

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ
ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ



ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

В.Н. Берцун

МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ НА ГРАФАХ

Часть 1

Учебное пособие



Томск – 2006

УДК 519.17
ББК 22.174
Б 527

Берцун В.Н. Математическое моделирование на графах.
Б 527 Часть 1: Учебное пособие. – Томск: Изд-во НТЛ, 2006. –
88 с.

ISBN 5-89503-312-1

Учебное пособие предназначено для студентов механико-математических и физико-математических факультетов университетов. Оно также может быть полезно студентам других специальностей и аспирантам, занимающимся математическим моделированием прикладных задач и оптимизацией вычислительных алгоритмов для высокопроизводительных компьютеров (кластеров).

В части 1 пособия содержится два раздела теории графов, приведён необходимый теоретический материал, а также включены задачи для самостоятельного выполнения.

УДК 519.17
ББК 22.174

Рецензент:

доктор физико-математических наук А.В. Старченко

ISBN 5-89503-312-1

© В.Н. Берцун, 2006
© Томский госуниверситет, 2006

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
Глава 1. Основные понятия теории графов	5
1.1. Из истории теории графов	5
1.2. Граф и его дополнение	6
1.3. Маршрут в графе, цикл, связность	12
1.4. Компоненты связности графа	16
1.5. Изоморфизм графов	18
1.6. Двудольные графы и их свойства	20
1.7. Ориентированные графы	23
1.8. Деревья и их свойства	28
1.9. Ациклические графы	38
<i>Задачи</i>	44
Глава 2. Плоские и планарные графы	48
2.1. Свойства плоского графа	48
2.2. Эйлеровы графы	53
2.3. Гамильтоновы графы	57
2.4. Гиперкуб и его свойства	61
2.5. Графы сеточных функций	63
<i>Задачи</i>	73
ЛИТЕРАТУРА	76
БИОГРАФИЧЕСКИЙ УКАЗАТЕЛЬ	80

ВВЕДЕНИЕ

В настоящее время компьютерное моделирование резко расширило сферу своего применения в различных областях знаний. Это относится и к теории графов – разделу прикладной математики, который нашёл своё применение: в теории игр и квантовой химии, экономике и политике, логистике и социологии, биологии и медицине, оптимальном управлении и навигации, создании сложных программных комплексов и анализе современных компьютерных систем на основе сетей Петри [1 – 7]. Свойства графов активно используются и для решения краевых задач на сетевых системах (нефтепроводах, газопроводах, электросетях и т.д.), в решении сложных задач на многопроцессорных вычислительных системах (МВС). Граф алгоритма позволяет получить представление о том, как распространяется и преобразуется информация при его реализации [8, 9], что особенно важно для оптимизации параллельных вычислительных алгоритмов. Перспективным направлением решения сложных задач теории графов являются имитационные методы, основанные на природных механизмах принятия решений (клеточных автоматах, муравьиных алгоритмах, генетических алгоритмах и др.) [10].

Глава 1

ОСНОВНЫЕ ПОНЯТИЯ ТЕОРИИ ГРАФОВ

1.1. Из истории теории графов

Теория графов как математическая дисциплина стала активно развиваться еще со времен Эйлера (1707 – 1783 гг.), который в 1736 г. решил задачу о Кёнигсбергских мостах [2]. *В городе два острова, соединенные семью мостами* (см. рис. 1.1). *Можно ли побывать на всех четырёх частях суши, пройдя по каждому мосту один раз и оказаться на той части суши, с которой началось движение?*

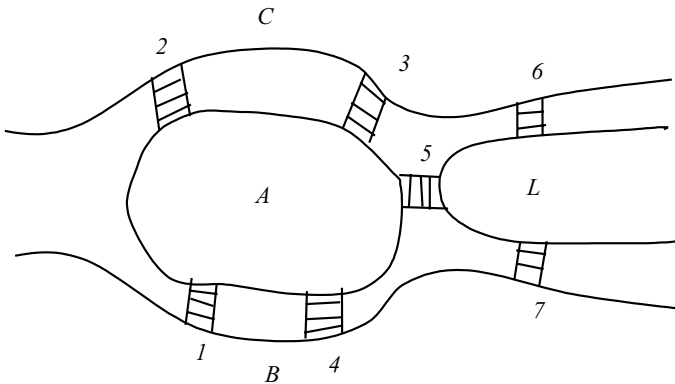


Рис. 1.1

Термин *граф*, который был введен в употребление Кенигом, подразумевает наличие наглядной графической интерпретации рассматриваемого объекта. Эйлер отождествил с точкой (вершиной графа) каждую часть суши, а каждый мост – с линией (ребром графа),

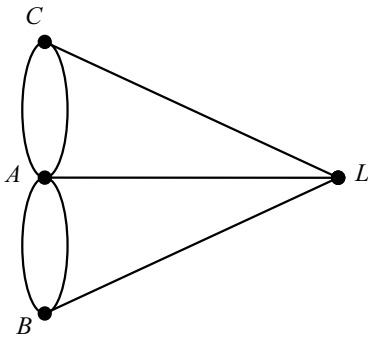


Рис. 1.2

соединяющей соответствующие точки. Получился граф, изображённый на рис. 1.2.

Анализируя этот граф, Эйлер доказал, что сформулированная выше задача о мостах не имеет решения.

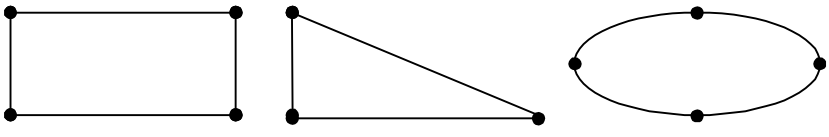
Важным стимулом к развитию теории графов явилась возникшая в середине XIX в. задача о четырех красках. Любую карту на плоскости раскрасить в четыре цвета

так, чтобы смежные страны имели различные цвета. Решить эту задачу удалось только в конце XX в. с помощью компьютера [11].

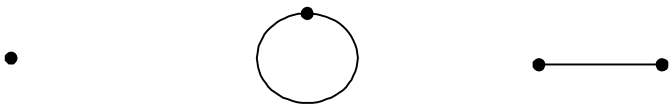
1.2. Граф и его дополнение

Графом $G(V, E)$ называется совокупность двух множеств – непустого множества объектов некоторой природы V (вершин графа) и множества E неупорядоченных пар элементов множества V , называемых ребрами графа ($V \neq \emptyset, E \subseteq V \times V$). Таким образом, граф определяется множеством вершин V , множеством рёбер E (подмножеством двухэлементных подмножеств множества V) и отношением инцидентности, которое каждому ребру сопоставляет одну или две вершины. При изображении графов на рисунках отрезки (ребра) могут быть криволинейными и прямолинейными, а длины отрезков и расположение точек произвольно. Будем рассматривать только такие графы, у которых множества $V(G)$ и $E(G)$ конечны ($n = n(G) = |V|, m = m(G) = |E|$) [11 – 16].

Ребро графа называется *звенем*, если у него два конца, и петель, если конец один (петля это ребро, у которого два конца совпадают). Два или более звеньев, имеющих одинаковые пары концов, образуют кратное соединение и называются кратными рёбрами. Граф без петель и кратных рёбер называется *простым*. Примеры графов изображены на рис. 1.3.



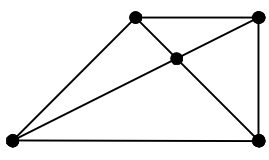
Один и тот же граф $n = 4, m = 4$



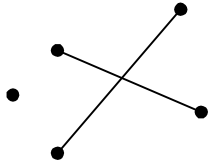
Граф-вершина

Граф-петля

Граф-звено



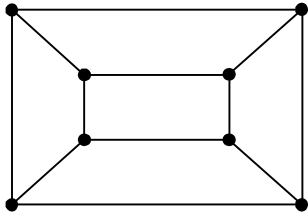
$n = 5, m = 8$



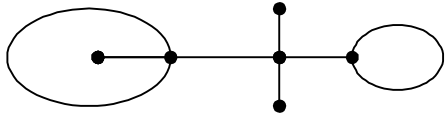
$n = 5, m = 2$



$n = 3, m = 0$



Граф куба



Граф с петлями

Рис. 1.3

Вершины графа будем обозначать буквами русского и латинского алфавитов или цифрами, а ребра графа – парами вершин $(A, B), (B, C)$ или буквами латинского алфавита.

Мультиграфом называется пара множеств, состоящая из множества вершин и множества ребер, причем две вершины могут быть соединены более чем одним ребром. Например, граф на рис. 1.4 является мультиграфом.

Говорят, что вершины графа x, y – *смежные*, если они соединены ребром. Поэтому вершины C и D на рис. 1.4 являются смежными, а D и A – нет.

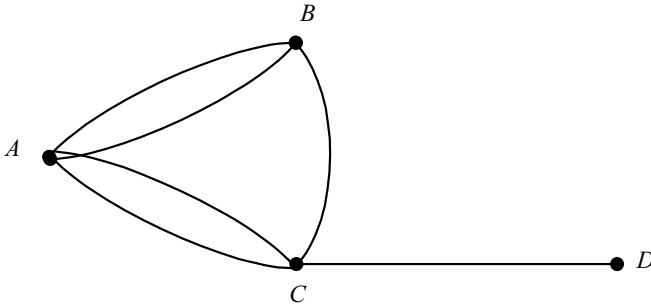


Рис. 1.4

Граф называется *полным*, если любые две различные его вершины соединены одним и только одним ребром. Такой граф имеет максимальное число ребер. Каждой вершине в полном графе с n вершинами, который в дальнейшем будем обозначать K_n , принадлежит $(n - 1)$ ребро. Но в произведении $n(n - 1)$ каждое ребро учитывается дважды, поэтому в полном графе $n(n - 1)/2$ ребер. На рис. 1.5 приведены примеры полных графов: K_3, K_4, K_5 .

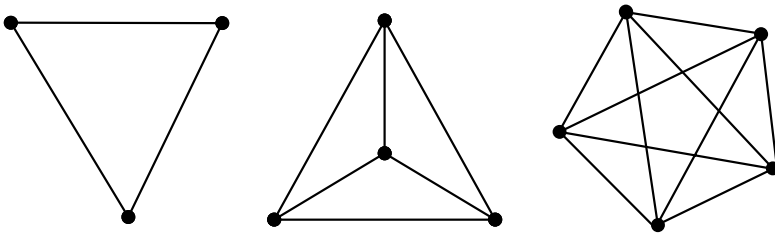


Рис. 1.5

Граф, не являющийся полным, можно преобразовать в полный, добавив к нему недостающие ребра. *Дополнением* графа G называется граф D с теми же вершинами, что и граф G , и теми и только теми ребрами, которые необходимо добавить к графу G , чтобы

получился полный граф. На рис. 1.6 представлен неполный граф и его дополнение.

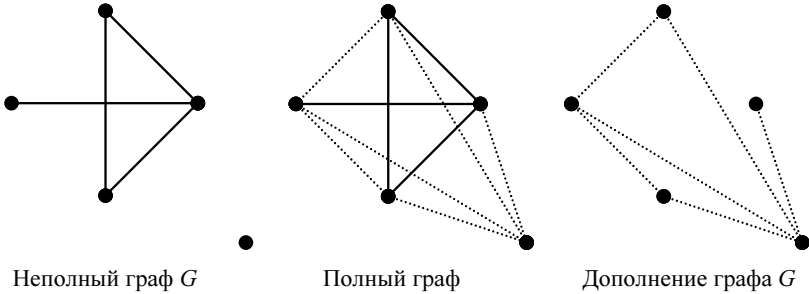


Рис. 1.6

В табл. 1.1 приведено число различных графов с n вершинами и $m \leq n(n - 1)/2$ ребрами, которые можно получить из полного графа [3].

Таблица 1.1

$m \setminus n$	2	3	4	5
0	1	1	1	1
1	1	1	1	1
2		1	2	2
3		1	3	4
4			2	6
5			1	6
6			1	6
7				4
8				2
9				1
10				1
Σ	2	4	11	34

В общем случае, в графе число ребер, которым принадлежит та или иная вершина, различно.

Степенью (валентностью) v_A вершины A называется число рёбер графа, которым принадлежит эта вершина (число ребер, инцидент-

ных вершине A , причем петля считается дважды). На рис. 1.7 представлены графы с различными степенями вершин.

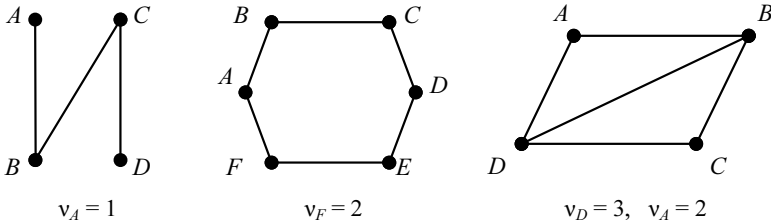


Рис. 1.7

Вершина A является *четной*, если v_A – четно, и *нечетной*, если v_A – нечетно. Вершины, у которых $v_A = 1$, называются *висячими*. У полного графа с n вершинами степень любой вершины $v_A = n - 1$, для изолированной вершины $v_A = 0$.

Утверждение 1.1. Во всяком графе G сумма степеней всех его вершин число четное, равное удвоенному числу ребер графа.

Доказательство. При определении степеней вершин графа каждое ребро учитывается два раза, поэтому

$$\sum_{i=1}^n v_i = 2p,$$

где n – число вершин, p – число ребер. ■

Утверждение 1.2. Во всяком графе с n вершинами ($n \geq 2$) всегда найдется по крайней мере две вершины с одинаковыми степенями.

Доказательство. Каждая вершина графа с n вершинами может иметь степень $0, 1, 2, \dots, n - 1$. Пусть все вершины имеют разную степень. Но этого не может быть, так как если есть вершина степени 0, то не может быть вершины степени $n - 1$ (так как она может быть соединена всего лишь с $n - 2$ оставшимися вершинами). Таким образом, найдутся хотя бы две вершины с одинаковыми степенями. ■

Часть вершин графа G и все инцидентные к ним ребра образуют *подграф* графа G . Если такой подграф полный, то он называется *кликой* графа G . Очевидно, что множество подграфов определяется количеством вершин исходного графа. Все вершины и часть инци-

дентных им рёбер называется *суграфом* (остовным подграфом) графа G . Например, на рис. 1.8 приведены подграф и суграф графа G_1 .

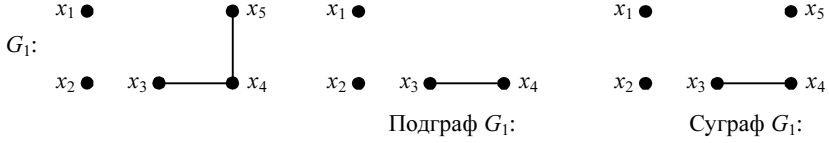


Рис. 1.8

Если A и B два множества, то для них можно ввести следующие операции: объединение, разность и пересечение множеств. На рис. 1.9 для этих операций приведены диаграммы Эйлера.

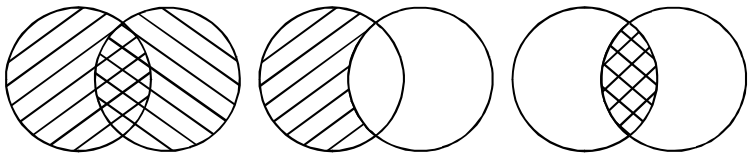


Рис. 1.9

Соответствующие операции (объединение, разность и пересечение) вводятся и для графов. При этом предполагается, что при удалении вершины графа удаляются и все инцидентные ей ребра. На рис. 1.10 для графов G_1 и G_2 приведены графы $G_1 \cup G_2$, $G_1 \setminus G_2$, $G_1 \cap G_2$.

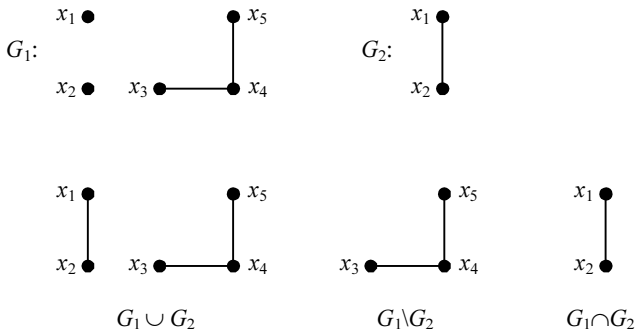


Рис. 1.10

Используя символы таблицы Менделеева, в химии графы применяются для изображения молекул. Например, на рис. 1.11 представлены мультиграфы молекул ацетилена (C_2H_2) и этилена (C_2H_4).

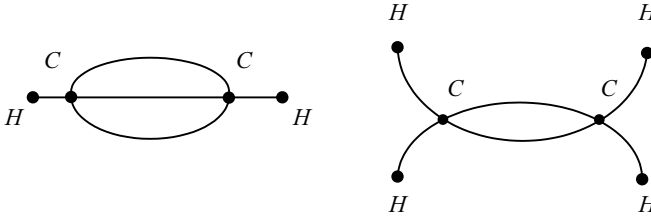


Рис. 1.11

1.3. Маршрут в графе, цикл, связанность

Рассмотрим граф на рис. 1.12, в котором из вершины x_1 в x_5 можно попасть различными способами.

Маршрутом в графе называется чередующаяся последовательность вершин a_i и ребер e_i

$$a = a_0, e_0, a_1, e_1, \dots, e_{n-1}, a_n = b,$$

где $e_i = (a_i, a_{i+1})$, a – начало маршрута, b – конец маршрута. Маршрут можно задавать, например, перечислением его вершин (a_0, a_1, \dots, a_n) . В маршруте ребра и вершины могут повторяться. Если в нем все ребра различны, то он называется *цепью*. В цепи вершины могут повторяться. Если все вершины в цепи различны, то она является *простой цепью*. Например, на рис. 1.12 последовательность вершин x_1 ,

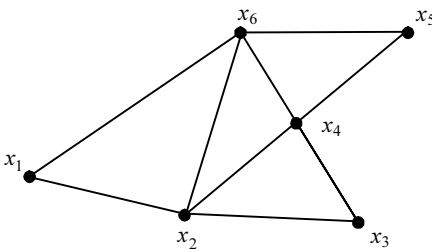


Рис. 1.12

$x_2, x_3, x_4, x_6, x_4, x_5$ является маршрутом из x_1 к x_5 , а вершины x_1, x_2, x_3, x_4, x_5 определяют простую цепь (простой путь) из x_1 в x_5 . *Длина маршрута* в графе измеряется количеством ребер в нем (с повторениями). Длина кратчайшей простой цепи, соединяющей вершины A и B , определяет *расстояние* между ними.

Циклом называется замкнутая цепь, в которой совпадают ее начальная и конечная вершины. *Простым циклом* в графе называется простая замкнутая цепь. На рис. 1.12 $x_2, x_3, x_4, x_6, x_5, x_4, x_2$ – цикл, x_2, x_3, x_4, x_2 – простой цикл. Очевидно, что в простом цикле с n вершинами ($n \geq 3$), который будем обозначать C_n , содержится n рёбер. На рис.1.13 представлены простые циклы: C_3, C_5, C_6 . *Длина цикла* измеряется числом рёбер в этом цикле.

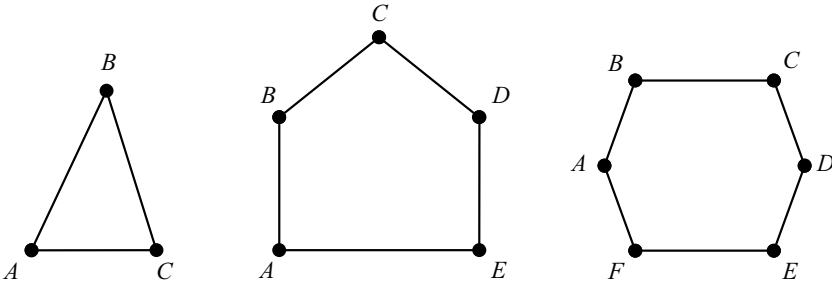


Рис. 1.13

В простой цепи число вершин на единицу больше, чем число рёбер, а в простом цикле их количество совпадает.

Для графа G на рис. 1.14 все циклы длины 4: (AB, BC, CD, DA) , (AE, ED, DC, CA) , (AD, DE, EB, BA) , (AC, CB, BE, EA) , являются простыми.

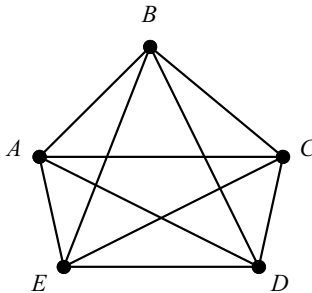


Рис. 1.14

Утверждение 1.3. В графе G любой цикл содержит простой цикл.

Доказательство. При числе ребер $m = 1$ цикл (петля с одним ребром) является простым. Пусть утверждение верно для циклов длины не больше $m - 1$. Рассмотрим произвольный цикл C длины m . Он либо является простым, либо проходит через некоторые вершины более одного раза. На таких вершинах существует цикл длины не больше $m - 1$, содержащий простой цикл. ■

Утверждение 1.4. Если у графа G все простые циклы четной длины, то граф не имеет ни одного цикла нечетной длины.

Доказательство. 1. Если граф является простым циклом, то доказательство очевидно.

2. Пусть у графа с простыми циклами четной длины найдется цикл нечетной длины. Во всяком непростом цикле нечетной длины найдется вершина, которая повторяется более одного раза, как, например, на рис. 1.15.

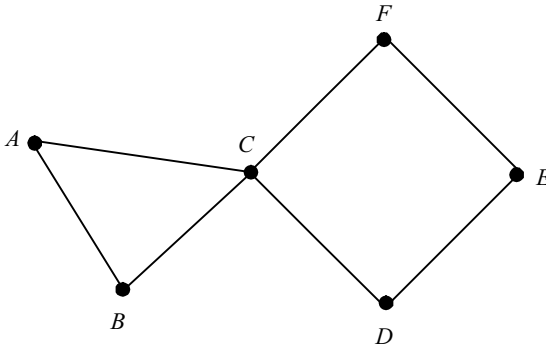


Рис. 1.15

В такой вершине цикл можно разделить на два: четной и нечетной длины. Будем продолжать расчленять непростые нечетные циклы на четные и нечетные, пока не дойдем до простых циклов. Один из таких циклов должен иметь нечетную длину. А это противоречит условию, что все циклы четные. ■

Две вершины A и B графа G называются *связанными*, если в графе есть цепь (путь) с концами A и B . Две вершины A и B не связаны в G , если в графе нет ни одного пути, связывающего их. Граф называется *связным*, если любые две его вершины связаны. Граф называется

1.4. Компоненты связности графа

Точкой сочленения связного графа называется вершина, после удаления которой граф становится несвязным (ребро с таким же свойством называется *мостом*). *Неразделимым графом* называется связный, непустой, не имеющий точек сочленения граф. *Блок графа* – это максимально неразделимый подграф. На рис. 1.17 граф имеет две точки сочленения, мост и четыре блока.

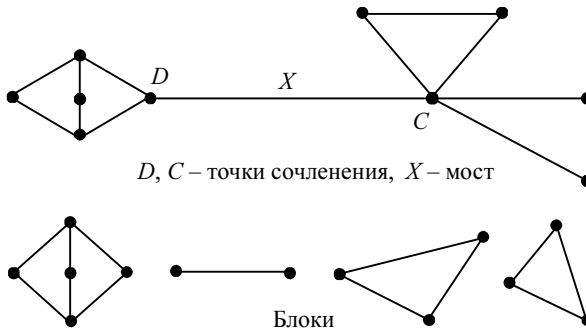


Рис. 1.17

Вершинной связностью χ графа G называется наименьшее число вершин, удаление которых приводит к несвязному или тривиальному (одновершинному) графу. Следовательно, связность несвязного графа $\chi = 0$, связность связного графа с точкой сочленения $\chi = 1$. Если в полном графе K_n удалить $n - 1$ вершину, то получим тривиальный граф, поэтому для такого графа $\chi = n - 1$. Для простого цикла всегда $\chi(C_n) = 2$. На рис. 1.18 для полного графа K_4 и простого цикла C_8 вершинная связность $\chi(K_4) = 3$, $\chi(C_8) = 2$.

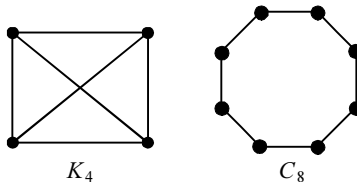


Рис. 1.18

Рассмотрим граф на рис. 1.19 с точкой сочленения C . Удалив эту вершину, получим несвязный граф.

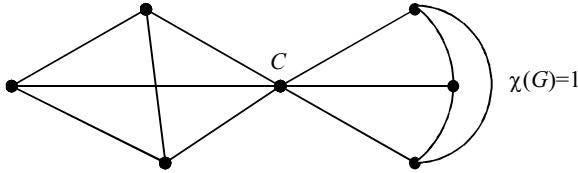


Рис. 1.19

Однако чтобы этот граф стал несвязным путем удаления ребер, надо в нем удалить не менее трех ребер. *Число реберной связности* $\lambda(G)$ называется наименьшее число ребер, удаление которых приводит к несвязному графу. Для любого графа G верны неравенства $\chi(G) \leq \lambda(G) \leq \delta(G)$, где $\delta(G)$ – минимальная степень вершин графа G .

Граф связан, если любые две его вершины соединены цепью. Если в произвольном графе G вершина x_i связана с x_j , а x_j связана с x_k , то x_i связана с x_k . Тогда, если V_i есть множество вершин G , связанных с вершиной x_i , то множество вершин графа G , которое обозначим через V , можно представить в виде попарно не пересекающихся подмножеств [2]

$$V = \bigcup_i V_i.$$

Соответственно

$$G = \bigcup_i G_i,$$

где G_i – связные непересекающиеся подграфы G .

Если число таких подграфов P , то говорят, что G имеет P компонент связности. Любой граф можно рассматривать как совокупность связных графов. Всякий максимально связный подграф графа G называется *компонентой связности* графа G .

Например, графы на рис. 1.16 имеют соответственно одну и две компоненты связности.

Утверждение 1.5. Граф G связан тогда и только тогда, когда он состоит из единственной компоненты связности.

Доказательство. Пусть множества вершин V_i и V_j определяют две различные компоненты связности. Тогда нет цепи (пути) из вершины $x_i \in V_i$ в вершину $x_j \in V_j$, а это противоречит связности графа. ■

1.5. Изоморфизм графов

Как уже отмечалось, один и тот же граф на рисунках может выглядеть по-разному. Например, на рис. 1.20 изображен один и тот же граф [15, 16].

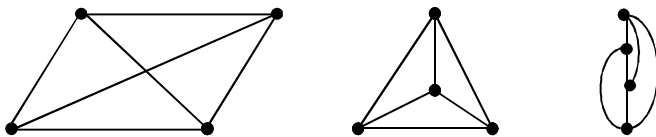


Рис. 1.20

Говорят, что два графа $G_1(V_1, E_1)$ и $G_2(V_2, E_2)$ *изоморфны* ($G_1 \sim G_2$), если существует биекция (взаимно-однозначное отображение) $f: V_1 \rightarrow V_2$, сохраняющая смежность [16]:

$$e_1 = (a, b) \in E_1 \Rightarrow e_2 = (f(a), f(b)) \in E_2,$$

$$e_2 = (a, b) \in E_2 \Rightarrow e_1 = (f^{-1}(a), f^{-1}(b)) \in E_1.$$

Изоморфизм графов есть отношение эквивалентности, так как он обладает свойствами: *рефлексивности* ($G_1 \sim G_1$); *симметричности* (если $G_1 \sim G_2$, то $G_2 \sim G_1$); *транзитивности* (если $G_1 \sim G_2$ и $G_2 \sim G_3$, то $G_1 \sim G_3$). Понятие изоморфизма для графов имеет наглядное толкование. Допустим, что рёбра графов являются эластичными нитями, связывающими узлы – вершины. Тогда изоморфизм можно представить как перемещение узлов и растяжение нитей. Очевидно, что два графа являются изоморфными, если они отличаются лишь идентификацией своих вершин. Например, на рис. 1.21 приведены два графа G и G_1 , изоморфизм которых определяется изоморфной подстановкой [17]

$$f_1 = \begin{pmatrix} A B C D E F \\ 1 6 5 4 3 2 \end{pmatrix}.$$

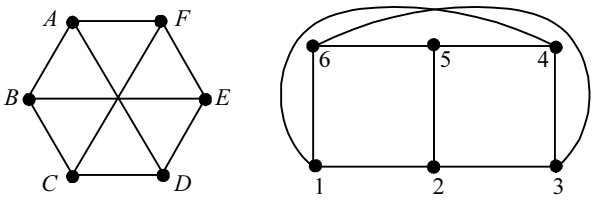


Рис. 1.21

Для изоморфных графов некоторые числовые характеристики (*инварианты*) сохраняются. К инвариантам относятся, например, число вершин, число ребер графа и др. Однако совпадение числа вершин, числа ребер и степеней вершин не всегда соответствует изоморфным графам. Неизвестно набора инвариантов, определяющих граф с точностью до изоморфизма.

Свойства графов можно исследовать и с помощью матриц, связанных с ними. *Матрицей смежности вершин* помеченного графа G называется матрица $A = [a_{ij}]_1^n$, где элемент a_{ij} есть число ребер, соединяющих вершины x_i и x_j (при этом петля означает два ребра). Очевидно, что для простых графов такие матрицы являются симметричными. На рис. 1.22 представлены неизоморфные графы, различающиеся отсутствием или наличием простых циклов нечетной длины.

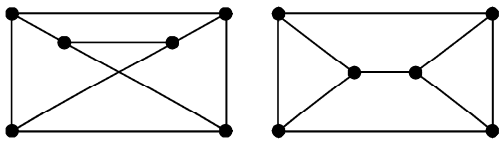


Рис. 1.22

Матрицы смежности для этих графов имеют соответственно вид

$$A = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \end{pmatrix}, \quad A_1 = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \end{pmatrix}$$

Теоретически можно проверить два графа G и G_1 с n вершинами на изоморфизм, переставляя строки и соответствующие столбцы матрицы смежности, например, графа G , пока она не превратится в соответствующую матрицу графа G_1 [1]. Если это не произойдет после $n!$ перестановок, то графы неизоморфны. Однако практическое использование такого алгоритма при компьютерном анализе малоэффективно для больших значений n .

Задачи распознавания изоморфизма графов, изоморфного вложения и изоморфного пересечения возникают, например, при автоматизации проектирования радиоэлектронной аппаратуры, анализе молекулярных структур химических соединений, разработке эффективных параллельных вычислительных алгоритмов [16].

1.6. Двудольные графы и их свойства

Множество вершин $U \subseteq V$ называется *вершинным покрытием* множества E , если каждое ребро графа $G(V, E)$ инцидентно некоторой вершине из U . Очевидно, что все вершины связного графа V всегда образуют одно из вершинных покрытий. Наименьшее число вершин в вершинных покрытиях графа G называется *числом вершинного покрытия* α_0 , а наименьшее число ребер в реберных покрытиях определяет *число реберного покрытия* α_1 . Эти два числа являются инвариантами графа [16]. Например, $\alpha_0(K_n) = n - 1$, $\alpha_1(K_{2n}) = n$, $\alpha_1(K_{2n+1}) = n + 1$, $\alpha_1(C_{2n}) = n$, $\alpha_1(C_{2n+1}) = n + 1$. Для изоморфных графов на рис. 1.20 $\alpha_0 = 3$, $\alpha_1 = 2$, а для графов на рис. 1.21 $\alpha_0 = 3$, $\alpha_1 = 3$.

Множество вершин (ребер) графа называется *независимым*, если никакие две (два) из них не смежные. Максимальная мощность вершин в независимом множестве вершин называется *вершинным числом независимости* (числом внутренней устойчивости) β_0 . Соответственно максимальная мощность ребер в независимом множестве ребер называется *реберным числом независимости* β_1 . Например, $\beta_0(K_n) = 1$, $\beta_1(K_{2n}) = n$, $\beta_1(K_{2n+1}) = n - 1$. Известно [16], что в связном графе ($n > 1$) имеют место соотношения

$$\alpha_0 + \beta_0 = n, \alpha_1 + \beta_1 = n.$$

Граф $G = (V, E)$ называется *двудольным* (или *биграфом*), если существует разбиение множества его вершин V на две части (доли) $V = A \cup B$ ($A \cap B = \emptyset$) такое, что никакие две вершины из A или из B не являются смежными. В таком графе $G(A, B, E)$ концы каждого ребра принадлежат разным долям. Например, на рис. 1.23, *в* граф не является двудольным [2, 11, 18].

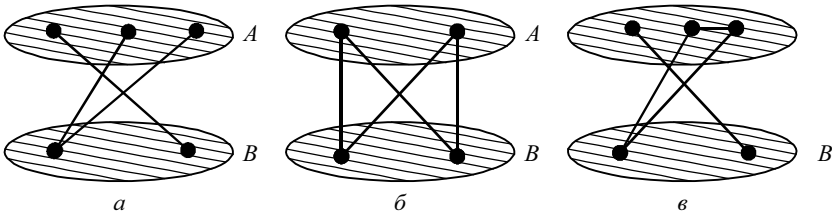


Рис. 1.23

Отметим, что у графа на рис. 1.23, *б* две любые вершины, принадлежащие разным долям, являются смежными. Такие двудольные графы называются *полными* и обозначаются через $K_{n, m}$, где n, m — количество вершин каждой доли. Граф $K_{1, m}$ называется *звездным*.

Утверждение 1.6. Граф является двудольным тогда и только тогда, когда все простые циклы в нем имеют четную длину.

Доказательство. Необходимость. Пусть $G(A, B, E)$ — двудольный граф, имеющий простой цикл нечетной длины $a_1, a_2, \dots, a_{2j+1}, a_1$, в котором вершины с нечетными номерами находятся в доле A . Тогда наличие ребра $(a_1, a_{2j+1}) \in A$ противоречит двудольности графа.

Достаточность. Пусть G — связный граф (для несвязного графа каждую компоненту можно рассматривать отдельно), который не содержит простых нечетных циклов. Выберем произвольную вершину $a_1 \in G$, которую поместим в долю A .

В эту же долю включим все вершины, удаленные от a_1 на четное число ребер (на четное расстояние). Остальные вершины (с нечетным расстоянием до $a_1 \in A$) поместим в долю B . Допустим, что во множестве вершин доли B есть две вершины p, q , соединенные ребром $(p, q) \in E$. Рассмотрим две кратчайшие простые цепи (a_1, p) и (a_1, q) . Они имеют нечетную длину и хотя бы одну общую вершину.

Пусть b наиболее удаленная от a_1 общая вершина рассматриваемых цепей, тогда расстояние

$$|(b, p)| + |(b, q)| = |(a_1, p)| + |(a_1, q)| - 2|(a_1, b)|$$

оценивается четным числом ребер. Поэтому у графа есть простой цикл $(b, \dots, p, q, \dots, b)$ нечетной длины, что противоречит условию. Таким образом, никакие две вершины множества B не соединены ребром. Это утверждение имеет место и для вершин во множестве A . ■

Паросочетанием (независимым множеством ребер) называется множество M попарно не смежных ребер графа $G(V, E)$. На рис. 1.24 для графа $K_{5,5}$ приведен пример паросочетания и максимального паросочетания [11, 16, 17], для которого $\beta_1 = 5$.

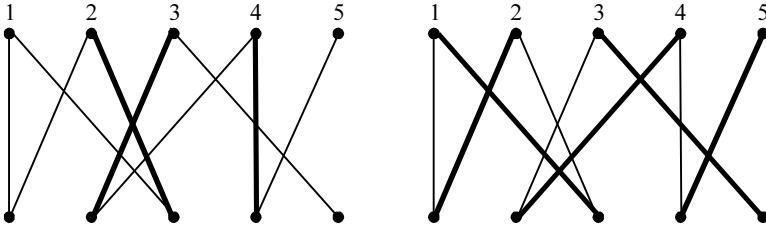


Рис. 1.24

Если каждая вершина множества $U \subseteq V$ инцидентна какому-либо ребру паросочетания M , то говорят, что M покрывает множество U . Известно [11], что максимальная мощность паросочетания в двудольном графе равна минимальной мощности его вершинного покрытия α_0 .

Для полного двудольного графа

$$\alpha_0(K_{n,m}) = \min(n,m), \alpha_1(K_{n,m}) = \max(n,m),$$

$$\beta_0(K_{n,m}) = \max(n,m), \beta_1(K_{n,m}) = \min(n,m).$$

К необходимости нахождения паросочетания в графе приводят различные практические задачи. Например, пусть имеется группа из n рабочих, каждый из которых может выполнить один или несколько из m необходимых видов работ, но каждый рабочий должен вы-

полнить только один вид работы. Требуется оптимально распределить имеющиеся работы среди рабочих, чтобы выполнить наибольшее число работ. Очевидно, что эта задача сводится к нахождению максимального паросочетания в двудольном графе.

1.7. Ориентированные графы

Для решения различных практических задач часто возникает необходимость рассмотрения ориентированных графов (орграфов). Например, орграф возникает при изображении графа улиц и перекрестков города с односторонним движением (рис. 1.25).

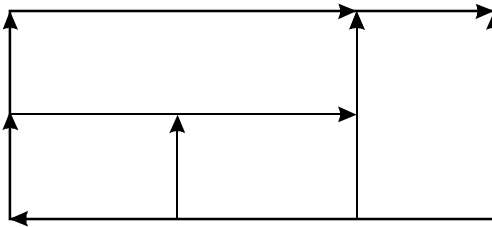


Рис. 1.25

Схемы программ для ЭВМ или, например, сетевые графики также являются ориентированными графами [2, 9]. На рис. 1.26 орграфом представлены результаты кругового турнира по футболу без ничьих в один круг, где A, B, C, D, E – команды. Как следует из рисунка, команда B выиграла все встречи, C проиграла все встречи, команда A выиграла встречи с C, D, E . Граф такого вида называется *турниром*.

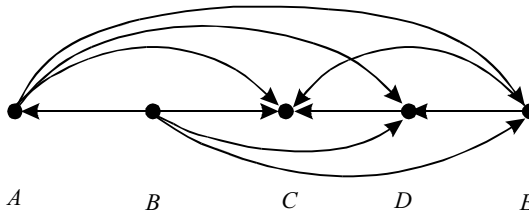


Рис. 1.26

Ориентированным графом $\bar{G} = (X, \Gamma)$ называется множество X , не обязательно конечное, рассматриваемое вместе с некоторым отображением $\Gamma: X \rightarrow X$, которое может не быть однозначным. Элементы множества X (или точки) называются вершинами орграфа \bar{G} , а пара вершин (x, y) называется дугой, если $y = \Gamma x$. Точка x в этом случае – начало дуги (предшествующая вершина), а точка y – конец дуги (последующая вершина)[1 – 3]. Например, в орграфе \bar{G} на рис. 1.27 изображено 10 дуг для числа вершин $n = 5$.

$$X = \{x_1, x_2, x_3, x_4, x_5\}, \Gamma x_1 = \{x_2, x_4, x_5\}, \Gamma x_2 = \{x_2, x_3\}, \\ \Gamma x_3 = \{x_4, x_5\}, \Gamma x_4 = \{x_1, x_5\}, \Gamma x_5 = \{x_1\}.$$

Две вершины называются смежными, если они соединены дугой, а две дуги смежные, если у них общая вершина. Например, на рис. 1.27 вершины x_1, x_2 и дуги u_1, u_3 являются смежными.

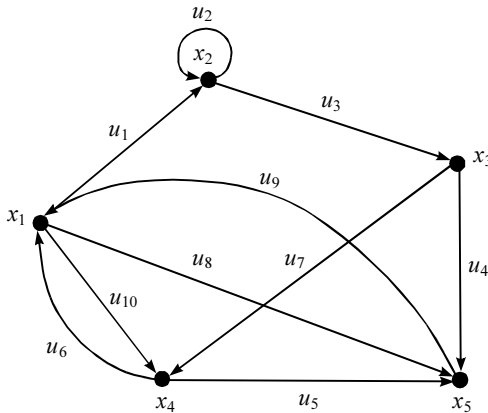


Рис. 1.27

Говорят, что вершина x и дуга графа u являются инцидентными, если вершина x есть начало или конец этой дуги. На рис. 1.27 вершина x_1 инцидентна дуге u_1 , а вершина x_5 не инцидентна дуге u_3 .

Полустепенью исхода (степенью выхода) μ вершины A орграфа называется число выходящих из A дуг, а полустепенью захода (сте-

пенью входа) v – число входящих дуг. Очевидно, что в графе \bar{G} для изолированной вершины степень входа и степень выхода равны нулю. Если в вершине графа есть только выходящие дуги, то она называется *источником*, если есть только входящие дуги, то она называется *стоком*. На рис. 1.26 вершина C является стоком, а вершина B – источником.

Путем в графе \bar{G} от вершины A_1 к A_n называется последовательность ориентированных ребер $(A_1, A_2), \dots, (A_{n-1}, A_n)$ такая, что конец каждого предыдущего совпадает с началом следующего. Очевидно, что если в \bar{G} есть путь от A к B , то пути B к A может не быть. Если существует ориентированный путь от A к B , то говорят, что B достижима из A . Длина пути измеряется числом дуг в пути. В графе \bar{G} расстояние $d(A, B)$ от вершины A до B , определяется длиной кратчайшего пути от A до B . Если пути от A к B нет (вершина B не достижима из A), то длина пути между этими вершинами считается бесконечной ($d(A, B) = \infty$). Например, у графа на рис. 1.28 $d(A, B) = 1$, $d(C, B) = 2$, $d(B, C) = \infty$.

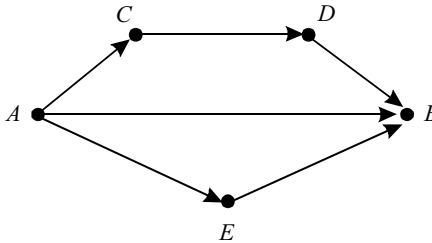


Рис. 1.28

Очевидно, что можно провести аналогию между понятием пути и цепи в неориентированном графе. Для орграфов цепь называется *путем*, простая цепь – *простым путем*, цикл (замкнутая цепь) – *контуром*, а простой цикл – *простым контуром*.

Орграф называется *сильным* (сильно связным), если любые две его вершины достижимы друг для друга и *односторонним*, если для любой пары его вершин, по меньшей мере, одна достижима из другой.

На рис. 1.29 приведен пример сильного и одностороннего графа, где на рис. 1.29, a (C, A, B, C) – простой контур.

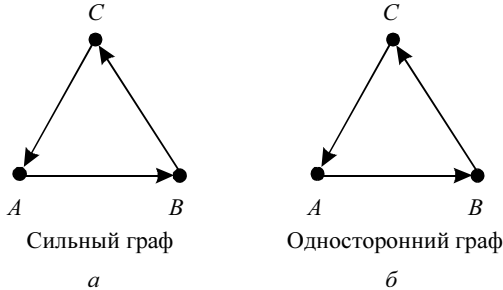


Рис. 1.29

Каждый сильный граф всегда является и односторонним.

Граф \bar{G} называется *полным (турниром)*, если каждая пара его вершин соединена в точности одним ориентированным ребром. В полном ориентированном графе без петель с n вершинами и m дугами

$$m = n(n - 1)/2.$$

Орграф может не содержать контуров. *Ациклический (бесконтурный) орграф* – орграф, не содержащий контуров, но возможно имеющий циклы. Например, орграф на рис. 1.29, b является ациклическим. Для орграфов бесконтурные графы играют роль, подобную роли деревьев в множестве неориентированных графов.

При анализе и оптимизации сложных программ часто пользуются понятием *управляющего графа программы* (уграфа) [19]. Вершинам такого орграфа соответствуют операторы алгоритмического языка, а дуги соответствуют возможным передачам управления. Если каждой вершине приписать число операций, которым она соответствует, а дугам приписать условия перехода к новой вершине, то получим направленный взвешенный уграф. В таком графе имеется одна начальная вершина и одна или несколько конечных вершин, вес которых считается равным нулю [20]. На рис. 1.30 представлены основные фрагменты уграфов вычислительных алгоритмов.

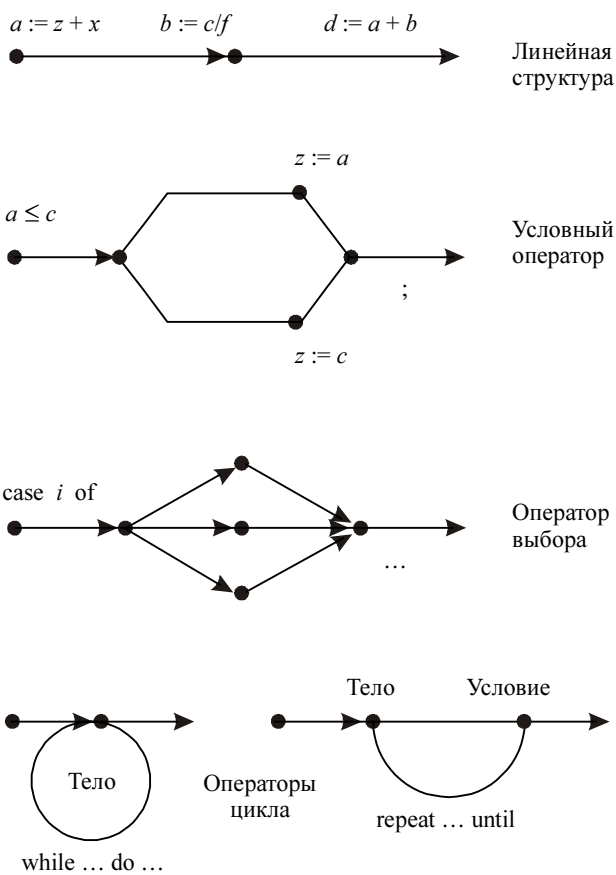


Рис. 1.30

Для изучения структурных характеристик и свойств сетей ЭВМ, параллельных вычислительных систем, систем автоматического управления процессами, которые являются дискретными динамическими системами, используются сетевые модели К. Петри [21, 22]. *Сеть Петри* является примером двудольного орграфа $G(A, B, E)$, где A – множество переходов, B – множество мест, $A \cap B = \emptyset$, E – множество дуг, соединяющих места и переходы.

1.8. Деревья и их свойства

Простейшим классом графов являются деревья, которые играют важную роль в современных компьютерных технологиях и программировании [23, 24]. Это понятие в теорию графов было введено еще в XIX в. Г. Кирхгофом для анализа электрических цепей и А. Кэли при анализе структур химических формул. Деревья – очень удобный инструмент представления информации самого разного вида. *Деревом* называется всякий связный граф, не имеющий циклов [17, 25]. Изолированную вершину также можно считать деревом. Деревом является, например, изображение на карте реки с ее притоками. Примеры деревьев представлены на рис. 1.31.

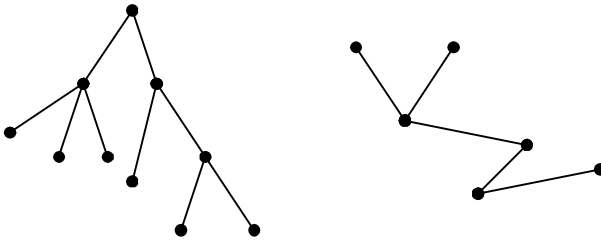


Рис. 1.31

Если ребрам (дугам) графа приписаны некоторые веса, то он называется *взвешенным*. Каждое ребро дерева является мостом, а вершины степени 1 называются *висячими*.


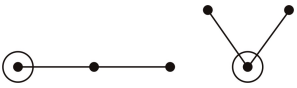
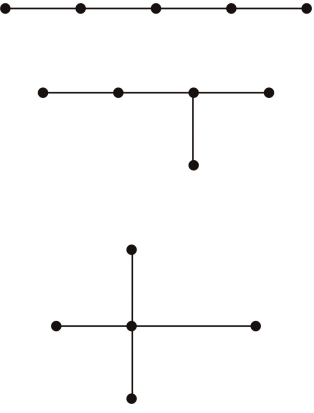
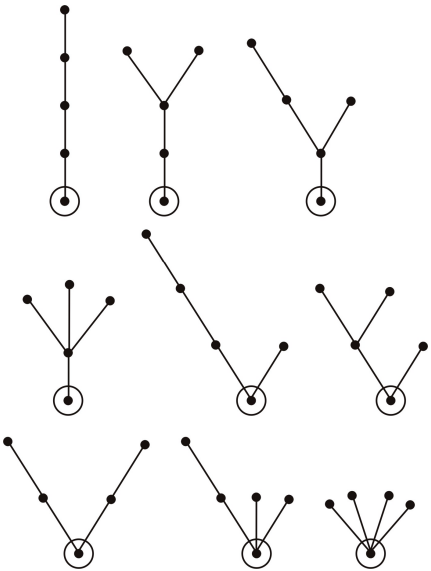
Утверждение 1.7. Дерево с n вершинами имеет $n - 1$ ребро.

Доказательство. Для изолированной вершины доказательство очевидно. Если G – дерево (см. рис. 1.31), то удалив из G одно ребро, получим два дерева с теми же вершинами. Для получения трех деревьев необходимо удалить два ребра. Для того чтобы из дерева с n вершинами получить n изолированных вершин, необходимо удалить $n - 1$ ребро. ■

Отметим, что любые две различные вершины дерева соединяет единственная простая цепь. Центр дерева содержит одну или две смежные вершины, и каждая цепь наибольшей длины проходит че-

рез центр дерева. *Лесом* называется несвязный граф, представляющий собой объединение деревьев. Очевидно, что лес является графом без циклов (*ациклическим графом*). Дерево с выделенной вершиной (корнем) называется *корневым деревом*, а без выделенной вершины – *свободным деревом*. В дереве с n вершинами можно выбрать n различных корней. В табл. 1 приведено множество различных свободных и корневых деревьев для $n = 3, 5$.

Таблица 1.2

n	Свободные деревья	Корневые деревья
3		
5		

Известно, что понятие корневого дерева используется, например, при организации работы файловой системы компьютеров, где файлы объединены в особые структуры – папки и каталоги. *Помеченным* называется граф, вершинам которого приписаны метки (номера,

цвета и т.п.). Например, на рис. 1.32 изображены два различных помеченных дерева, содержащих одно и то же множество вершин.

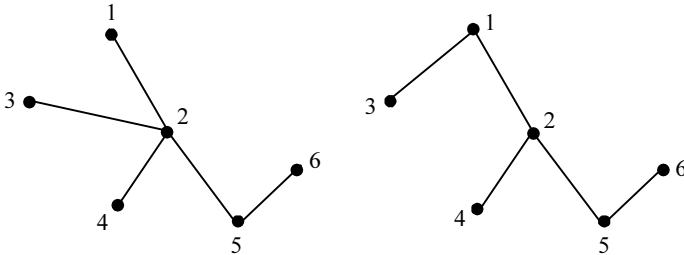


Рис. 1.32

Пусть нумерация вершин дерева G с n вершинами $(1, 2, \dots, n)$ при различном соединении их ребрами остается неизменной. В дереве всегда есть хотя бы две висячие вершины. Выберем ту из них, которая имеет минимальный номер, и удалим ее вместе с инцидентным ей ребром. Если другой конец этого ребра был инцидентен вершине с номером i_1 , то эту вершину поместим в вектор $v(i_1)$. На втором и последующих шагах процесс повторяется, пока не останется одно ребро. Таким образом, получим вектор $v(i_1, i_2, \dots, i_{n-2})$, состоящий из $n - 2$ компонент, который называется *кодом дерева G* . Например, для дерева на рис. 1.32, $a - v(2, 2, 2, 5)$, а для дерева на рис. 1.32, $b - v(1, 2, 2, 5)$.

Утверждение 1.8 (теорема Кэли). Число различных помеченных деревьев с n вершинами $t_p = n^{n-2}$.

Доказательство. Число возможных вариантов записи кода дерева $v(i_1, i_2, \dots, i_{n-2})$ совпадает с числом размещений с повторениями из n элементов по $n - 2$ [26, 27]. Тогда число различных помеченных деревьев

$$t_p = \overline{A}_n^{n-2} = n^{n-2}. \blacksquare$$

Отметим, что в формуле Кэли подсчитывается число всех помеченных деревьев. Если их рассматривать как свободные деревья, то многие из них изоморфны. В табл. 1.3 для четырех значений n приведено число различных деревьев [28, 29].

Таблица 1.3

n	3	4	5	7
t_p	3	16	125	16807
t_k	2	4	9	48
t_c	1	2	3	11

t_p – число помеченных деревьев, t_k – число корневых деревьев, t_c – число свободных деревьев.

На рис. 1.33 представлены помеченные деревья, соответствующие расположению атомов в молекуле бутана (C_4H_{10}) и изобутана (C_4H_{10}). Эти два вещества имеют одинаковое количество атомов углерода и водорода, но различная структура молекул обуславливает, например, различие их температуры кипения и плавления [30].

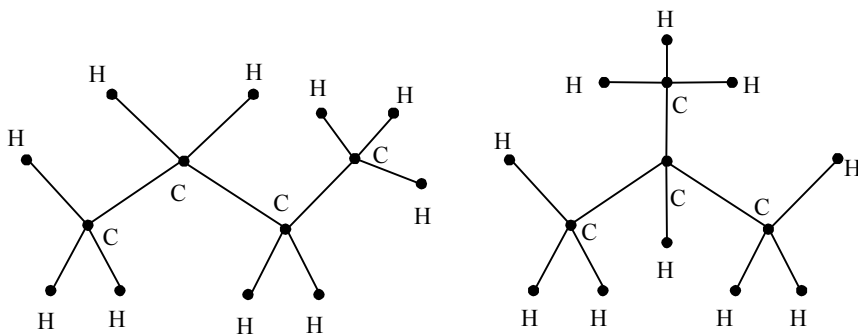


Рис. 1.33

Число простых помеченных графов $T(n)$ с n вершинами и число помеченных графов с n вершинами и m ребрами $T_p(n, m)$ определяется соответственно по формулам [3, 29]

$$T_p(n) = 2^q,$$

$$T_p(n, m) = \binom{q}{m} = C_q^m = \frac{q!}{m!(q-m)!},$$

$$q = C_n^2 = \frac{n!}{2!(n-2)!} = n(n-1)/2.$$

На рис. 1.34 изображены все помеченные графы с тремя вершинами.

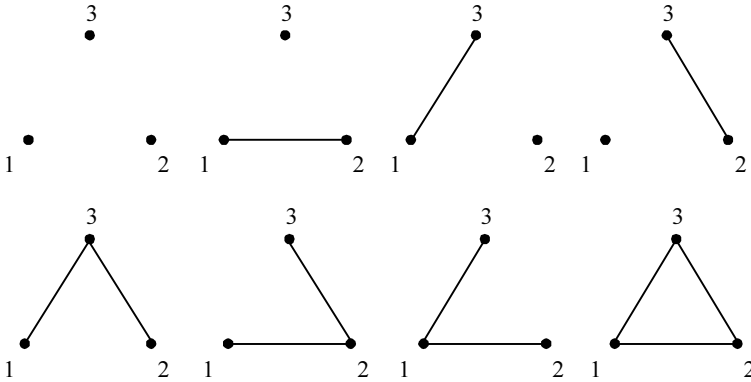


Рис. 1.34

Таким образом, четыре различных графа с тремя вершинами порождают 8 помеченных графов, из которых 3 являются помеченными деревьями с тремя вершинами. Отметим, что число помеченных корневых деревьев $t_{pk} = n \cdot t_p$, так как существует n вариантов выбора корня. Из асимптотической формулы Пойа для числа помеченных графов [15] следует, что число таких графов «примерно» в $n!$ раз меньше числа помеченных графов на n вершинах.

Остовным деревом (каркасом) называется суграф в виде дерева. На рис. 1.35 представлены каркасы графа G .

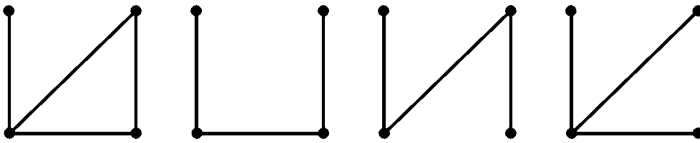


Рис. 1.35

При решении задач проектирования сетей часто требуется минимизировать, например, их стоимость или длину. В этом случае возникает задача построения минимального остовного дерева для графа, ребрам которого приспаны некоторые веса (стоимость, рас-

стояние). Для нахождения остовных деревьев, удовлетворяющих определенным условиям, можно использовать, например, алгоритм поиска «в глубину» или алгоритм поиска «в ширину» из [15, 16, 25].

Ориентированным деревом с корнем a называется корневое дерево, в котором каждое ребро заменено дугой таким образом, что либо из каждой вершины можно попасть в корень, двигаясь вдоль ориентации дуг (*входящее дерево*), либо в каждую вершину можно попасть из корня, двигаясь вдоль ориентации дуг (*выходящее или растущее дерево*). Таким образом, задание корня превращает дерево в ордереве, которое является ациклическим графом.

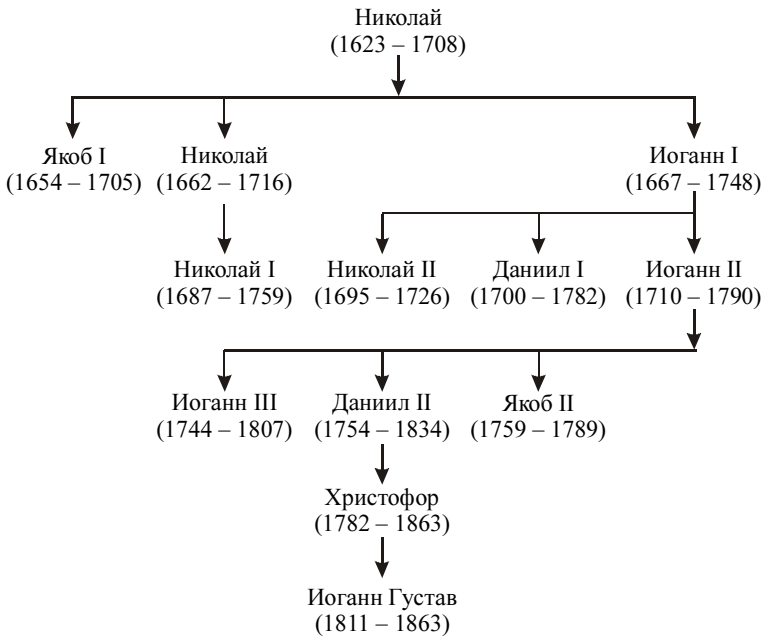


Рис. 1.36. Упорядоченное родословное дерево семьи Бернулли

В ордереве каждая концевая вершина называется *листом*, а путь, ведущий из корня в лист, называется *ветвью*. Вершины дерева, находящиеся на одном расстоянии от корня, называются *ярусом* дерева. Номер яруса определяется расстоянием до корня. Длина наи-

большей ветви называется *высотой* ордерова. Изображение генеалогических (родословных) деревьев предполагает, что для дуги (u,v) вершина u называется *отцом* (родителем) вершины v , а вершина v называется *сыном* (потомком) вершины u . Сыновья одной вершины называются *братьями*. Если в ордерове множество сыновей каждой вершины упорядочено слева направо, то такое дерево называется *упорядоченным*. На рис. 1.36 приведено упорядоченное родословное дерево семьи швейцарских ученых Бернулли.

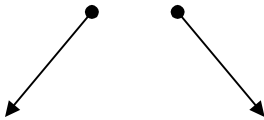


Рис. 1.37

В классе орграфов существуют бинарные (двоичные) деревья, которые часто используются для хранения, обработки и представления информации. *Бинарное дерево* состоит из корня и двух непересекающихся бинарных деревьев – левого и правого, возможно пустых. Такое дерево не является упорядоченным. Например, на рис. 1.37 приведено левое и правое бинарные деревья, которые изоморфны как упорядоченные, ориентированные и свободные деревья, но не изоморфны как бинарные деревья.

На рис. 1.38 изображено двоичное дерево с весами, которое используется при кодировании информации по алгоритму Хаффмана [16]. В соответствии с этим деревом кода, например, слово MMF соответствует двоичному коду: 000110001101001.

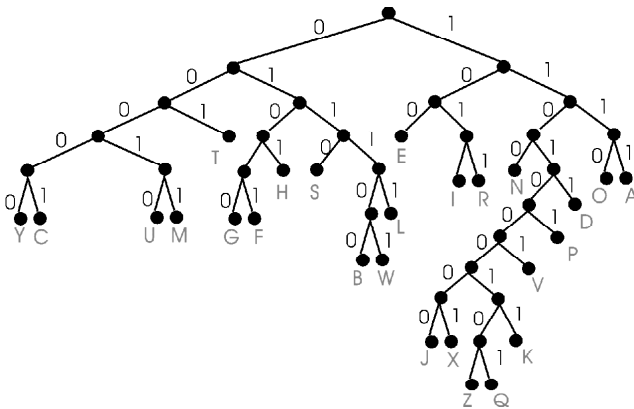


Рис. 1.38

На рис. 1.39 представлено бинарное четырехъярусное дерево решений для сортировки по убыванию трехэлементного массива простым выбором

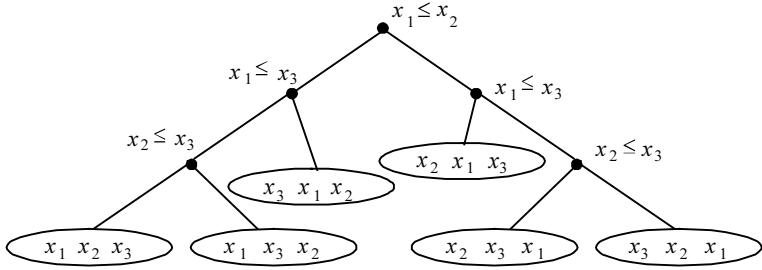


Рис. 1.39

Если корень дерева соответствует первому ярусу, то на m -м ярусе бинарного дерева находится $2^{(m-1)}$ узел, а у полного двоичного (идеального) дерева с k ярусами $2^k - 1$ узел. У дерева с $N!$ листьями число ярусов k есть наименьшее целое число, для которого

$$N! \leq 2^{(k-1)} \text{ или } \log_2 N! \leq k - 1 .$$

Если у каждой вершины дерева, кроме последней, есть только один потомок, то такое дерево называется *вырожденным* и соответствует линейному списку. Для числа бинарных деревьев на n вершинах, которое определяется как сумма чисел возможных комбинаций левых и правых поддеревьев, имеют место формулы [20]

$$t_n = C_{2n}^n / (n + 1), \quad t_v = 2^{(n-1)},$$

где t_n – число бинарных деревьев, t_v – число вырожденных деревьев. На рис. 1.40 приведены все возможные вырожденные деревья для $n = 3$.

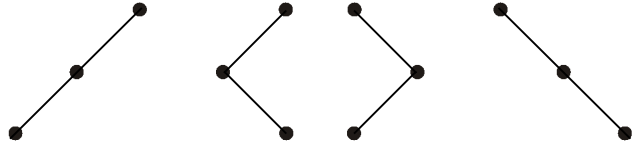


Рис. 1.40

Всякий алгоритм сортировки, основанный на попарном сравнении, имеет свое дерево решений, где самый короткий путь из корня в лист соответствует наилучшему случаю. Для оптимального алгоритма каждая перестановка из N элементов массива появится только один раз, а в дереве решений должно быть не меньше $N!$ листьев. Так как

$$\log_2 N! = \log_2 (N(N-1)(N-2)\dots 1) \approx N \log_2 N,$$

то количество операций в такой сортировке $O(N \log_2 N)$.

На рис. 1.41 изображено кодовое дерево азбуки Морзе [31], где используется не более пяти знаков (\bullet , —) для кодирования букв русского языка.

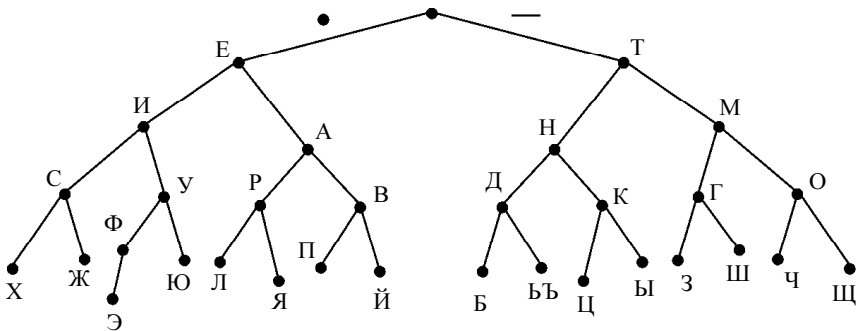


Рис. 1.41

Кодирование начинается с вершины двоичного дерева. В каждом узле поворот к левому дереву добавляет в коде буквы точку, а к правому – тире.

Рассмотрим пример построения и анализа дерева в задаче о лабиринте. В лабиринте (см. рис. 1.42, *а*) морская свинка должна найти пищу [32]. Сколькими способами она может это сделать, если ни в один тупик она не заходит более одного раза. Причём, попав в тупик, она возвращается на перекрёсток, с которого свернула. Дерево маршрутов для этого лабиринта изображено на рис. 1.42, *б*. Как следует из этого графа, всего существует 8 возможных маршрутов, а самый оптимальный из них имеет длину 4.

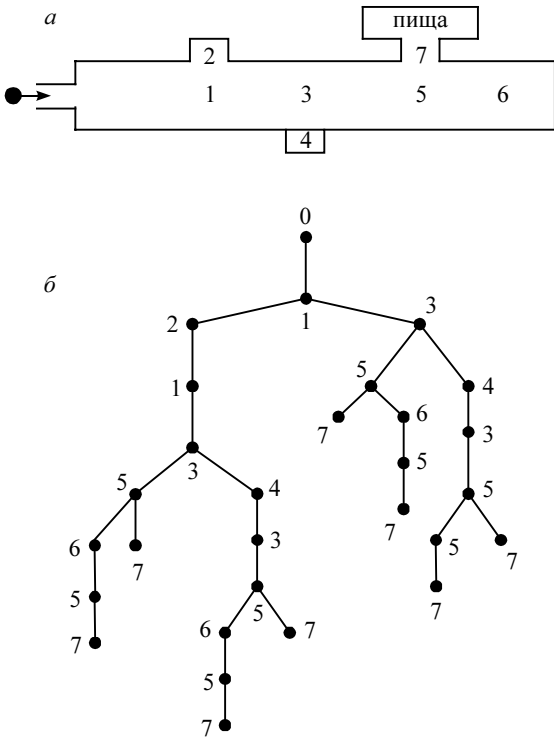


Рис. 1.42

С задачей о нахождении минимального остовного дерева (МОД), которую можно решить с помощью «жадного» алгоритма Дейкстры – Прима [23], тесно связана задача Штейнера о кратчайшем соединении (по сумме расстояний) точек на плоскости [16]. При решении этой задачи разрешается к исходному графу добавлять новые вершины (точки Штейнера) с целью обеспечения минимальной суммы расстояний. Частным случаем этой постановки является задача Торричелли – Ферма: *в треугольнике ABC найти точку P такую, чтобы сумма расстояний от P до вершин A, B, C была минимальной* [18]. Известно, что в этом случае все три угла в точке P, образованные прямыми PA, PB, PC, равны 120°.

1.9. Ациклические графы

В орграфе с конечным числом вершин всегда существует хотя бы один простой путь максимальной длины. Такой путь называется *критическим путем*. Это понятие играет важную роль, например, при анализе сетевых графиков.

Утверждение 1.9. В любом ациклическом орграфе существуют хотя бы одна вершина с нулевой полустепенью захода и хотя бы одна вершина с нулевой полустепенью исхода.

Доказательство. В графе \bar{G} всегда существует хотя бы один критический путь. Рассмотрим один из них. Допустим, что первая его вершина имеет предшествующую вершину. Если предшествующая вершина совпадает с одной из вершин выбранного критического пути, тогда в графе есть контур. Если предшествующая вершина не совпадает ни с одной из вершин, образующих критический путь, тогда существует путь, имеющий большую длину, чем критический. Таким образом, в обоих случаях получаем противоречие с предположением о том, что полустепень захода первой вершины отлична от нуля. Аналогичным образом доказывается существование вершины, не имеющей последующей. ■

Утверждение 1.10. Пусть ациклический орграф имеет n вершин. Существует целое число $s \leq n$, для которого все вершины графа можно так пометить одним из индексов $1, 2, \dots, s$, что если дуга из вершины с индексом i идет в вершину с индексом j , то $i < j$.

Доказательство. Выберем в графе любое число вершин, которые не имеют предшествующих, и пометим их индексом 1. Удалим их из графа вместе с инцидентными им дугами. В оставшемся ациклическом графе выберем любое число вершин, не имеющих предшествующих, и пометим их индексом 2. Снова удалим из графа помеченные вершины и инцидентные им дуги. Так как при каждом шаге помечается не менее одной вершины, то число различных индексов не превышает числа вершин графа. ■

Следствия из утверждения 1.10:

1) никакие две вершины с одним и тем же индексом не связаны дугой;

2) длина критического пути графа на 1 меньше минимального числа индексов, которыми можно пометить все его вершины;

3) для любого целого числа $s \leq n$, но большего длины критического пути, существует такая разметка вершин ациклического графа G , в которой используются все s индексов.

Граф, размеченный в соответствии с утверждением 1.10, называется *параллельной формой графа*, а процесс разметки вершин называется *топологической сортировкой графа* [9]. Такая сортировка называется *линейной*, если все вершины помечены разными индексами. Например, на рис. 1.43 вершины графа линейно отсортированы и $s = 5$.

Если при разметке графа все входные вершины окажутся в одной группе с индексом 1, то такая форма называется *канонической*. У графа такая форма единственна.

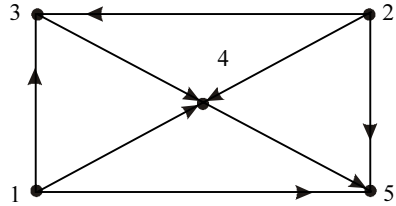


Рис. 1.43

Для детерминированных алгоритмов программная реализация не содержит условных операторов. Соответствующие им информационные графы имеют наиболее простую структуру. В этом случае всегда существует взаимно-однозначное отображение между вершинами информационного графа алгоритма и всеми операциями соответствующей ему программы. Отметим, что в любой программе реализуются только явные вычисления, поэтому информационный граф детерминированного алгоритма всегда является ациклическим орграфом. Допустим, что все операции алгоритма разделены на группы, причем каждая операция любой группы зависит либо от начальных данных, либо от результатов выполнения операций, находящихся в предыдущих группах. Такое представление алгоритма называется *параллельной формой алгоритма (ПФА)*. Каждая группа операций называется *ярусом ПФА*, число групп – *высотой ПФА*, максимальное число операций в ярусе – *шириной ПФА*.

Пусть точками на плоскости, которые будем считать вершинами орграфа, изображено множество всех операций алгоритма. Каждой вершине можно поставить в соответствие имя и размер (количество

операций, соответствующих данной вершине графа). Дугами будем изображать каналы обмена данными. Каждой дуге можно поставить в соответствие время пересылки и пересылаемую переменную. Если (в соответствии с алгоритмом) операция u поставляет аргумент для выполнения операции v , то u соединяется дугой с вершиной v . Когда результат выполнения одной операции используется в N других операциях, то из соответствующей вершины выходит N дуг. Если аргументами являются входные данные, то соответствующие дуги графа могут отсутствовать. Так как в топологической сортировке никакие две вершины с одним индексом не связаны дугой, то их можно отождествить с ярусом ПФА, а число вершин с одним индексом с шириной яруса. Таким образом, если граф описывает некоторый алгоритм, то множество параллельных форм алгоритма определяется количеством его топологических сортировок. Например, на рис. 1.44 приведены две топологические сортировки одного и того же алгоритма вычисления суммы

$$S = a_1 + a_2 + \dots + a_8$$

с помощью алгоритма попарного суммирования (алгоритма сдвоявания). Здесь в качестве начальных вершин выбраны начальные данные $a_i, i = \overline{1,8}$.

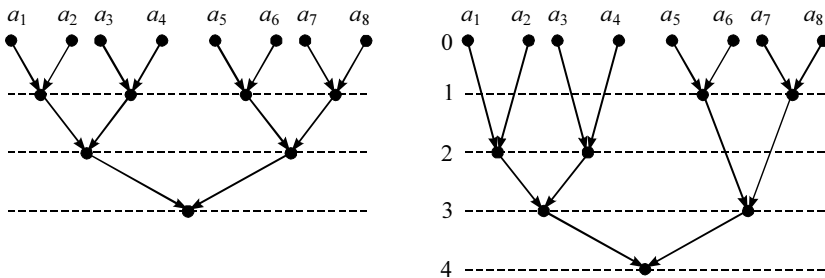


Рис. 1.44

Таким образом, зная параллельные формы алгоритма, можно выбрать наиболее оптимальную из них для реализации на конкретной МВС. На выбор оптимальных ПФА влияет и структура вычисли-

тельных формул. Например, пусть требуется вычислить

$$S = ((a_1 a_2) a_3 + a_4) a_5 + a_6. \tag{1}$$

С учетом правил действий с алгебраическими выражениями (1) перепишем в иной форме

$$S = (a_1 a_2)(a_3 a_5) + (a_4 a_5) + a_6. \tag{2}$$

Формула (2) более удобна для распараллеливания. На рис. 1.45 приведены соответствующие графы последовательного алгоритма (формула 1) и параллельного (формула 2).

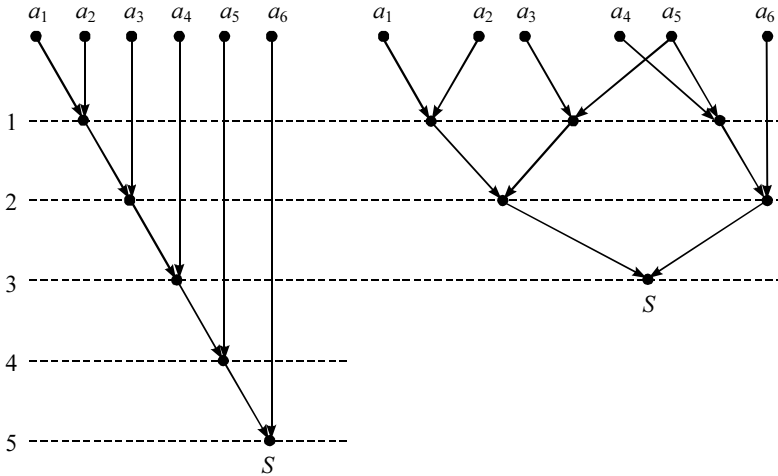


Рис. 1.45

Как следует из этого рисунка, форма записи алгебраического выражения может изменить число операций и высоту ПФА.

При реализации алгоритма на МВС операции на ярусе ПФА выполняются за один такт, а количество тактов совпадает с высотой ПФА. Например, на рис. 1.44 для четырехпроцессорной вычислительной системы по алгоритму сдваивания результат получается за три такта, а для двух процессоров – за четыре такта.

При суммировании по алгоритму сдваивания четного числа слагаемых $N = 2^k$ общее число операций суммирования

$$L = N/2 + N/4 + \dots + 1 = N - 1,$$

а количество параллельных операций $k = \log_2 N$ соответствует числу тактов.

Концепция неограниченного параллелизма [9], возникшая в середине прошлого века, предполагает, что алгоритм, разработанный для МВС, зависит только от числа процессоров p и имеет необходимую память, одновременно доступную всем процессорам. В рамках этой идеальной концепции алгоритмы оптимизируются только по высоте ПФА. Эффективная реализация алгоритмов на МВС зависит не только от числа процессоров p , которые пронумерованы от 0 до $p - 1$, но и строения оперативной памяти. МВС может иметь общую для всех процессоров память; либо распределенную память, когда каждый процессор имеет свою локальную память. При создании вычислительного алгоритма необходимо учитывать и организацию связи между процессорами (топологию сети), которая характеризуется диаметром соответствующего связного графа.

Например, на рис. 1.46 представлен граф одного из вариантов связи между процессорами $i = 0, \dots, 15$ в виде двумерной решетки (4×4). Для таких процессорных графов внутренний узел i непосредственно связан с узлами $i \pm 1, i \pm 4$.

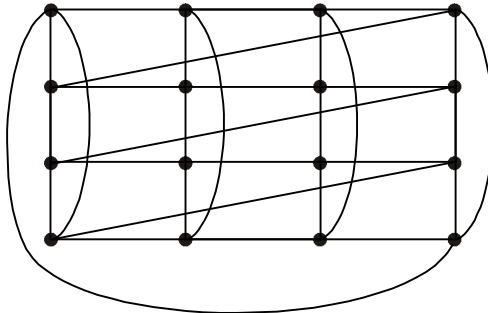
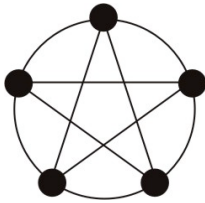


Рис. 1.46

В этом случае обмен информацией между вершиной i и несмежными вершинами осуществляется как последовательность обменов между соседними вершинами.

Аналогичный граф связи для $p = 64$ использовался в МВС с распределенной памятью ILLIAC-IV, созданной в 1974 г.

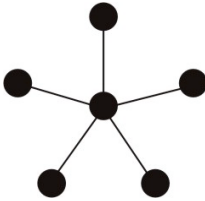
На рис 1.47 приведены примеры шести основных способов связи процессоров МВС [33]: полный граф, линейка, кольцо, звезда, решетка. Отметим, что, если в топологии решетки по каждой размерности сетки имеется только два процессора (узла), то она называется гиперкубовой.



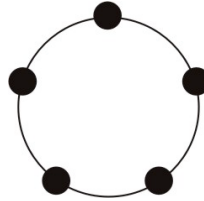
Полный граф



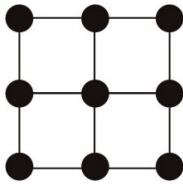
Линейка



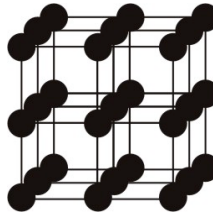
Звезда



Кольцо



2-мерная решётка

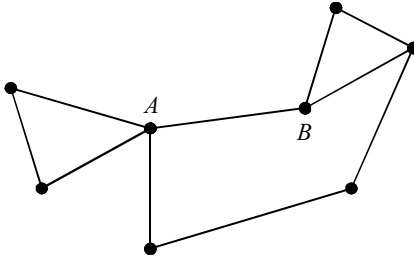


3-мерная решётка

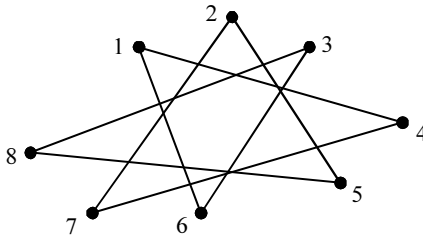
Рис. 1.47. Примеры топологий многопроцессорных вычислительных систем

Задачи

1. Является ли в графе G ребро (AB) мостом?



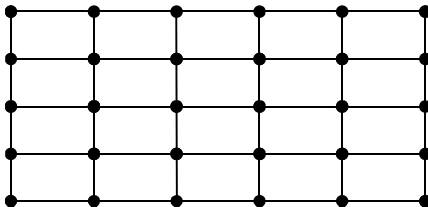
2. Найти простой цикл в графе. Является ли этот граф двудольным?



3. Нарисуйте связный граф с семью вершинами и шестью ребрами.

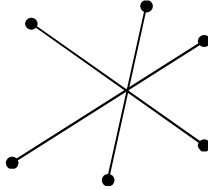
4. Можно ли из полного графа с одиннадцатью вершинами удалить часть рёбер так, чтобы степень каждой вершины была равна семи?

5. Какое наибольшее число ребер можно удалить, чтобы граф остался связным?



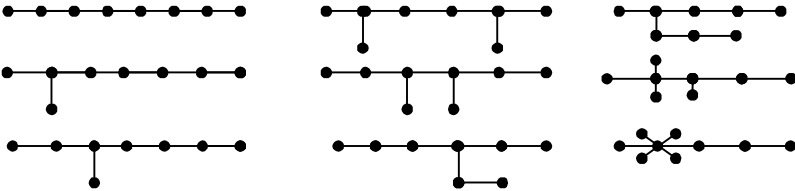
6. При встрече n друзей обменялись рукопожатиями. Сколько было друзей, если рукопожатий было 28.

7. Является ли граф двудольным?



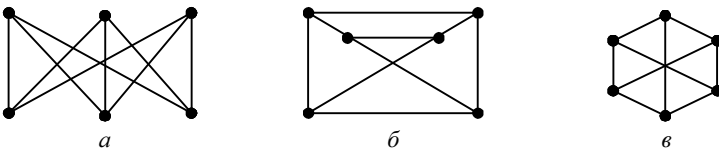
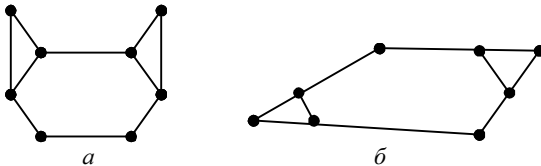
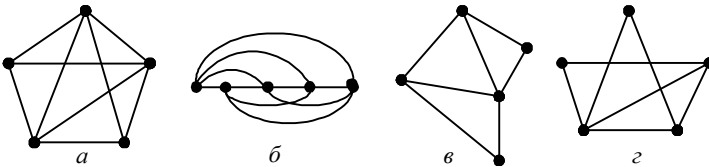
8. Сколько существует свободных деревьев с шестью вершинами?

9. Есть ли среди свободных деревьев изоморфные?



10. Составить дерево розыгрыша кубка по футболу среди 8 команд по олимпийской системе: без ничьих, проигравшая команда выбывает.

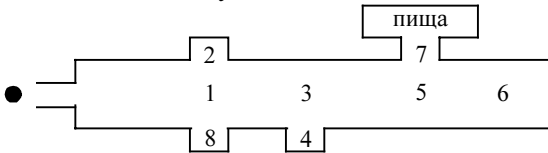
11. Изоморфны ли графы?



12. В город прибыло три группы иностранных туристов T_i , $i = 1, 2, 3$, говорящих соответственно на английском, французском и итальянском языках. Туристическое бюро располагает пятью переводчиками P_i , $i = 1, 2, \dots, 5$, владеющими соответственно языками: английским, французским и итальянским, немецким и французским, китайским и итальянским, английским. Нарисовать двудольный граф распределения всех переводчиков по группам.

13. Имеется две урны, первая из которых содержит два белых и один черный шар, а вторая – один белый и два черных шара. Нарисовать дерево логических возможностей и определить число вариантов выбора шаров из урн.

14. Нарисовать дерево маршрутов движения, если в лабиринте на рис. 1.42 появится еще один тупик.



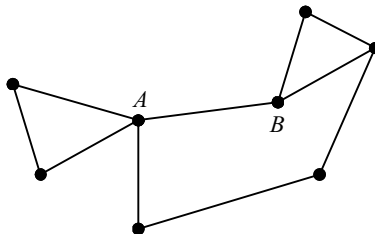
15. Доказать, что каждое дерево является двудольным графом.

16. Для кодового дерева на рис. 1.41 записать таблицу кодирования букв и проверить ее оптимальность по частотной таблице букв русского языка [31].

17. Найти радиус, диаметр и центр графа

18. Нарисовать все помеченные графы с четырьмя вершинами.

19. Найти всевозможные цепи, соединяющие вершины графа A и B . Определить длину максимальной простой цепи.



20. Нарисовать все возможные вырожденные бинарные деревья для $n = 4$.
21. Определить вершинную и реберную связность, диаметр и центр графов на рис. 1.47.
22. Доказать, что полный граф K_n имеет в точности n^{n-2} остовных деревьев.
23. Самая длинная простая цепь является диаметром графа. Доказать, что любые два диаметра имеют общую вершину.
24. Доказать, что простой граф на n вершинах не является двудольным, если он имеет более $n^2/4$ ребер.
25. Доказать, что в дереве существуют хотя бы две висячие вершины.
26. Доказать, что при добавлении ребра между двумя любыми вершинами дерева в полученном графе образуется ровно один цикл.
27. При каких условиях а задаче Торричелли – Ферма точка P находится внутри треугольника.
28. Записать матрицы смежности для графов $C_3, K_3, K_{3,3}$.
29. Доказать, что диаметр графа не превосходит его удвоенного радиуса.
30. Сколько помеченных графов порождает простой цикл C_5 ?

Владимир Николаевич Берцун

**МАТЕМАТИЧЕСКОЕ
МОДЕЛИРОВАНИЕ НА ГРАФАХ**

Часть 1

Учебное пособие

Редактор *Л.Н. Полковникова*

Дизайн, верстка *Д.В. Фортес*

К-ОКП ОК-005-93, код продукции 954240

Изд. лиц. ИД № 04000 от 12.02.2001. Подписано к печати 18.12.2006.
Формат 60 × 84 ¹/₁₆. Бумага офсетная. Печать офсетная. Гарнитура «Таймс».
Уч.-изд. л. 8,23 Тираж 200 экз. Заказ № 55.

ООО «Издательство научно-технической литературы»
634050, Томск, пл. Ново-Соборная, 1, тел. (3822) 533-335

Отпечатано в типографии ЗАО «М-Принт», г. Томск, ул. Пролетарская, 38/1