа равно p .

Упражнения

1. В неорграфе $G = (X, U)$, в котором $|X| = 10$ и $X = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}\}$ и $U \neq \emptyset$, были склеены вершины x_3 и x_7 . В результате выполнения этой операции был получен граф $G' = (X', U')$ с новым числом вершин. Установить соответствие между номерами вершин графа G и графа G' .

2. Определить число компонент связности в графе G , заданного матрицей смежности $R=(r_{ij})$, элементы которой равны: $r_{1,2}=2$, $r_{2,3}=1$, $r_{4,11}=1$, $r_{5,9}=1$, $r_{6,7}=2$, $r_{4,10}=2$, $r_{11,10}=2$, $r_{7,8}=1$ (значения симметричных элементов матрицы R получить самостоятельно; значения неуказанных элементов приравнять «нулю»), с помощью алгоритма рассмотренного в данной теме.

3. Исследовать алгоритм определения количества компонент связности для классов орграфов, неориентированных графов и смешанных графов.

4. Разработать алгоритм определения числа компонент связности для компьютерной обработки.

ТЕМА 10. Оптимальные потоки в транспортных/информационных сетях

Теория транспортных и информационных сетей возникла при решении задач, связанных с организацией перевозки грузов и передачей информации. Тем не менее, понятие потока на транспортной сети и алгоритм нахождения потока наибольшей величины, критерий существования потока, насыщающего выходные дуги сети, оказались плодотворными для многих других прикладных и теоретических вопросов комбинаторного характера.

Введём основные понятия теории.

10.1. СЕТЬ (ТРАНСПОРТНАЯ, ИНФОРМАЦИОННАЯ)

Сетью называется ориентированный связный граф $G=(V,U)$, в котором отсутствуют петли и существует одна и только одна вершина $v_0 \in V$ такая, что множество $\Gamma^{-1}v_0 = \emptyset$, т.е. степень захода вершины v_0 равна 0, и существует одна и только одна вершина $z \in V$, такая, что множество $\Gamma z = \emptyset$, т.е. степень исхода вершины z равна 0. Вершина v_0 называется **исток** (вход) сети, вершина z называется – **сток** (выход) сети.

Каждой дуге $u \in U$ сети сопоставлено целое положительное число $c(u)$, которое называется пропускной способностью дуги.

10.2. ПОТОК В СЕТИ

Пусть U_v^- - множество дуг, заходящих в вершину v , а U_v^+ - множество дуг выходящих из вершины v .

Целочисленная неотрицательная функция $\varphi(u)$, определенная на множестве U дуг графа G , называется потоком, если она удовлетворяет следующим требованиям:

$$1. \quad \sum_{u \in U_v^-} \varphi(u) = \sum_{u \in U_v^+} \varphi(u) \quad (v \neq v_0, v \neq z)$$

для всех $v \in V \setminus \{v_0, z\}$;

$$2. \quad \varphi(u) \leq c(u) \quad \text{для всех } u \in U.$$

10.3. ЗАДАЧА О МАКСИМАЛЬНОМ ПОТОКЕ

Группу задач топологического анализа составляют задачи на распределение сетевых потоков. Это задача о максимальном потоке, в содержание которой оказываются взаимосвязанными топология сети, пропускные способности каналов связи или транспортных коммуникаций и распределение сетевых потоков. Подбирая определенным образом значение сетевых потоков и, исходя из заданного распределения пропускных способностей каналов, можно получить единственно возможное максимальное для заданной топологии значение общего суммарного потока в сети. В этом случае очевидно, ресурсы сетевой связи используются наиболее полно и эффективно.

Задача о максимальном потоке формулируется следующим образом: пусть задано исходное распределение потоков по дугам графа, отображающего топологическую структуру сети, а также пропускные способности дуг. Необходимо найти максимально возможное для данной сети значение суммарного потока между источниками и стоками, т. е.

определить, как увеличить поток, если он не достиг этого значения. Применительно к рассматриваемой задаче используется одно из важных положений теории потоков, которое сформулировано и доказано в виде теоремы Фордом и Фалкерсоном. Согласно этой теореме максимально возможное значение суммарного потока на конечных дугах равно минимальной пропускной способности выбранного разреза. При этом под пропускной способностью разреза понимается *сумма пропускных способностей дуг, образующих разрез*.

В символической форме записи соотношение, отражающее содержание теоремы Форда — Фалкерсона, - выглядит следующим образом:

$$\sum_{i \in B} \sum_{j \in \tilde{B}} \varphi(v_i, v_j) = \sum_{i \in B} \sum_{j \in \tilde{B}} c(v_i, v_j),$$

где $\varphi(v_i, v_j)$ — значение потока по дугам заданного графа; $c(v_i, v_j)$ — пропускная способность дуги; B — множество вершин подграфа, образующих разрез; \tilde{B} — дополнение множества B до множества V . Множество B вершин подграфа, образующих разрез всегда содержит вершину z (сток) и никогда не содержит вершину v_0 (исток). Разрез $R(G)$ сети включает множество дуг $U' \subset U$ сети, исходящих из вершин множества \tilde{B} и входящих в вершины множества B .

Доказательство теоремы Форда-Фалкерсона строится методом от противного на следующих предположениях: граф имеет две характерные вершины — *исток* и *сток*, а разрез $R(G)$ вершины графа делит на два взаимодополняющих множества B и \tilde{B} .

Допустим, что на графе задан максимальный поток Φ , а сток z не отделен от множества вершин B разрезом, т. е. $z \in B$. Тогда из определения разреза и при данном условии следует, что существует хотя бы один путь

из истока v_0 в сток — вершину z , для которого должны выполняться условия: — для прямых дуг пути $(v_0 - z)$: — $\varphi(v_i, v_j) < c(v_i, v_j)$;
для обратных дуг пути $(v_0 - z)$: — $\varphi(v_i, v_j) > 0$.

Это свидетельствует о том, что поток на сети не является максимальным и его величину можно увеличить, насыщая отдельные дуги по путям, идущим от истока к стоку.

Иначе, если задан разрез, то он своими дугами однозначно определяет максимально возможный, проходящий через них поток. Очевидно, разрез $(v_0—z)$ минимальной общей пропускной способностью дуг задает максимально возможный для данного графа поток от истока v_0 к стоку z .

10.4. АЛГОРИТМ ФОРДА — ФАЛКЕРСОНА.

Для транспортной/информационной сети важно найти такие минимальные разрезы и оценить соответствующие им значения максимального потока. Это можно осуществить с помощью алгоритма Форда — Фалкерсона.

Принцип, лежащий в основе алгоритма Форда — Фалкерсона, заключается в том, чтобы найти все возможные насыщенные пути (цепи), ведущие от v_0 к z . С этой целью последовательно, начиная с вершины x_0 , просматриваются сначала все смежные ей $\{v_i\}$ вершины. Из множества дуг $\{(v_0, v_i)\}$, соединяющих v_0 с $\{v_i\}$, выбирают одну, у которой значение потока ближе всех подходит к значению насыщения. Помечают вершину v_i знаком, показывающим, что она была просмотрена, и приписывают ей величину δ , на которую можно увеличить поток по дуге, ведущей в эту вершину. Затем просматривают все последующие смежные с v_i вершины $\{v_j\}$ и останавливаются на той, в которую ведёт дуга с потоком, ближайшим к значению насыщения. По этой дуге

переходят в соответствующую вершину v_j . Делают пометки вершин и идут далее в направлении вершины z . Если путь, на котором будут отмечены все пройденные вершины, приведёт в вершину z , то это говорит о том, что найден один из путей, наиболее близкий к насыщению. Нужно довести поток по нему до насыщения, увеличивая тем самым поток на графе. Значение, на которое можно увеличить поток, находится как минимальное δ из множества отмеченных значений в пройденных по данному пути вершинах. Для выяснения вопроса, является ли полученный таким образом поток максимальным или нет, необходимо просмотреть все другие возможные пути, ведущие от x_0 к z . Для этого необходимо возвратиться в x_0 и повторить описанные выше действия, но идти следует по еще не помеченным вершинам. В результате выполнения этих действий можно столкнуться с двумя вариантами: — 1) пройденный путь снова приводит от x_0 к z , т. е. удастся найти ненасыщенные пути и, значит, можно увеличить потоки на их дугах; 2) придя в некоторую вершину, обнаружить, что все смежные с ней вершины помечены и, следовательно, больше путей, ведущих к z нет. Это указывает на то, что на сети G получен максимальный поток.

Согласно теореме Форда и Фалкерсона значение полученного максимального потока можно вычислить, выделив дуги разреза $R(G)$ с минимальной пропускной способностью и просуммировав их пропускные способности.

10.5. ПРИМЕР РЕШЕНИЯ ЗАДАЧИ О МАКСИМАЛЬНОМ ПОТОКЕ С ПОМОЩЬЮ АЛГОРИТМА ФОРДА-ФАЛКЕРСОНА

Пусть задана сеть $G=(X,U)$ (рисунок 10.1), на дугах u_{ij} которой в скобках обозначены значения пропускных способностей дуг — (c_{ij}) .

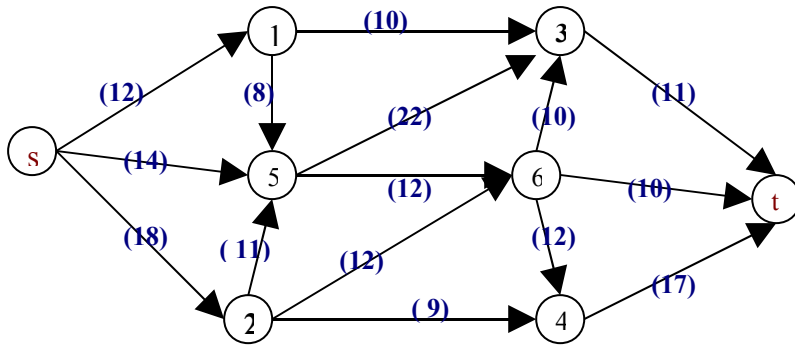


Рисунок .10.1.

1. Зададим на сети нулевой поток (на всех дугах величина потока φ_{ij} равна 0). Нулевой поток — это начальный допустимый поток на сети.

Значение потока на каждой дуге u_{ij} будем указывать за скобками пропускной способности дуги: — $(c_{ij}) \varphi_{ij}$. Значение потока, равное «0» не указываем.

2. Выбираем на сети (произвольно) путь, ведущий из вершины s в вершину t :

$$s \rightarrow 1 \rightarrow 5 \rightarrow 6 \rightarrow t.$$

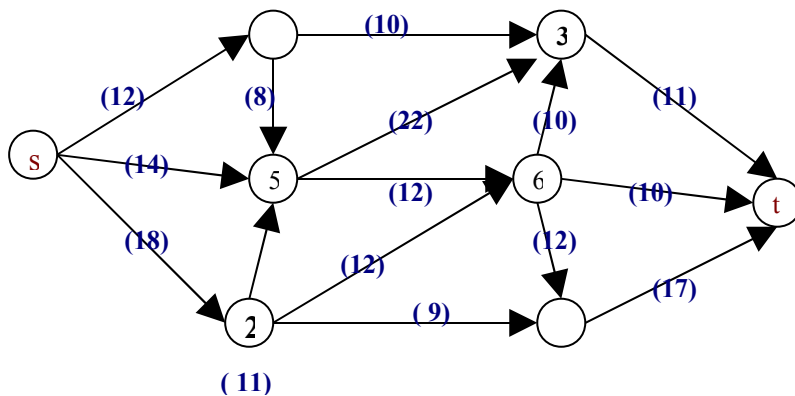


Рисунок 10.2.

3. Вычисляем значения $\{\delta_{ij} = c_{ij} - \varphi_{ij}\}$ на каждой дуге пути $s \rightarrow 1 \rightarrow 5 \rightarrow 6 \rightarrow t$ и выбираем минимальное из них. $\min \delta_{ij} = 8$ (так как $\varphi_{ij} = 0$

на всех дугах данного пути). На величину $\min \delta_{ij} = 8$ увеличиваем поток на каждой дуге пути $s \rightarrow 1 \rightarrow 5 \rightarrow 6 \rightarrow t$. Дуга $(1,5)$ становится «насыщенной», так как поток на ней стал равен $c_{1,5}$.

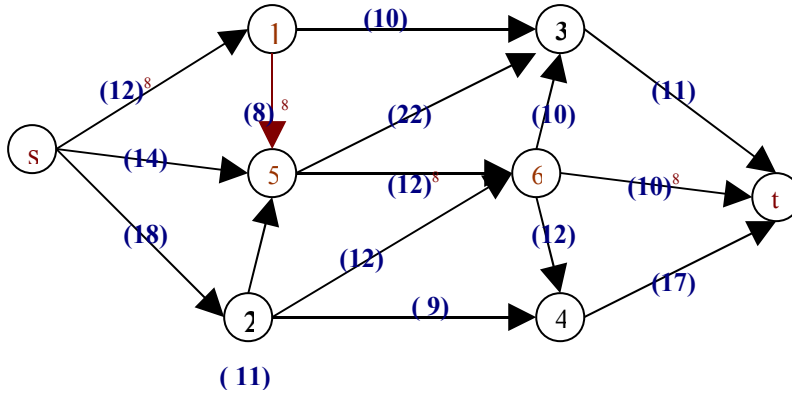


Рисунок 10.3

4. Выбираем следующий путь, ведущий из s в t : $s \rightarrow 1 \rightarrow 3 \rightarrow t$ (рисунок 10.4) и для него вычисляем значение $\min \delta_{ij}$ (аналогично тому, как это делалось в п.3). $\min \delta_{ij} = 4$.

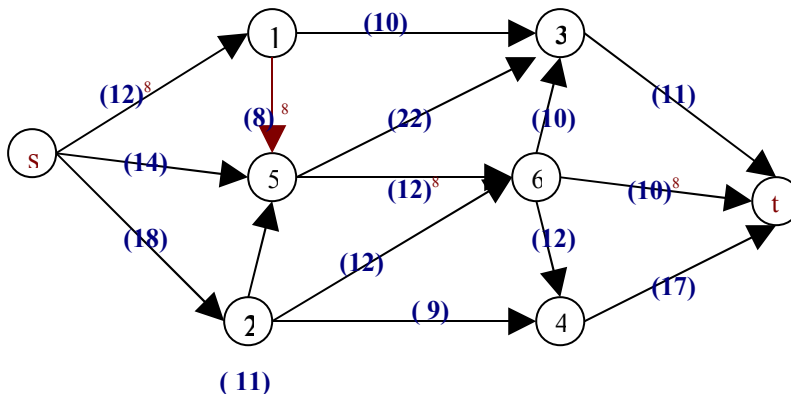


Рисунок 10.4

Увеличиваем величину потока на каждой дуге пути $s \rightarrow 1 \rightarrow 3 \rightarrow t$ на 4. Отмечаем насыщенную дугу $(s,1)$ (рисунок 10.5).

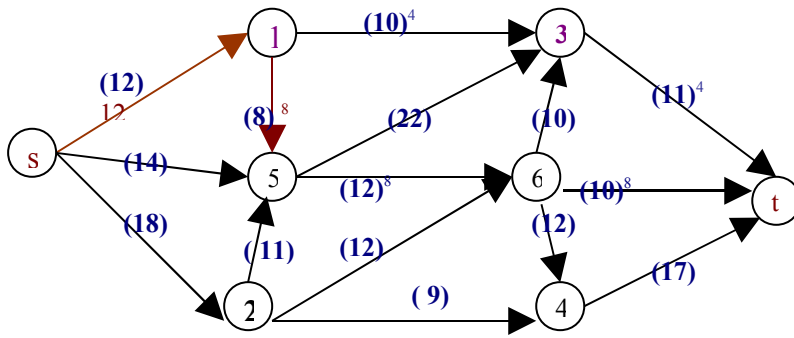


Рисунок 10.5.

5. Выбираем следующий путь, ведущий из s в t : $s \rightarrow 5 \rightarrow 6 \rightarrow t$ (рисунок 10.6).

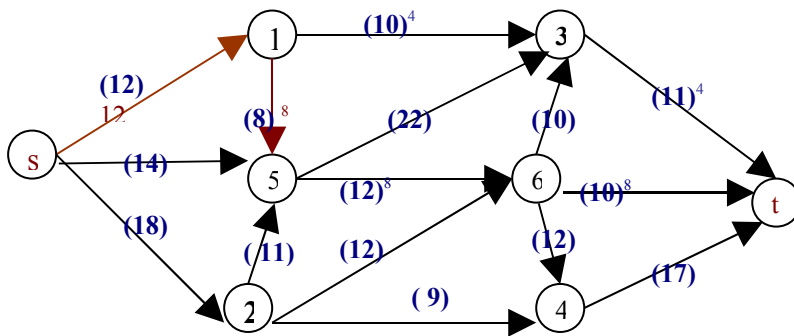


Рисунок 10.6.

Увеличиваем величину потока на каждой дуге пути $s \rightarrow 5 \rightarrow 6 \rightarrow t$ на 2, что отражено на рисунке 10.7. Насыщенная дуга (6,t).

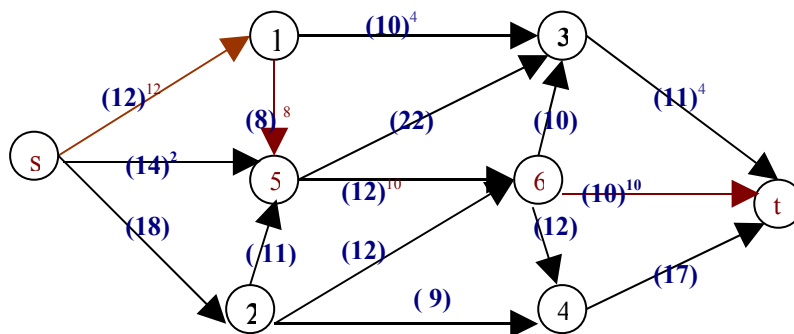


Рисунок 10.7.

6. Дальнейшие аналогичные

действия, связанные с увеличением потока на сети, отражены на рисунках 10.8 – 10.16. Все оставшиеся вычисления и выкладки предлагается провести самостоятельно в качестве упражнения.

Для однозначности, отметим лишь последовательность рассматриваемых путей: $s \rightarrow 5 \rightarrow 3 \rightarrow t$ (рисунки 10.8 – 10.10); $s \rightarrow 2 \rightarrow 6 \rightarrow 4 \rightarrow t$ (рисунки 10.11 – 10.12); $s \rightarrow 2 \rightarrow 4 \rightarrow t$ (рисунки 10.13 – 10.16).

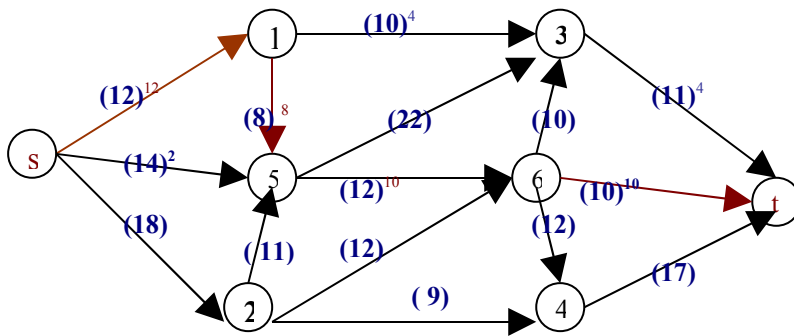


Рисунок 10.8.

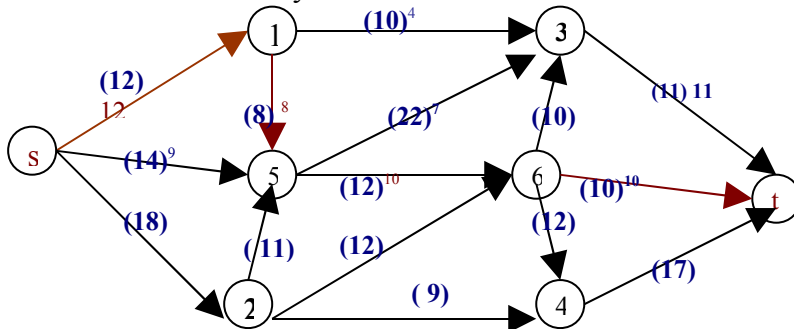


Рисунок 10.9.

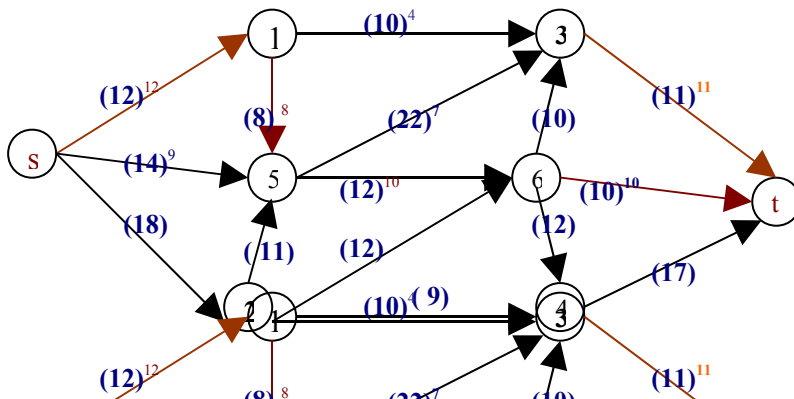


Рисунок 10.10.

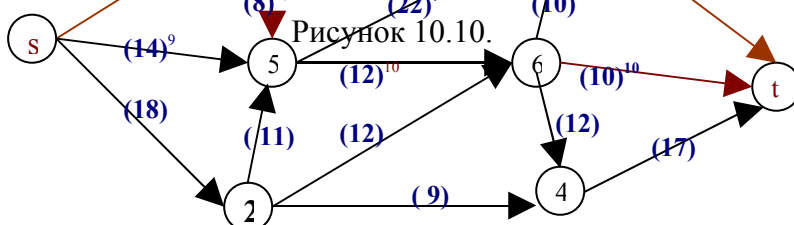


Рисунок 10.11.

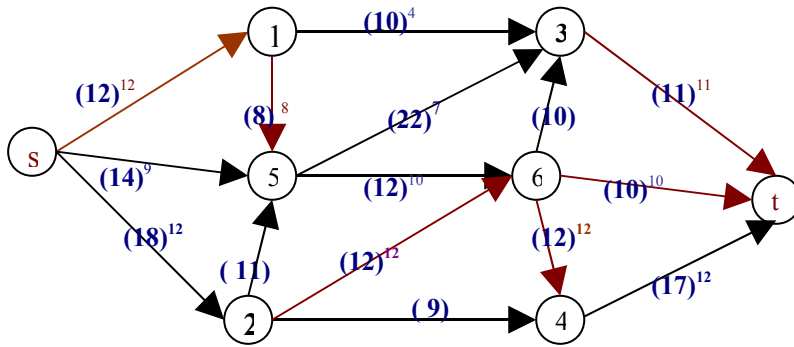


Рисунок 10.12.

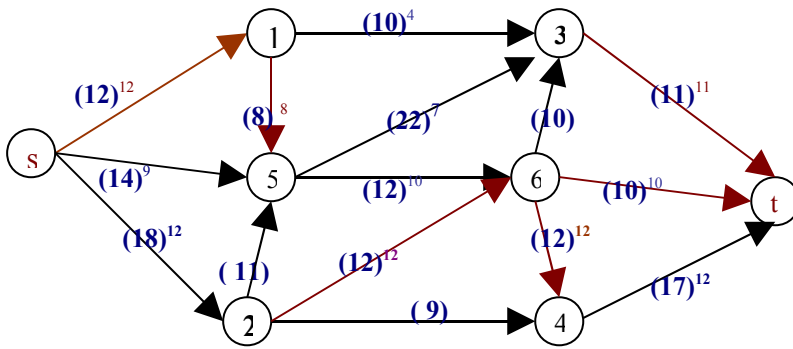


Рисунок 10.13.

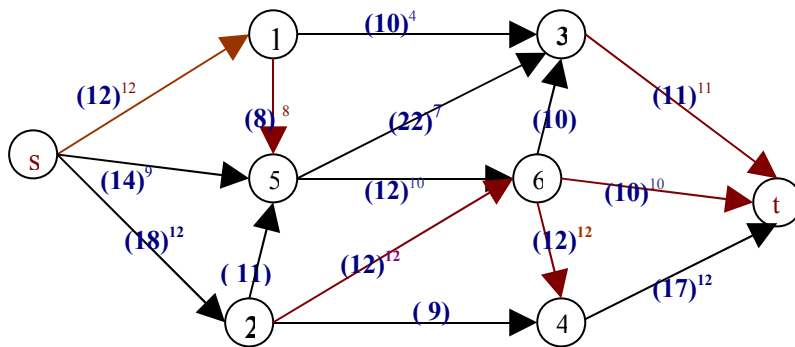


Рисунок 10.14.

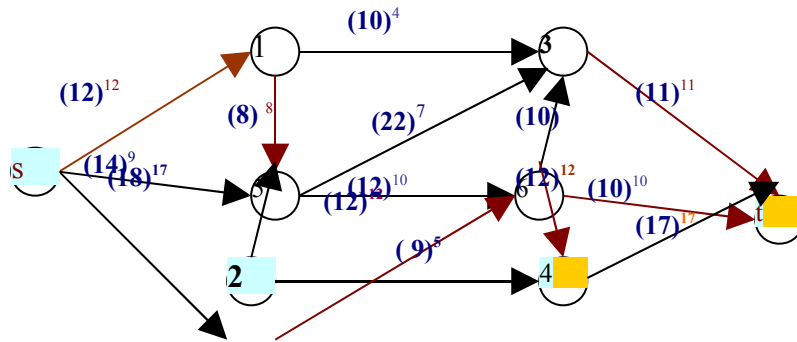


Рисунок 10.15.

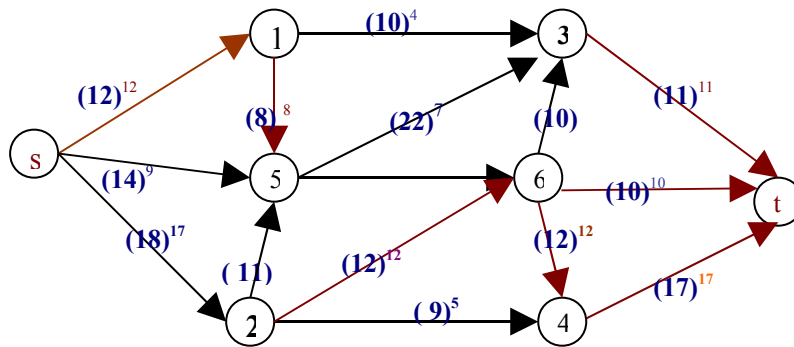


Рисунок 10.16.

7. Выполнив действия, связанные с увеличением допустимого потока в сети (рисунки 10.1 –10.16) от вершины s к вершине t , переходим к построению минимального разреза сети T .

8. Алгоритм нахождения минимального разреза сети

8.1. Процедура “пометок вершин”

- Начальное состояние: — все вершины не имеют пометок.
- Вершине s присваивается пометка.
- Всем вершинам $x_i \in \{\Gamma s\}$, для которых дуга (s, x_i) не насыщена:

—
 $c_{si} > \phi_{si}$ присваиваются пометки.

- Всем вершинам $x_k \in \{\Gamma x_i\}$, для которых дуга (x_i, x_k) не насыщена:

—
 $c_{ij} > \phi_{ij}$ присваиваются пометки.

В ходе присвоения пометок вершинам сети возможны две ситуации:—

1. Удалось присвоить пометку вершине t , из чего следует, что в сети есть путь от вершины s к вершине t , все дуги которого не насыщены. Следовательно, поток на сети может быть увеличен за счёт его увеличения на пути s, \dots, t (с помощью рассмотренного выше алгоритма).

2. Не удалось присвоить пометку вершине t . Следовательно, на сети получен максимальный поток и, для его вычисления, возможно, построить минимальный разрез.

На рисунке 10.17 приведён результат пометок вершин для рассматриваемой сети:

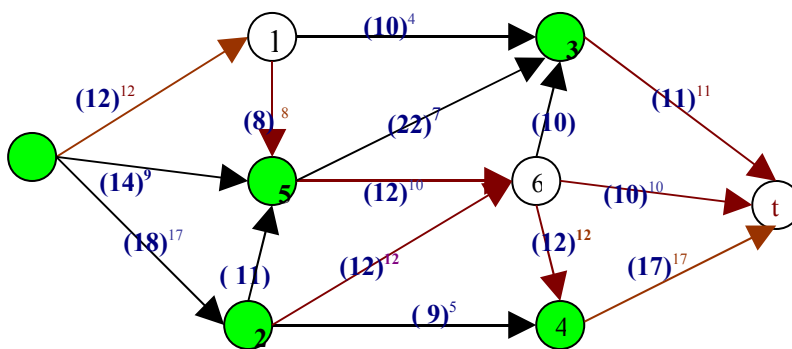


Рисунок 10.17.

8.2. Определение дуг минимального разреза

По результату алгоритма пометок, когда не возможно пометить вершину t (сток), определяем дуги минимального разреза: — это дуги, начала которых находятся в помеченных вершинах, а концы — в не помеченных вершинах.

В наше примере это дуги $u_{s,1}$; $u_{5,6}$; $u_{2,6}$; $u_{3,t}$; $u_{4,t}$ (см. рисунок 10.18).

Таким образом минимальный разрез данной сети $T = (u_{s,1}$; $u_{5,6}$; $u_{2,6}$; $u_{3,t}$; $u_{4,t})$.

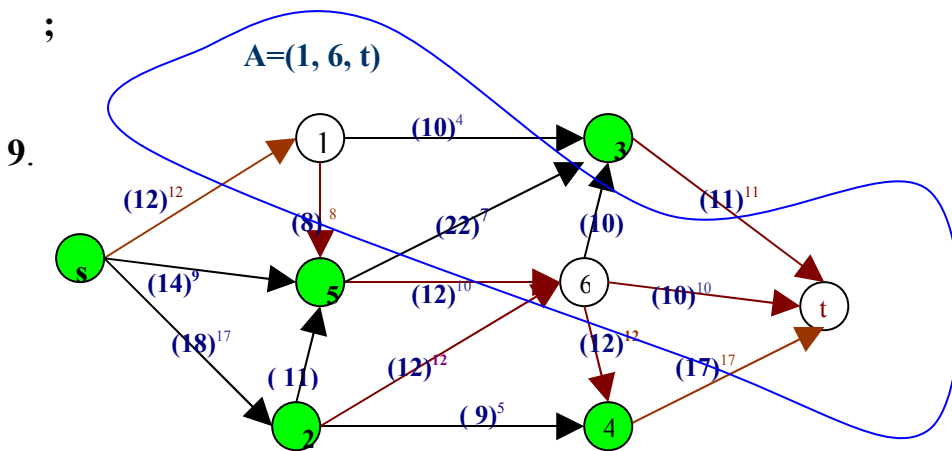


Рисунок 10.18.

Вычисление величины максимального потока Φ_{\max} :

$$\Phi_{\max} = c_{s,1} + c_{5,6} + c_{4,t} + c_{2,6} + c_{3,t} = 12 + 10 + 17 + 12 + 11 = 62.$$

Допустимый поток максимальной величины на заданной сети G — найден.

Упражнения

1. Для сети $G = (X, U, C)$, где $|X| = 6$, $|U| = 10$ и $U = \{(x_1, x_2), (x_1, x_3), (x_2, x_4), (x_2, x_5), (x_3, x_2), (x_3, x_5), (x_4, x_3), (x_5, x_4), (x_4, x_6), (x_5, x_6)\}$, определить: — является G насыщенной, т.е. в G задан максимальный

допустимый поток, или G ненасыщенная, если известно, что дуги сети (x_1, x_2) , (x_4, x_6) , (x_3, x_5) , (x_2, x_4) , (x_5, x_4) — насыщенные.

2. Определить максимальный поток в сети $G = (X, U, C)$, где $|X| = 6$, $|U| = 10$, пропускные способности дуг равны: — $c_{12} = 7$, $c_{13} = 5$, $c_{24} = 4$, $c_{25} = 10$, $c_{32} = 8$, $c_{35} = 6$, $c_{43} = 5$, $c_{45} = 3$, $c_{46} = 11$, $c_{56} = 6$.

3. Определить минимальный разрез в сети $G = (X, U, C)$, где $|X| = 6$, $U = \{(x_1, x_2), (x_1, x_3), (x_2, x_4), (x_2, x_5), (x_3, x_2), (x_3, x_5), (x_4, x_3), (x_4, x_5), (x_4, x_6), (x_5, x_6)\}$, если известно, что дуги сети (x_1, x_2) , (x_4, x_6) , (x_3, x_5) , (x_2, x_4) , (x_4, x_5) , (x_5, x_6) — насыщенные.

РАЗДЕЛ 2. ПЕРЕКЛЮЧАТЕЛЬНЫЕ ФУНКЦИИ И ИХ РАЗЛОЖЕНИЯ

2.1. ПЕРЕКЛЮЧАТЕЛЬНЫЕ ФУНКЦИИ И СПОСОБЫ ЗАДАНИЯ

Определение. *Переключательной функцией* (ПФ) $f(x_1, x_2, \dots, x_n)$ называется однозначное отображение вектора $X = (x_1, x_2, \dots, x_n)$, каждая переменная x_i ($x_i \in X$) которого принимает значение из множества $\{0, 1, \dots, k_i - 1\}$, $i = 1, 2, \dots, n$ в множество f , $f \in \{0, 1, \dots, k_f - 1\}$.

В случае, если $k_1 = k_2 = \dots = k_i = \dots = k_n = k_f = k$, получаем определение k -значной функции; в пределе, когда $k = 2$ – определение булевой функции.

ПФ можно задавать:

- табличным способом;
- теоретико-графовым;
- аналитическим способом.

Табличными способами являются одномерные и двумерные таблицы. Одномерная таблица, определяющая функцию $f(x_1, x_2, \dots, x_n)$, имеет размеры:

$$\prod_{i=1}^n k_i \times (n + 1),$$

где $\prod_{i=1}^n k_i$ – количество строк, которые взаимно однозначно соответствуют векторам $X = (x_1, x_2, \dots, x_n)$; $(n + 1)$ – количество столбцов; из них n столбцов взаимно однозначно соответствуют предметным переменным x_1, x_2, \dots, x_n , $(n + 1)$ -й столбец – функциональной переменной f .

Пример 2.1. Пусть $x_1 \in \{0, 1, 2\}$; $x_2 \in \{0, 1\}$; $x_3 \in \{0, 1\}$; $f \in \{0, 1, 2, 3\}$. Определим переключательную функцию с помощью одномерной таблицы.

Таблица 2.1

	x_1	x_2	x_3	$f(x_1, x_2, x_3)$
1	0	0	0	3
2	0	0	1	3
3	0	1	0	1
4	0	1	1	0
5	1	0	0	2
6	1	0	1	3
7	1	1	0	0
8	1	1	1	1
9	2	0	0	1
10	2	0	1	1
11	2	1	0	0
12	2	1	1	3

Мощность $P(x_1, x_2, \dots, x_n, f)$ переключательных функций

$f(x_1, x_2, \dots, x_n)$ равна:
$$P(x_1, x_2, \dots, x_n, f) = K_f \prod_{i=1}^n k_i .$$

Пусть:

если $\forall x_i \in \{0, 1\}$ – то для $i = \overline{1, 2}$

количество наборов: $2^2 = 4$,

количество (мощность) функций: $2^{2^2} = 16$.

Двумерная таблица, определяющая переключательную функцию $f(x_1, x_2, \dots, x_n)$, строится в результате разбиения переменных $X = (x_1, x_2, \dots, x_n)$ на два подмножества: подмножество строчных переменных X_a и подмножество столбцевых переменных X_b ; в клетке (i, j) записывается соответствующее значение функции $f(x_1, x_2, \dots, x_n)$. Такие таблицы называют **таблицами Вейча**.

Таблица Вейча для нашего примера:

Таблица 2.2

x_1	x_2x_3			
	00	01	10	11
0	3	3	1	0
1	2	3	0	1
2	1	1	0	3

При теоретико-графовом задании переключательной функции используются категории (классы):

- мографа $G^M(f)$ (модельный граф),
- гиперграфа $H_B(f)$,
- графа Кёнига $K(f)$ и
- гиперкуба $H(f)$.

Рассмотрим их.

Задание переключательной функции мографом

Задание переключательной функции мографом $G^M(f)$ связано с одномерной таблицей (см. таблицу 2.1).

Каждая вершина v_i мографа взаимно однозначно соответствует первичному терму $v_i \leftrightarrow x_i^{\sigma_i}, \sigma_i = 0, 1; f_j^{\sigma_j}, \sigma_j = 0, 1, 2$.

Взаимно однозначно идентифицируем строки одномерной таблицы $0\dots 00, 0\dots 01, \dots$ как $1, 2, 3, \dots, 12$.

Вершина f^k соединяется с вершиной x_i^k , если f^k и x_i^k входят в одинаковые наборы.

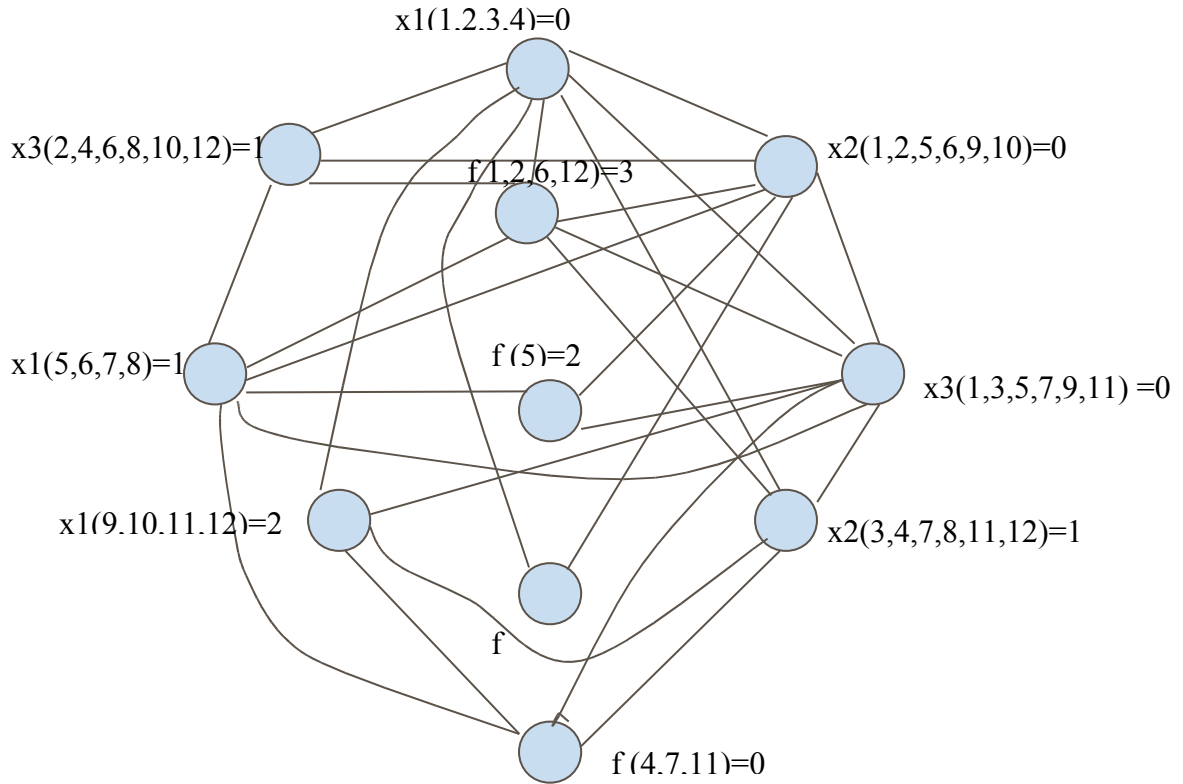


Рисунок 2.2

Задание переключательной функции гиперграфом

Гиперграф $H_B(f)$, определяющий функцию $f(x_1, x_2, \dots, x_n)$ по существу является диаграммой Эйлера.

Фрагмент гиперграфа $H_B(f)$, определяющий первые два значения рассматриваемой функции $f(x_1, x_2, x_3)$, представлен на рисунке 2.3.

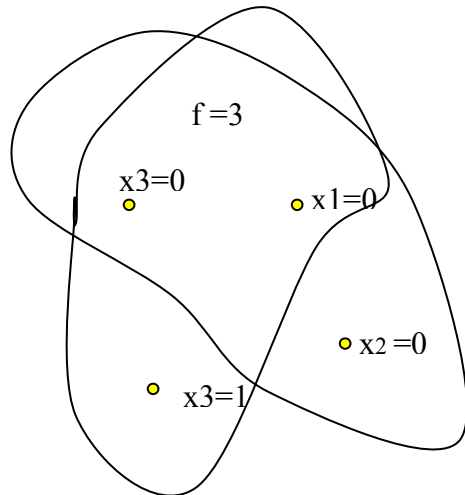


Рисунок 2.3

Очевидно, что при большом числе строк соответствующей одномерной таблицы, геометрическое представление гиперграфа функции не обеспечивает должной наглядности. В этом случае его целесообразнее представлять модифицированной матрицей инциденций.

Задание переключательной функции гиперкубом

Гиперкуб $H(f)$ – каждый вектор X функции $f(x)$ взаимно однозначно сопоставляется вершине гиперкуба, и две вершины смежны, если соответствующие им векторы отличаются в одном и только одном разряде ровно на 1.

Значение переменной называют *фазой*.

Каждая вершина гиперкуба взвешена значением функции $f(x_1, x_2, \dots, x_n)$.

В i -м ярусе, если счёт начинать с «0», будут векторы, сумма фаз переменных которых равна i . Для рассматриваемой функции $f(x_1, x_2, x_3)$

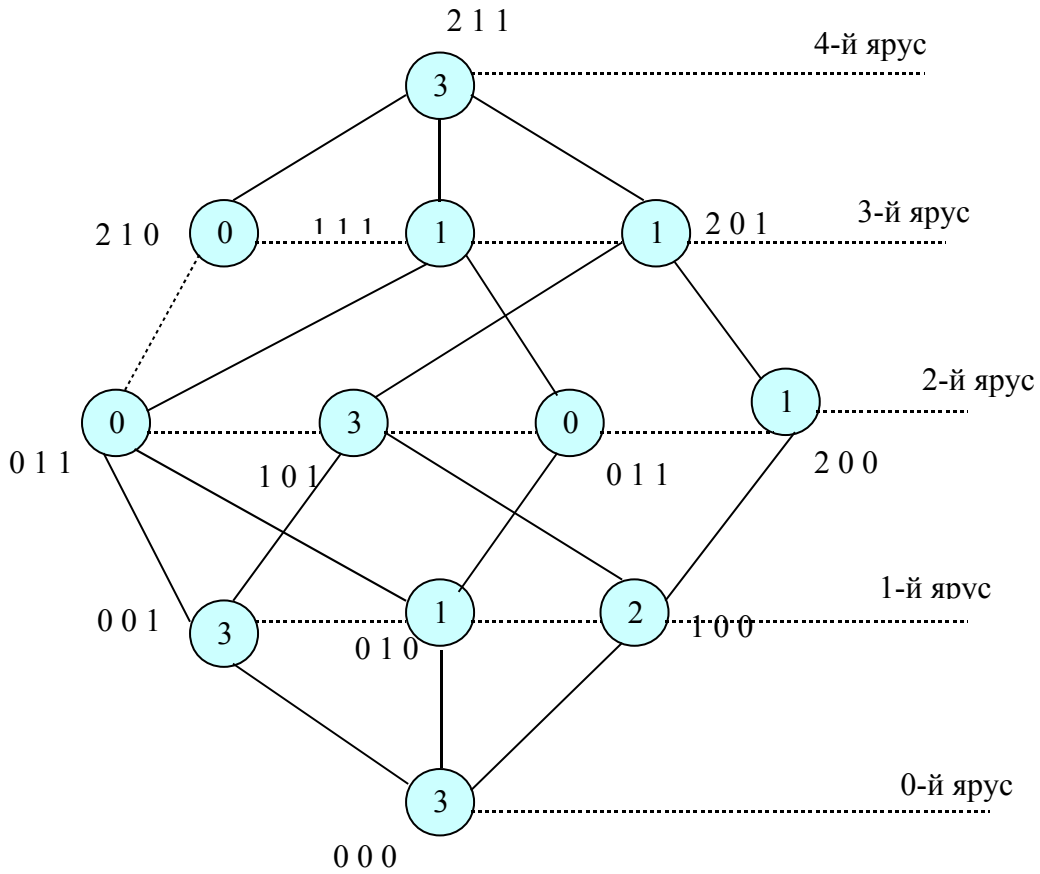


Рисунок 2.4

гиперкуб $H(f)$ представлен на рисунке 2.4.

Количество вершин в каждом ярусе можно подсчитать по формуле

$$\sum_i \left(n_j! \cdot \left(\prod_{\sum l_i=2} n(l_i)! \right)^{-1} \right)_j \tag{2.1}^*$$

(2.1) – количество размещений из n по l элементов с повторениями

Так, количество вершин во втором ярусе равно:

n_j – количество разрядов, в которых могут встречаться символы 1, 1; $n_j = 3$, $n(l_i) = 2$ (так как «1» встречается сразу в двух разрядах). Символ «2» может встречаться только в одном разряде – $n_j = 1$, $n(l_i) = 1$ ($n(l_i)$ – количество символов). Отсюда имеем:

$$3! \cdot (2!)^{-1} + 1! \cdot (1!)^{-1} = 3 + 1 = 4.$$

Рассчитаем количество вершин в первом ярусе. Первому ярусу соответствуют вершины, которые взвешены значениями функции $f(x_1, x_2, x_3) = \{3, 1, 2\}$.

Выпишем из таблицы (2.1) вектора, соответствующие этим значениям:

$$f(x_1, x_2, x_3) = 3 \quad \left\{ \begin{array}{l} (000), (001), (101), (211) \end{array} \right.$$

$$f(x_1, x_2, x_3) = 1 \quad \left\{ \begin{array}{l} (010), (111), (200), (201) \end{array} \right.$$

$$f(x_1, x_2, x_3) = 2 \quad \left\{ \begin{array}{l} (100) \end{array} \right.$$

Применяя формулу (2.1), учитываем, что вектора вершин смежных должны отличаться только в одном разряде и всего на «1», и сумма фаз переменных векторов равна i (i – номер фазы; фаза – это значение переменной).

Определяем данные для формулы (2.1):

$$\sum_i \left(n_j! \left(\prod n(l_i)! \right)^{-1} \right),$$

$i = 1$, $n_j = 3$ (так как кортеж имеет длину $l = 3$ – количество разрядов, в которых встречается «1»); каждый разряд содержит только «1», которое может повторяться $n(l_i)$ раз; $n(l_i) = 2$ («1» только в одном разряде).

Будем размещать символы 0, 0 в $n_j = 3$ разрядах. Тогда $n(l_i) = 2$. Теперь количество вершин определяется по формуле:

$$\sum_i \left(n_j! \left(\prod n(l_i)! \right)^{-1} \right) = \left(n_j! \left(\prod n(l_i)! \right)^{-1} \right) = 3! \cdot (2!)^{-1} = 3,$$

так как $i = 1$, то число вершин первого яруса равно 3 ($i = 1$, так как рассматриваем только одну комбинацию символов: «0, 0»).

Количество вершин третьего яруса:

- символ «2» – в одном разряде, $n_j = 1$, $n(l_i) = 1$;
- символ «0, 1» – в двух разрядах, $n_j = 2$, $n(l_i) = 2$;
- символ «1, 1, 1» – один раз;
- $\sum_{i=3} \left(n_2! \left(n(l_i)! \right)^{-1} \right) = 1! \cdot (1!)^{-1} + 2! \cdot (2!)^{-1} + 1 = 3$.

Задание переключательной функции графом Кёнига

Для задания ПФ графом Кёнига используется таблица Вейча

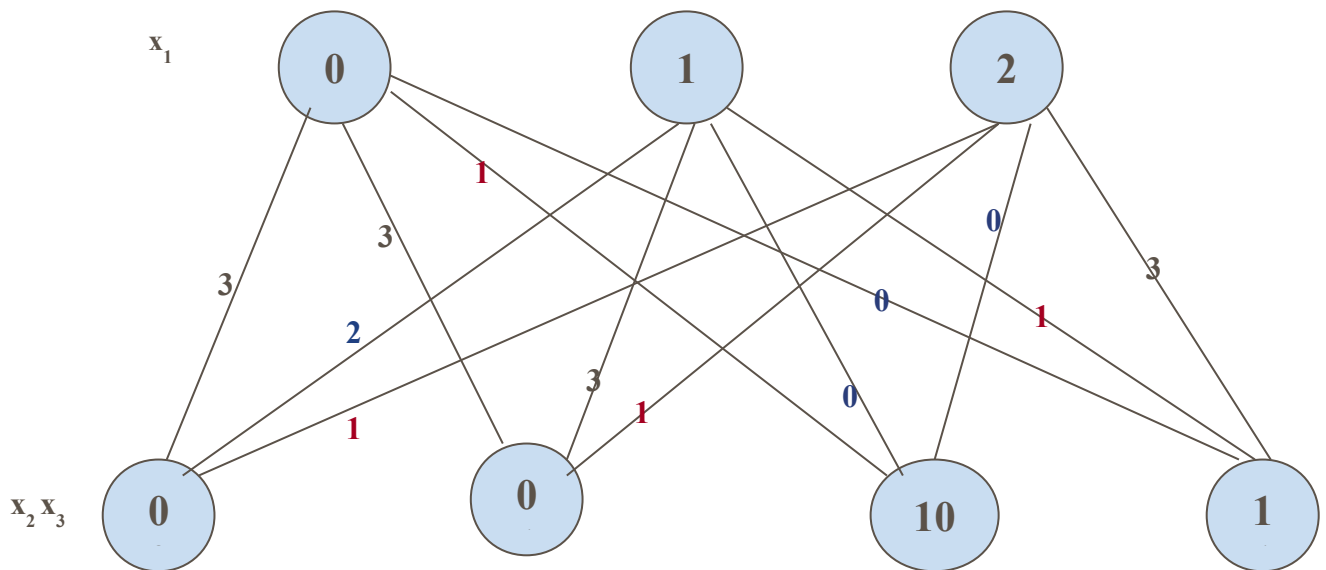


Рисунок 2.5

Каждая вершина яруса взаимно однозначно сопоставляется вектору, определяемому соответствующими переменными.

Ребро, соединяющее вершину одного яруса с вершиной другого, соответствует вектору $X = (x_1, x_2, \dots, x_n)$, который определяет значение функции $f(x_1, x_2, \dots, x_n)$. Для геометрического отображения значения функции $f(x_1, x_2, \dots, x_n)$

Каждому ребру поставим в соответствие число-значение, которое принимает $f(x_1, x_2, \dots, x_n)$ на соответствующем наборе значений переменных.

Для рассматриваемой функции $f(x_1, x_2, x_3)$ граф Кёнига представлен на рисунке 2.5.

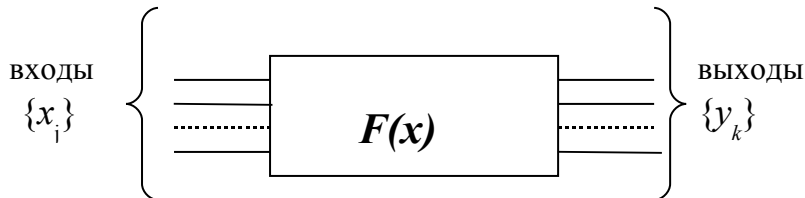
2.2. БУЛЕВЫ ФУНКЦИИ (БФ)

Определения.

1. **Булева функция** – это функция, аргументы которой и сама функция принимают значения на множестве $B = \{0, 1\}$ элементами

которого являются формальные символы 1 и 0, интерпретируемые как {«да» и «нет»}:

2. Булева функция – это математическая модель дискретных устройств переработки информации:



- На вход устройства подаётся одна комбинация, которой закодирована некоторая информация, на выходе получают другую комбинацию;

- Количество наборов комбинаций переменных – 2^n , где n – число

x	БФ(0)	БФ(x)	БФ(\bar{x})	БФ(1)
0	0	0	1	1
1	0	1	0	1
	f_0	f_1	f_2	f_3

переменных;

- Количество различных БФ – 2^{2^n} .

Булева функция иначе называется **логической функцией** или **переключательной функцией**. БФ можно задавать теми же способами, что и любую переключательную функцию.

2.3. АНАЛИТИЧЕСКОЕ ПРЕДСТАВЛЕНИЕ БУЛЕВЫХ ФУНКЦИЙ

Логическая функция может быть задана формулой, содержащей символы переменных, знаки операций и скобки, либо суперпозицией, содержащей вместо переменных функции, определяющие их.

Логическая функция может быть задана эквивалентными формулами:

Например: —

$$f_8: x_1 \downarrow x_2 \text{ (функция Пирса), } f_8 = \overline{x_1 \wedge x_2} \text{ или } f_8 = \overline{x_1 \vee x_2}$$

$$f_{14}: x_1 \mid x_2 \text{ (Функция Шеффера), или}$$

$$f_{14} = \overline{x_1} \wedge \overline{x_2} \vee x_1 \wedge \overline{x_2} \vee \overline{x_1} \wedge x_2 = \overline{x_1} \vee \overline{x_2}$$

Рассмотрим способы получения эквивалентных формул для представления булевых функций.

Переключательная функция называется монотонной, если при любом возрастании набора значения этой функции не убывают (наборы 0; 1 и 1; 0 несравнимы).

f_2 – немонотонна, так как на (1; 0) она равна «1», а на (1; 1) – «0»; f_3 – монотонна; f_5 – монотонна.

Пусть дано множество исходных функций (конечное или бесконечное) $\sum \{f_1, f_2, \dots, f_m\}$.

Символы переменных будем считать формулами глубины «0». Формула F имеет глубину $k + 1$, если F имеет вид

$$f_i(F_1, F_2, \dots, F_{n_i}),$$

где $f_i \in \Sigma$, n_i – число аргументов f_i , а F_1, F_2, \dots, F_{n_i} – формулы, максимальная из глубин которых равна k . F_1, F_2, \dots, F_{n_i} – подформулы F , f_i – внешняя или главная операция формулы F . Все подформулы формул F_1, F_2, \dots, F_{n_i} также называются подформулами F .

Пример. 1). $f_2(x_1, x_2, x_3)$ – формула глубины 1.

2). $f_3(f_1(x_3, x_1), f_2(x_1, f_3(x_1, x_2)))$ – формула глубины 3, так как она содержит одну подформулу глубины 2 – $f_2(x_1, f_3(x_1, x_2))$ – и две – глубины 1 – $f_1(x_3, x_1)$ и $f_3(x_1, x_2)$.

Пусть: f_1 – дизъюнкция, f_2 – конъюнкция, f_3 – сложение по модулю 2. Тогда $f_3(f_1(x_3, x_1), f_2(x_1, f_3(x_1, x_2)))$ запишется: $(x_3 \vee x_1) \oplus (x_1 \wedge (x_1 \oplus x_2))$.

Правила подстановки формулы вместо переменной.

При подстановке формулы F вместо переменной x , все вхождения переменной x в исходное соотношение должны быть заменены одновременно формулой F .

Пример.

Сделаем подстановку в формулу $x \vee \bar{x} = 1$, подставив в неё вместо переменной x формулу F .

Соотношение $F \vee \bar{F} = 1$ получено по правилу подстановки и верно для любых F .

Соотношение $F \vee \bar{x} = 1$ получено с нарушением правила подстановки и для некоторых F может быть неверным.

Правило подстановки позволяет получать из исходных соотношений новые **эквивалентные соотношения**.

Во всякой алгебре, в том числе и в булевой алгебре равенство $F_1 = F_2$ означает, что формулы F_1 и F_2 описывают один и тот же элемент алгебры, то есть одну и ту же логическую функцию. Если какая-либо формула F

содержит подформулу F_1 , то замена F_1 на F_2 не изменяет элемента булевой алгебры f , над которым производится операция в формуле F .

Формула F'' , полученная такой заменой, эквивалентна F . Это утверждение представляет собой **правило замены подформул**. Оно позволяет получать формулы, эквивалентные данной.

Рассмотрим разницу между правилами подстановки и замены.

При подстановке:

- переменная заменяется на формулу;
- формула может быть любой, но требуется её одновременная подстановка во все вхождения переменной.

При замене:

- может быть заменена любая подформула, но не на любую другую, а только на **эквивалентную**. При этом замена всех вхождений исходной подформулы не обязательна.

Пусть имеется эквивалентность $F_1 = F_2$, где F_1 и F_2 содержат переменную x . Если вместо всех вхождений x в F_1 и F_2 подставить произвольную формулу F , то получаются новые формулы F'_1 и F'_2 , причём не обязательно $F_1 = F'_1$; $F_2 = F'_2$ однако между собой F'_1 и F'_2 будут эквивалентны.

Пример.

$$\text{Возьмём: } \underbrace{\overline{x_1 x_2}}_{F_1} = \underbrace{\overline{x_1} \vee \overline{x_2}}_{F_2} \quad (2.2).$$

Подставим $\overline{x_1} x_3$ вместо x_1 .

$$\text{Получим: } \underbrace{\overline{\overline{x_1} x_3} x_2}_{F'_1} = \underbrace{\overline{\overline{x_1} x_3} \vee \overline{x_2}}_{F'_2} \quad (2.3).$$

Причём F_1 не эквивалентна F'_1 ;

F_2 не эквивалентна F'_2 ;

однако F'_1 эквивалентна F'_2 .

Если в правой части (2.3) $\overline{x_1 x_3}$ заменить формулой $\overline{x_1} \vee \overline{x_3}$, эквивалентной ей в силу $(\overline{x_1 x_2} = \overline{x_1} \vee \overline{x_2})$, а в полученной подформуле $\overline{x_1}$ заменить на эквивалентную формулу x_1 (в силу $\overline{\overline{x}} = x$), то все формулы в построенной цепи преобразований —

$$\overline{x_1 x_3 x_2} \Leftrightarrow \overline{x_1 x_3} \vee x_2 \Leftrightarrow \overline{x_1} \vee \overline{x_3} \vee x_2 \Leftrightarrow x_1 \vee \overline{x_3} \vee x_2 \quad \text{эквивалентны. Это и}$$

есть эквивалентные преобразования.

Эквивалентные преобразования при доказательстве эквивалентности формул более эффективны, нежели их вычисления на наборах значений аргументов (переменных).

Основные правила эквивалентных преобразований.

В булевой алгебре принято опускать скобки в случаях:

а) при последовательном выполнении нескольких « \wedge » или « \vee » (например, вместо $x_1(x_2 x_3)$ пишут $x_1 x_2 x_3$);

б) если они являются внешними скобками у « \vee »-и. Например, вместо $(x_1(x_2 \vee x_3)) \vee (x_4 x_5)$ пишут $x_1(x_2 \vee x_3) \vee x_4 x_5$.

Упрощение формул (получение эквивалентных формул, содержащих меньшее число символов)

а) Поглощение.

$$x \vee xy = x; \quad x(x \vee y) = x.$$

Доказательство:

$$1). \quad x \vee xy = x \& 1 \vee xy = x(1 \vee y) = x \& 1 = x.$$

$$2). \quad x(x \vee y) = xx \vee xy = x \vee xy = x.$$

б) Склеивание.

$$xy \vee x\bar{y} = x$$

в) Обобщённое склеивание.

$$xz \vee y\bar{z} \vee xy = xz \vee y\bar{z}$$

Доказательство: применим склеивание в обратную сторону (расщепление) и поглощение:

$$xz \vee y\bar{z} \vee xy = xz \vee y\bar{z} \vee xyz \vee xy\bar{z} = xz \vee y\bar{z}.$$

г) $x \vee \bar{x}y = x \vee y$

Доказательство: $x \vee \bar{x}y = xy \vee x\bar{y} \vee \bar{x}y = xy \vee x\bar{y} \vee xy \vee \bar{x}y = x \vee y$.

Приведение булевых функций к дизъюнктивной нормальной форме (ДНФ) и к совершенной дизъюнктивной нормальной форме (СДНФ).

Элементарными конъюнкциями называются конъюнкции переменных или их отрицаний, в которых каждая переменная или её отрицание встречается не более одного раза.

ДНФ называется формула, имеющая вид дизъюнкции элементарных конъюнкций (ДНФ функции может быть не единственной).

Приведение к ДНФ

1) С помощью двойного отрицания ($\bar{\bar{x}} = x$) и правил де Моргана все отрицания «спускаются» до переменных;

2) Раскрываются скобки (применяя свойства: а) $xx = x$; б) $x \vee x = x$ (идемпотентность); в) $x\bar{x} = 0$ (закон противоречия) и г) $x \vee \bar{x} = 1$ (закон исключённого третьего);

3) Удаляются лишние конъюнкции и повторения переменных;

4) Удаляются константы (применяя свойства констант).

Пример. Привести к ДНФ выражение $xy \vee \bar{x}(y \vee xz)(\overline{x(\bar{y} \vee z) \vee yz})$.

1. преобразуем выражение $\overline{x(\overline{y \vee z}) \vee yz}$;

$$\begin{aligned} \overline{x(\overline{y \vee z}) \vee yz} &= \overline{x(\overline{y \vee z})} \cdot \overline{yz} = (\overline{x} \vee \overline{\overline{y \vee z}}) \cdot (\overline{y} \vee \overline{z}) = (\overline{x} \vee \overline{\overline{y}} \cdot \overline{\overline{z}}) (\overline{y} \vee \overline{z}) = \\ &= (\overline{x} \overline{y} \vee \overline{x} \overline{z} \vee \overline{y} \overline{y} \overline{z} \vee \overline{y} \overline{z}) = (\overline{x} \overline{y} \vee \overline{x} \overline{z} \vee \overline{y} \overline{z}); \end{aligned}$$

2. подставим полученное выражение в исходное и продолжим преобразования:

$$\begin{aligned} xy \vee (\overline{x} \overline{y} \vee \overline{x} \overline{z}) (\overline{y} \vee \overline{z}) &= xy \vee \overline{x} \overline{y} (\overline{y} \vee \overline{z}) \vee \overline{x} \overline{z} (\overline{y} \vee \overline{z}) = xy \vee \overline{x} \overline{y} \overline{y} \vee \overline{x} \overline{y} \overline{z} \vee \overline{x} \overline{z} \overline{y} \vee \overline{x} \overline{z} \overline{z} = \\ &= xy \vee \overline{x} \overline{y} \overline{z} = y(x \vee \overline{x} \overline{z}) = y(x \vee \overline{x} \overline{z} \vee \overline{x} \overline{z}) = yx \vee y\overline{z}. \end{aligned}$$

Приведение булевых выражений к СДНФ.

Всякую ДНФ можно привести к СДНФ расщеплением конъюнкций, которые содержат не все переменные ($xy = xyz \vee xy \overline{z}$), например:

$$xy \vee \overline{x} \overline{y} \overline{z} = xyz \vee xy \overline{z} \vee \overline{x} \overline{y} \overline{z}.$$

2.3.1. БУЛЕВА АЛГЕБРА ФУНКЦИЙ И ЭКВИВАЛЕНТНЫЕ ПРЕОБРАЗОВАНИЯ В НЕЙ

Все эквивалентные преобразования в булевой алгебре проводятся с помощью основных эквивалентных соотношений (законов):

- ассоциативность;
- коммутативность;
- дистрибутивность
 - а) относительно дизъюнкции;
 - б) относительно конъюнкции;

идемпотентность $x \cdot x = x$; $x \vee x = x$;

- закон двойного отрицания $\overline{\overline{x}} = x$;
- свойства констант 0 и 1 :

а) $x \cdot 1 = x$; б) $x \vee 1 = 1$; в) $\overline{0} = 1$;
 г) $x \cdot 0 = 0$; д) $x \vee 0 = x$; е) $\overline{1} = 0$;

- правила де Моргана

а) $\overline{x_1 x_2} = \overline{x_1} \vee \overline{x_2}$;

б) $\overline{x_1 \vee x_2} = \overline{x_1} \cdot \overline{x_2}$;

- закон противоречия $x \cdot \overline{x} = 0$;
- закон исключённого третьего $x \vee \overline{x} = 1$.

Данные эквивалентные соотношения отличаются тем, что:

а) они не выводимы друг из друга – убедиться в их справедливости можно, используя стандартный метод доказательства эквивалентности формул, т.е. построение таблиц истинности;

б) этих соотношений достаточно для выполнения любых эквивалентных преобразований логических формул.

Упрощение формул

Наряду с основными соотношениями для упрощения формул также используются эквивалентные соотношения, выводимые из основных с помощью эквивалентных преобразований:

- поглощение

а) $x \vee xy = x$; б) $x(x \vee y) = x$;

- склеивание

$xy \vee x\overline{y} = x$;

- обобщённое склеивание

$xz \vee y\overline{z} \vee xy = xz \vee y\overline{z}$;

$$x \vee \bar{x}y = x \vee y.$$

Правила приведения к ДНФ

Элементарная конъюнкция – это конъюнкция переменных или их отрицаний, в которой каждая переменная встречается не более одного раза.

Примеры ДНФ

$$1) yz \vee xy \vee xz \vee \bar{x}\bar{y}\bar{z};$$

$$2) \bar{y}z \vee \bar{x}y \vee xz \vee \bar{x}y\bar{z}.$$

Все процедуры приведения к ДНФ основаны на законах булевой алгебры и сводятся к следующему:

- 1) Все отрицания «спустить» до переменных;
- 2) Раскрыть скобки с помощью основных эквивалентных преобразований;
- 3) Удалить лишние конъюнкции и повторения переменных в конъюнкциях с помощью;
- 4) Удалить константы с помощью.

Процедура приведения к КНФ

Элементарной дизъюнкцией называется дизъюнкция переменных или их отрицаний, в которой каждая переменная встречается не более одного раза.

КНФ называется конъюнкция элементарных дизъюнкций.

Пример булевой функции, заданной в КНФ:

$$(x \vee \bar{y})(\bar{x} \vee z)(\bar{x} \vee y \vee \bar{z}).$$

Процедура приведения булевой функции от ДНФ к КНФ:

- 1) Применить к формуле правило двойного отрицания:

$$F = \overline{K_1 \vee K_2 \vee \dots \vee K_m} \text{ и привести } \overline{K_1 \vee K_2 \vee \dots \vee K_m} \text{ к ДНФ}$$

$K'_1 \vee K'_2 \vee \dots \vee K'_p$, где $K'_1 \vee K'_2 \vee \dots \vee K'_p$ – элементарные конъюнкции.

Тогда: $F = K_1 \vee K_2 \vee \dots \vee K_m = \overline{\overline{K_1 \vee K_2 \vee \dots \vee K_m}} = \overline{K'_1 \vee K'_2 \vee \dots \vee K'_p}$.

2) С помощью правил де Моргана освободиться от второго отрицания и преобразовать отрицания элементарных конъюнкций в элементарные дизъюнкции D_1, D_2, \dots, D_p . Тогда

$$F = \overline{K'_1 \vee K'_2 \vee \dots \vee K'_p} = \overline{K_1} \cdot \overline{K_2} \cdot \dots \cdot \overline{K_p} = D_1 D_2 \dots D_p.$$

2.3.2. ПРЕДСТАВЛЕНИЕ ПЕРЕКЛЮЧАТЕЛЬНЫХ ФУНКЦИЙ В ВИДЕ МНОГОЧЛЕНОВ.

Конституентой «1» называется переключательная функция n аргументов, которая принимает значение «1» только на одном наборе аргументов. Число различных конституент «1» среди функций n аргументов равно 2^n . Так, для $n = 2$ это f_1, f_2, f_4 и f_8 .

Обычно конституенты «1» выражают через произведение всех аргументов, каждый из которых входит в произведение либо x_i , либо $\overline{x_i}$.

Теорема Жегалкина.

Любая переключательная функция может быть представлена в виде полинома

$$f(x_1, x_2, \dots, x_n) = a_0 \oplus a_1 x_1 \oplus a_2 x_2 \oplus \dots \oplus a_n x_n + \dots + a_{n+1} x_1 x_2 + \dots + a_n x_1 x_2 \dots x_n, \quad (2.4)$$

где a_0, a_1, \dots, a_n – некоторые константы, равные «0» или «1». Знак \oplus означает операцию сложения по модулю 2:

$$x \oplus 0 = x;$$

$$x \oplus 1 = \overline{x};$$

$$x \oplus x = 0; \text{ (если чётное число переменных)}$$

$$x \oplus x \oplus x = x; \text{ (если нечётное число переменных)}$$

$$x \oplus y = y \oplus x;$$

$$x \vee y = \overline{\overline{x} \overline{y}} = (x \oplus 1) (y \oplus 1) \oplus 1 = xy \oplus x \oplus y.$$

При записи конкретной функции коэффициенты a_0, a_1, \dots, a_n выпадают, так как члены, при которых $a_i = 0$, можно опустить, а коэффициенты, равные 1 – не писать.

Доказательство. Пусть задана произвольная переключательная функция $f(x_1, x_2, \dots, x_n)$, равная «1» на некоторых наборах с номерами m_1, m_2, \dots, m_p . Очевидно, что $f(x_1, x_2, \dots, x_n)$ можно записать:

$$f(x_1, x_2, \dots, x_n) = K_{m_1} + K_{m_2} + \dots + K_{m_p} = 1, \quad (2.5)$$

где K_{m_i} – конституента «1».

Так для набора с номером m_i получаем

$$K_{m_1} + K_{m_2} + \dots + K_{m_i} + \dots + K_{m_p} = 0 + 0 + \dots + 1 + \dots + 0 = 1.$$

От формулы (2.5) легко перейти к формуле (2.4), если представить K_{m_i} в виде произведений и заменить все переменные с отрицаниями, используя соотношения $\bar{x} = x \oplus 1$ (так как отрицания не входят в 2.4).

Пусть $K_i = \bar{x}_1 x_2 x_3 \bar{x}_4$. Получим $K_i = (x_1 \oplus 1)x_2 x_3 (x_4 \oplus 1)$. Далее, используя соотношения для конъюнкции:

$$\begin{aligned} x \cdot 0 &= 0; \\ x \cdot 1 &= x; \\ x \cdot x &= x; \\ x \cdot x \cdot \dots \cdot x &= x; \\ xy &= yx; \\ \bar{x}x &= 0; \\ x(y + z) &= xy + xz - \text{дистрибутивность}, \end{aligned} \quad (2.6)$$

и приводя подобные члены в соответствии с соотношениями (2.7):

$$\underbrace{x \oplus x \oplus \dots \oplus x}_n = \begin{cases} 0, & \text{при } n \text{ четном,} \\ 1, & \text{при } n \text{ нечетном.} \end{cases} \quad (2.7)$$

Пример. Представить в виде полинома Жегалкина функцию $f_{14}(x, y)$.

$$f_{14} = \bar{x} \bar{y} \vee x \bar{y} \vee \bar{x} y .$$

$$\begin{aligned} f_{14} &= K_0 \oplus K_1 \oplus K_2 = \bar{x}\bar{y} \oplus \bar{x}y \oplus x\bar{y} = (x \oplus 1)(y \oplus 1) \oplus (x+1)y \oplus x(y+1) = \\ &= 1 \oplus x \oplus x \oplus y \oplus y \oplus xy \oplus xy \oplus xy = 1 \oplus xy. \end{aligned}$$

2.4. ФУНКЦИОНАЛЬНО ПОЛНЫЕ СИСТЕМЫ

Система функций Σ называется функционально полной системой, если любая логическая функция может быть представлена формулой над Σ , т.е. являться суперпозицией функций из Σ .

Теорема. Всякая логическая функция может быть представлена булевой формулой, т.е. как суперпозиция дизъюнкции, конъюнкции и отрицания.

Действительно, для всякой логической функции, кроме константы «0», таким представлением может служить её СДНФ. (Дизъюнкция конститuent «1», равных «1» на тех же наборах, что и заданная функция, называется СДНФ переключательной функции).

Константу «0» также можно представить булевой формулой $x \bar{x}$.

Из теоремы следует, что система $\Sigma = \{\&, \vee, \neg\}$ является функционально полной:

$$f(x_1, x_2, \dots, x_n) = K_{j_1} \vee K_{j_2} \vee \dots \vee K_{j_m}$$

где K_{j_i} – это наборы переменных, где функция принимает значение «1». СДНФ функции f представляет собой разложение функции по всем n переменным.

Согласно принципу двойственности, справедливому в алгебре Буля, имеем следующий вывод: — любая булева функция $f(x_1, x_2, \dots, x_n)$ может быть представлена в виде конъюнкции дизъюнкций её аргументов на тех наборах их значений, на которых $f(x_1, x_2, \dots, x_n)$ принимает значение «0». Таким представлением булевой функции служит совершенная конъюнктивная нормальная форма (СКНФ).

Представление булевых функций двух аргументов в формах СДНФ и СКНФ приведены в таблице 2.3.

Таблица 2.3

x	0	0	1	1	СДНФ	СКНФ
y	0	1	0	1		
$f_0(x,y)$	0	0	0	0	не имеет	$(x \vee y)(x \vee \bar{y})(\bar{x} \vee y) \cdot (\bar{x} \vee \bar{y})$
$f_1(x,y)$	0	0	0	1	xy	$(x \vee y)(x \vee \bar{y})(\bar{x} \vee y)$
$f_2(x,y)$	0	0	1	0	$x \cdot \bar{y}$	$(x \vee y)(x \vee \bar{y})(\bar{x} \vee \bar{y})$
$f_3(x,y)$	0	0	1	1	$x \cdot \bar{y} \vee xy$	$(x \vee y)(x \vee \bar{y})$
$f_4(x,y)$	0	1	0	0	$\bar{x}y$	$(x \vee y)(\bar{x} \vee y)(\bar{x} \vee \bar{y})$
$f_5(x,y)$	0	1	0	1	$\bar{x}y \vee xy$	$(x \vee y)(\bar{x} \vee y)$
$f_6(x,y)$	0	1	1	0	$\bar{x}y \vee x\bar{y}$	$(x \vee y)(\bar{x} \vee \bar{y})$
$f_7(x,y)$	0	1	1	1	$\bar{x}y \vee x\bar{y} \vee xy$	$x \vee y$
$f_8(x,y)$	1	0	0	0	$\bar{x} \cdot \bar{y}$	$(x \vee \bar{y})(\bar{x} \vee y)(\bar{x} \vee \bar{y})$
$f_9(x,y)$	1	0	0	1	$\bar{x} \cdot \bar{y} \vee xy$	$(x \vee \bar{y})(\bar{x} \vee y)$
$f_{10}(x,y)$	1	0	1	0		

					$\bar{x} \cdot \bar{y} \vee x y$	$(x \vee \bar{y})(\bar{x} \vee \bar{y})$
$f_{11}(x,y)$	1	0	1	1	$\bar{x} \cdot \bar{y} \vee x \bar{y} \vee x y$	$(x \vee \bar{y})$
$f_{12}(x,y)$	1	1	0	0	$\bar{x} \cdot \bar{y} \vee \bar{x} y$	$(\bar{x} \vee y)(\bar{x} \vee \bar{y})$
$f_{13}(x,y)$	1	1	0	1	$\bar{x} \cdot \bar{y} \vee \bar{x} y \vee x y$	$\bar{x} \vee y$
$f_{14}(x,y)$	1	1	1	0	$\bar{x} \cdot \bar{y} \vee \bar{x} y \vee x \bar{y}$	$\bar{x} \vee \bar{y}$
$f_{15}(x,y)$	1	1	1	1	$\bar{x} \cdot \bar{y} \vee \bar{x} y \vee x \bar{y} \vee x y$	не имеет

2.5. ПРИМЕРЫ

Пример 1.

Логическую функцию $f(x_1, x_2, x_3) = (x_1 \sim \neg x_2) \rightarrow ((x_1 \vee x_3) \& x_2)$ представить булевой формулой - в виде СДНФ.

Решение.

1. Функция $f(x_1, x_2, x_3)$ задана не булевой формой.
2. По исходной формуле восстановим её таблицу истинности.
3. По таблице истинности составим СДНФ заданной функции:

$$f(x_1, x_2, x_3) = \bar{x}_1 \bar{x}_2 \bar{x}_3 \vee \bar{x}_1 \bar{x}_2 x_3 \vee \bar{x}_1 x_2 \bar{x}_3 \vee x_1 x_2 \bar{x}_3 \vee x_1 x_2 x_3$$

Пример 2. Привести к ДНФ функцию:

$$f(x, y, z) = xy \vee \bar{x}(y \vee xz) \overline{(x(\bar{y} \vee z) \vee yz)}$$

Решение.

$$\begin{aligned} f(x, y, z) &= xy \vee \bar{x}(y \vee xz) \overline{(x(\bar{y} \vee z) \vee yz)} = \\ &= xy \vee (\bar{x}y \vee \bar{x}xz) \cdot \overline{(x(\bar{y} \vee z) \vee yz)} = xy \vee \bar{x}y(\bar{x} \vee (\bar{y} \vee z)) \cdot \bar{y}z = \\ &= xy \vee \bar{x}y(\bar{x} \vee \bar{y}z) \cdot (\bar{y} \vee \bar{z}) = xy \vee \bar{x}y(\bar{x}y \vee \bar{x}z \vee y\bar{y}z \vee yz\bar{z}) = \\ &= xy \vee \bar{x}y(\bar{x}y \vee \bar{x}z \vee yz) = xy \vee \bar{x}y(\bar{x}y \vee yz) = xy \vee \bar{x}yz = y(x \vee \bar{x}z) = \\ &= y(x \vee \bar{z}) = xy \vee yz. \end{aligned}$$

Пояснение к решению.

При проведении эквивалентных преобразований функции $f(x, y, z)$ применяли правила де Моргана, **законы действия с константами 0 и 1** ($x \vee \bar{x} = 1$)

и закон идемпотентности.

$$x \vee xy = x \cdot 1 \vee xy = x \cdot (y \vee \bar{y}) \vee xy = xy \vee x\bar{y} \vee xy \vee x\bar{y} = x \vee y.$$

Пример 3.

Привести к СДНФ функцию: $f(x,y,z) = xy \vee \bar{x}y\bar{z}$

Решение.

$$\begin{aligned} f(x,y,z) &= xy \vee \bar{x}y\bar{z} = xy \cdot 1 \vee \bar{x}y\bar{z} = xy \cdot (z \vee \bar{z}) \vee \bar{x}y\bar{z} = \\ &= xyz \vee xy\bar{z} \vee \bar{x}y\bar{z}. \end{aligned}$$

Пример 4.

Представить булеву формулу $x_1 x_2 \vee \bar{x}_2 (x_3 \vee x_4)$ в базисе $\{\&, \bar{}\}$ и в базисе $\{\vee, \bar{}\}$.

Решение.

Применим правила де Моргана и правило двойного отрицания.

а) в базисе $\{\&, \bar{}\}$: $f = x_1 x_2 \vee \bar{x}_2 (x_3 \vee x_4) =$

$$\text{б) } = \overline{\overline{x_1 x_2 \vee \bar{x}_2 (x_3 \vee x_4)}} = \overline{\overline{x_1 x_2} \cdot \overline{\bar{x}_2 (x_3 \vee x_4)}}.$$

$$= \overline{\overline{x_1} \vee \overline{\bar{x}_2} \vee \overline{x_3 \vee x_4}} = \overline{\overline{x_1} \vee \overline{\bar{x}_2} \vee \overline{x_3} \vee \overline{x_4}}.$$

в базисе $\{\vee, \bar{}\}$ $f = x_1 x_2 \vee \bar{x}_2 (x_3 \vee x_4) =$

Пример 5. Упростить булеву формулу:

$$f(x_1, x_2, x_3) = x_1 \vee x_1 x_3 \vee \bar{x}_1 x_2 x_3 \vee x_2 \bar{x}_3$$

Решение:

$$\begin{aligned} f(x_1, x_2, x_3) &= x_1 \vee x_1 x_3 \vee \bar{x}_1 x_2 x_3 \vee x_2 \bar{x}_3 = \\ &= x_1 (1 \vee x_3) \vee \bar{x}_1 x_2 x_3 \vee x_2 \bar{x}_3 = x_1 \vee \bar{x}_1 x_2 x_3 \vee x_2 \bar{x}_3 = \\ &= x_1 \vee x_2 x_3 \vee x_2 \bar{x}_3 = x_1 \vee x_2. \end{aligned}$$

Пример 6.

Доказать тождество: $x_1 \vee \bar{x}_1 x_2 x_3 = x_1 \vee x_2 x_3$.

Доказательство.

Для доказательства будем применять эквивалентные преобразования, свойства констант 0 и 1, законы поглощения, идемпотентности, операции обратного склеивания, на основании которых можно записать: $x \vee x \neg y = x$;

$$x \vee \neg xy = xy \vee x \neg y \vee \neg xy \vee xy = y \vee x .$$

Отсюда следует:

$$\begin{aligned} x_1 \vee \neg x_1 x_2 x_3 &= x_1 x_2 x_3 \vee \neg x_1 x_2 x_3 \vee x_1 x_2 x_3 \vee x_1 \neg x_2 \neg x_3 = x_2 x_3 \vee x_1 x_2 x_3 \vee x_1 \neg x_2 \neg x_3 = \\ &= x_2 x_3 \vee x_1 . \end{aligned}$$

Пример7. Упростить булеву формулу $f(x, y, z) = x(\neg y \vee z)(\neg x \vee y \vee z)$.

Решение.

$$\begin{aligned} f(x, y, z) &= x(\neg y \vee z)(\neg x \vee y \vee z) = (x \neg y \vee xz)(\neg x \vee y \vee z) \\ &= x \neg y \neg x \vee x \neg y \cdot y \vee \\ &\vee x \neg y \cdot z \vee x \cdot z \neg x \vee x \cdot y \cdot z \vee x \cdot z \cdot z = x \neg y \cdot z \vee x \cdot y \cdot z \vee x \cdot z \cdot z = x \neg y \cdot z \vee x \cdot y \cdot z \vee x \cdot z \\ &= \\ &= x \neg y \cdot z \vee x \cdot z = x \cdot z . \end{aligned}$$

2.6. ПЕРЕЧЕНЬ ЗАДАНИЙ НА ЛАБОРАТОРНЫЕ РАБОТЫ

Лабораторная работа №1

1. Построить матрицы смежности и инциденций для графа своего варианта.
2. По матрицам, представленным в исходных данных, построить графы, предварительно определив их тип и класс.
3. Аналитическим способом определить число маршрутов длины $L=3$ для графа своего варианта.
4. Построить все маршруты длины $L=3$, связывающие заданные вершины графа.
5. Построить метрику графа. Определить радиус, диаметр графа, периферийные и центральные вершины.

Лабораторная работа №2

Разработать алгоритм и составить программу на алг. языке Паскаль для вычисления числа компонент связности для графа своего варианта.

Лабораторная работа №3

1. В заданном графе $G=(X,U)$ выделить все максимальные полные и все максимальные пустые подграфы графа G .
2. Разработать аналитический способ проверки правильности полученного решения.

Лабораторная работа №4

1. Выполнить раскраску вершин графа G .
2. Вычислить хроматическое число графа G .
3. Выполнить рёберную раскраску графа G .
4. Провести анализ полученных решений.

Лабораторная работа №5

1. По данным матрицы расстояний R своего варианта построить кратчайший маршрут, связывающий начальную вершину x_0 с конечной x_k (номера вершин указаны в исходных данных) с помощью алгоритма Дейкстры.
2. На языке Паскаль разработать программу, расчета таблицы (исходные данные записать в отдельный файл).
3. Описать процедуру «обратного хода».
4. Провести анализ полученного решения.

Лабораторная работа №6

1. Обходы графа. Изучить содержательную постановку и математическую модель задачи коммивояжера;
2. С помощью алгоритма Литтла найти в заданном графе гамильтонов цикл наименьшей длины.
3. Разработать алгоритм нахождения эйлеровой цепи и цикла в графе.
4. Провести анализ полученного решения.

Лабораторная работа №7

1. Вычислить значение максимального потока на сети своего варианта, с помощью алгоритма **Форда - Фалкерсона**.
2. Разработать программу на алгоритмическом языке паскаль для построения разреза на информационной сети.
3. Провести анализ полученного решения.

Лабораторная работа №8

1. На двудольном графе своего варианта построить максимальное паросочетание с помощью венгерского алгоритма.
2. Разработать алгоритм для вычисления числа рёбер в максимальном паросочетании в заданном двудольном графе с помощью теоремы Кёнига-Холла.
3. Провести анализ полученных решений.

Требования к оформлению отчетов по лабораторным работам

1. Отчет оформлять в текстовом редакторе WORD или EXCEL с обязательным приложением титульного листа, оформленным в соответствии с требованиями ГОС. На титульном листе необходимо указать номер варианта индивидуального задания.
2. Отчет должен содержать:
 - заголовок с указанием № лабораторной работы;
 - название темы данной лабораторной работы;
 - исходные данные;
 - перечень программных средств, использованных при выполнении задания;
 - описание основных этапов выполнения задания с демонстрацией изменения исходных данных в ходе их выполнения;
 - чётко оформленный результат выполнения работы по каждой теме.

Список литературы

Основная

1. Хаггарти, Род. Дискретная математика для программистов. Пер. С английского: Учебное пособие для вузов. – 2-е изд., доп.: -М.: Техносфера, 2005. – 393с.
2. Макоха, А.Н. Дискретная математика. Учебное пособие для вузов. – М.: Физматлит, 2005, 368с.
3. Шапорев, С.Д. математическая логика. Курс лекций и практических занятий. Учебное пособие для вузов. БХВ – Петербург, 2005.-410с.
4. Галкина, В.Н. Дискретная математика: комбинаторная оптимизация на графах. – М.: Гелиос АРВ, 2003, -232 с.
5. Яблонский С.В. Введение в дискретную математику. Учебное пособие для вузов, 4-е издание.: -М.: Высшая школа, 2003 –384с.

Дополнительная

1. Нефедов В. Н., Осипова В. А. Курс дискретной математики. - М., 1992.
2. Пападимитру Х., Стайглиц К. Комбинаторная оптимизация. Алгоритмы и сложность. - М.: Мир, 1985. - 510 с.
3. Кристофидес Н.Г. Теория графов. Алгоритмический подход. - М.: Мир, 1978. - 432 с.
4. Новиков Ф.А. Дискретная математика для программистов. Учебник. Санкт-Петербург.: Москва, Харьков, Минск.: Питер, 2000. - 301 с..