

# Структура информации

## Зачем структурировать информацию?

Давайте сравним четыре сообщения.

Первое:

«Для того чтобы добраться из Москвы до села Васино, нужно сначала долететь на самолёте до города Ивановска. Затем на электричке доехать до Ореховска. Там на пароме переправиться через реку Слоновую в посёлок Ольховка, и оттуда ехать в село Васино на попутной машине».

Третье:

Откуда	Куда	Транспорт
Москва	г. Ивановск	Самолёт
г. Ивановск	г. Ореховск	Электричка
г. Ореховск	пос. Ольховка	Паром (р. Слоновая)
пос. Ольховка	с. Васино	Попутная машина

Второе:

«Как ехать в Васино:

1. На самолёте из Москвы до г. Ивановска.
2. На электричке из г. Ивановска до г. Ореховска.
3. На пароме из г. Ивановска через р. Слоновую в пос. Ольховка.
4. На попутной машине из пос. Ольховка до с. Васино».

Четвёртое (рис. 1.7):

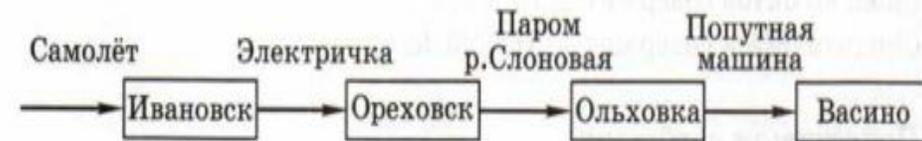


Рис. 1.7

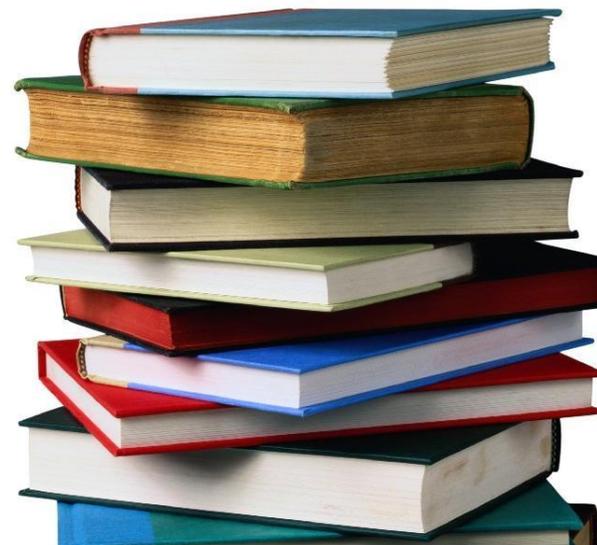
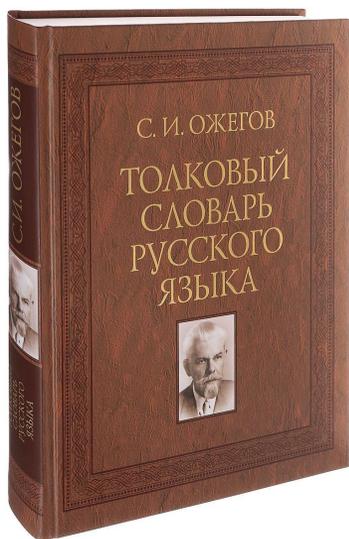
Можно считать, что все эти (такие разные по форме!) сообщения содержат одну и ту же информацию. Какие из них проще воспринимать? Очевидно, что человеку «вытащить» полезную информацию из сплошного текста (первое сообщение) сложнее всего. Во втором случае мы сразу видим все этапы поездки и понимаем, в каком порядке они следуют друг за другом. Третье сообщение (таблицу) и четвёртое (схему) можно понять сразу, с первого взгляда. Второй, третий и четвёртый варианты воспринимаются лучше и быстрее первого, потому что в них выделена структура информации, в которой самое главное — этапы поездки в Васино.

При автоматической (компьютерной) обработке правильно выбранная структура данных облегчает доступ к ним, позволяет быстро найти нужные данные.



Средства, облегчающие поиск информации, знакомы вам по работе с книгами. Самый простой (но очень утомительный!) способ найти в книге то, что нужно, — перелистывать страницу за страницей. Однако в большинстве книг есть оглавление, которое позволяет сразу найти нужный раздел, и это значительно ускоряет поиск.

В словарях слова всегда расставлены в алфавитном порядке (представьте себе, что было бы, если бы они были расположены произвольно!). Поэтому, открыв словарь в любом месте, мы можем сразу определить, куда дальше листать страницы для поиска нужного слова — вперёд или назад.



В больших книгах используют *указатели* (индексы) (рис. 1.8) — списки основных терминов с указанием страниц, на которых они встречаются.

**Оглавление:**

1. Информация . . . . .	5
1.1. Что такое информация? . . . . .	6
1.2. Виды информации . . . . .	8
1.3. Информация в технике . . . . .	11
2. Измерение информации . . . . .	12
2.1. Что такое бит? . . . . .	13
2.2. Байт и другие единицы . . . . .	14

**Словарь:**

автомат — <i>automaton</i>
автор — <i>author</i>
адрес — <i>address</i>
алгебра — <i>algebra</i>
алгоритм — <i>algorithm</i>
архив — <i>archive</i>
архитектура — <i>architecture</i>
асимметрия — <i>asymmetry</i>

**Указатель:**

<b>А</b>
аксиома 45
алгоритм 30, 78
архиватор 125
<b>Б</b>
бит 5, 15, 25, 43
брандмауэр 112
браузер 322

Рис. 1.8

**Структурирование** — это выделение важных элементов в информационных сообщениях и установление связей между ними.

Цели структурирования для человека — облегчение восприятия и поиска информации, выявление закономерностей. При компьютерной обработке структурирование ускоряет поиск нужных данных.

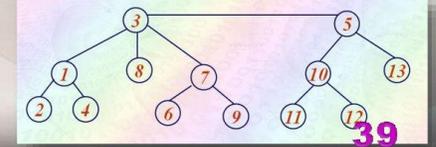
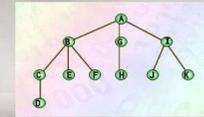


## Структуры данных



### ❖ Дерево –

- нелинейная иерархическая структура данных
- все элементы происходят от одного источника, называемого корнем
- каждый элемент, за исключением корня, имеет единственного предка



С некоторыми структурами данных вы уже знакомы. Например, на уроках математики вы изучали **множество** — некоторый набор элементов. Чтобы определить множество, мы должны перечислить все его элементы (например, множество, состоящее из Васи, Пети и Коли) или определить характерный признак, по

которому элементы включаются в это множество (например, множество драконов с пятью зелёными хвостами или множество точек, в которых функция принимает положительные значения).

Множество может состоять из конечного числа элементов (множество букв русского алфавита), бесконечного числа элементов (множество натуральных чисел) или вообще быть пустым (множество слонов, живущих на Северном полюсе). Множества, с которыми работает компьютер, не могут быть бесконечными, потому что его память конечна.



В документах множество часто оформляют в виде маркированного списка, например:

- процессор;
- память;
- устройства ввода;
- устройства вывода.

В таком списке порядок элементов не важен, от перестановки элементов множество не меняется (рис. 1.9).



Рис. 1.9

**Линейный список** состоит из конечного числа элементов, которые должны быть расположены в строго определённом порядке. В отличие от множества элементы в списке могут повторяться. Список обычно упорядочен (отсортирован) по какому-то правилу, например по алфавиту, по важности, по последовательности действий и т. д. В тексте он часто оформляется как нумерованный список, например:

- 1) надеть носки;
- 2) надеть ботинки;
- 3) выйти из дома.

Переставить местами элементы такого списка нельзя (это будет уже другой список). Список можно задать перечислением элементов, с первого до последнего:

(надеть носки, надеть ботинки, выйти из дома),

а также представить в виде цепочки связанных элементов (рис. 1.10).

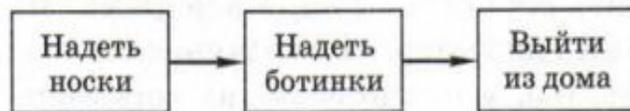
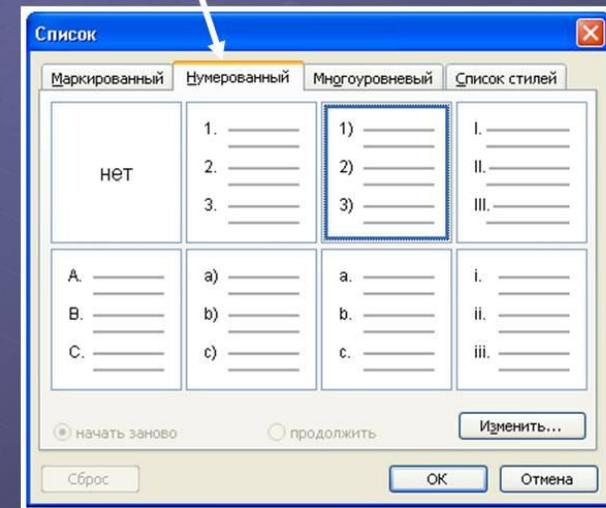


Рис. 1.10

## Нумерованный список

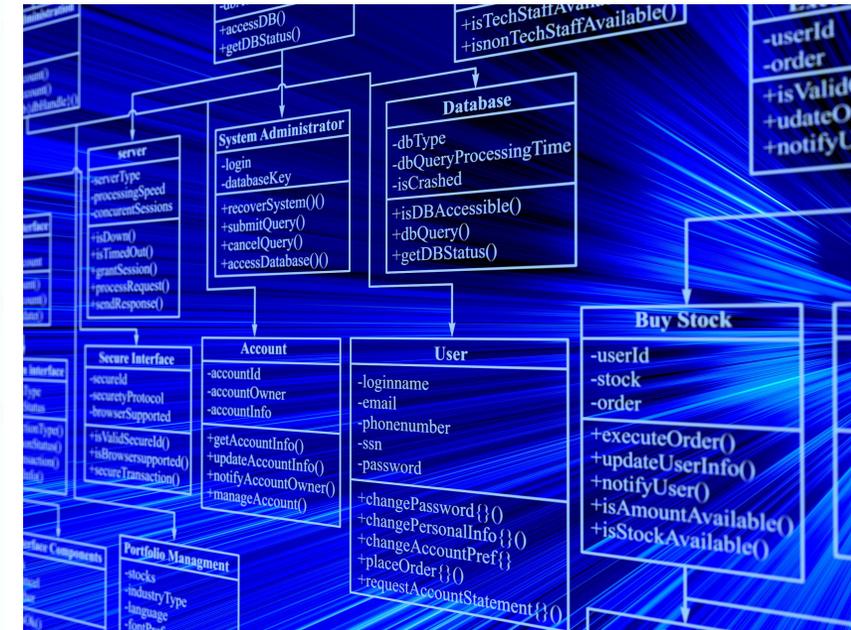


Ещё одна знакомая вам структура — **таблица**. С помощью таблиц устанавливается связь между несколькими элементами. Например, в табл. 1.2 элементы в каждой строке связаны между собой — это свойства некоторого объекта (человека).

Таблица 1.2

Фамилия	Имя	Рост, см	Вес, кг	Год рождения
Иванов	Иван	175	67	1996
Петров	Пётр	164	70	1998
Сидоров	Сидор	168	63	2000

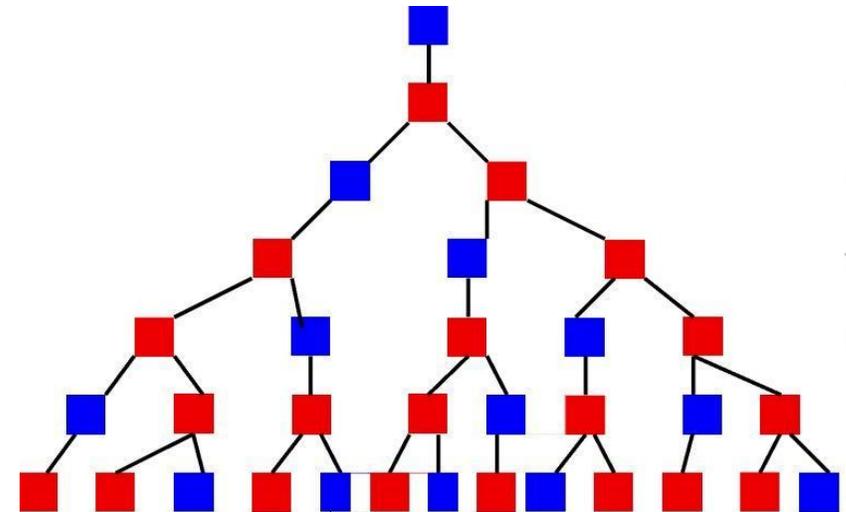
Именно так хранится информация в базах данных: строка таблицы, содержащая информацию об одном объекте, называется *записью*, а столбец (название свойства) — *полем*.



Такая структура, в которой одни элементы «подчиняются» другим, называется **иерархией** (от древнегреческого *íerarchía* — священное правление). В информатике иерархическую структуру называют **деревом**. Дело в том, что, если перевернуть схему на рис. 1.11 вверх ногами, она становится похожа на дерево (точнее, на куст, см. рис. 1.11, б).



Дерево состоит из узлов и связей между ними (они называются дугами). Самый первый узел, расположенный на верхнем уровне (в него не входит ни одна стрелка-дуга), — это **корень** дерева. Конечные узлы, из которых не выходит ни одна дуга, называются **листьями**. Все остальные узлы, кроме корня и листьев, — промежуточные.



Из двух связанных узлов тот, который находится на более высоком уровне, называется **родителем**, а другой — **сыном**. **Корень** — это единственный узел, у которого нет родителя; у листьев нет сыновей.

Используются также понятия **предок** и **потомок**. Потомок какого-то узла — это узел, в который можно перейти по стрелкам от узла-предка. Соответственно, предок какого-то узла — это узел, из которого можно перейти по стрелкам в данный узел.

Типичный пример иерархии — различные *классификации* (животных, растений, минералов, химических соединений). Например, отряд *Хищные* делится на два подотряда: *Псообразные* и *Кошкообразные*. В каждом из них выделяют несколько семейств (рис. 1.13).



Рис. 1.13

Конечно, на рис. 1.13 показаны не все семейства, остальные обозначены многоточиями.

В дереве на рис. 1.12 родитель узла *E* — это узел *B*, а предки узла *E* — это узлы *A* и *B*, для которых узел *E* — потомок. Потомками узла *A* (корня) являются все остальные узлы.

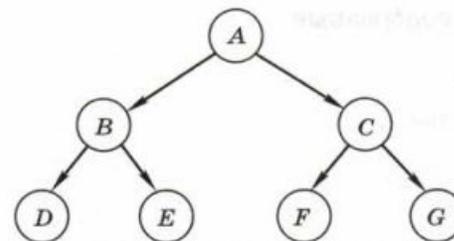


Рис. 1.12

Работая с файлами и папками, мы тоже встречаемся с иерархией: классическая файловая система имеет древовидную структуру<sup>1</sup>. Вход в папку — это переход на следующий (более низкий) уровень иерархии (рис. 1.14).

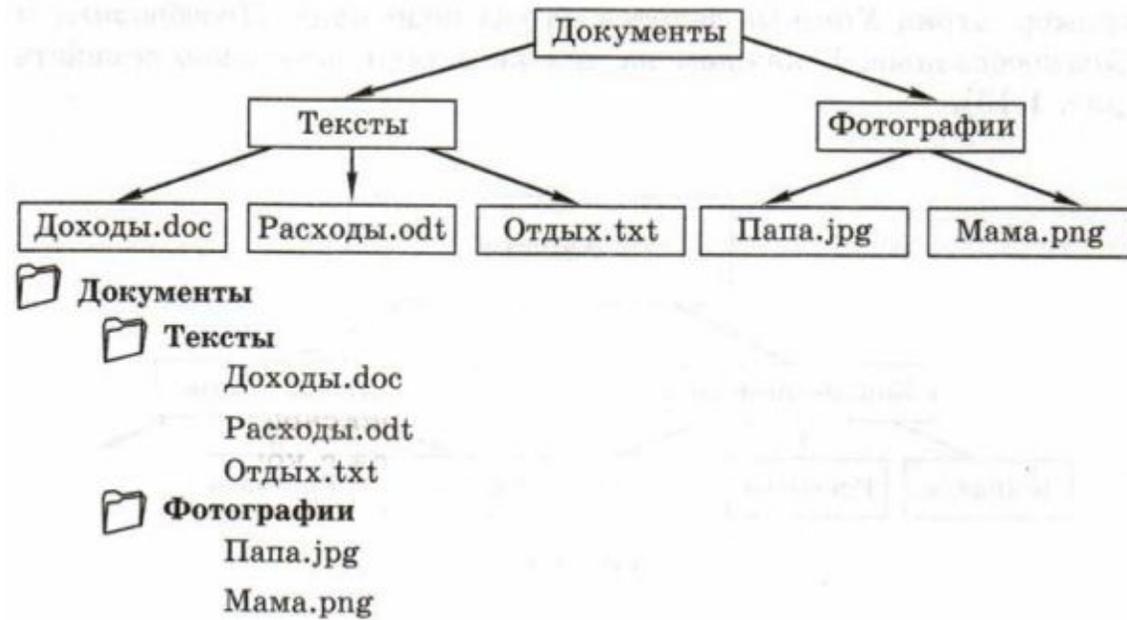
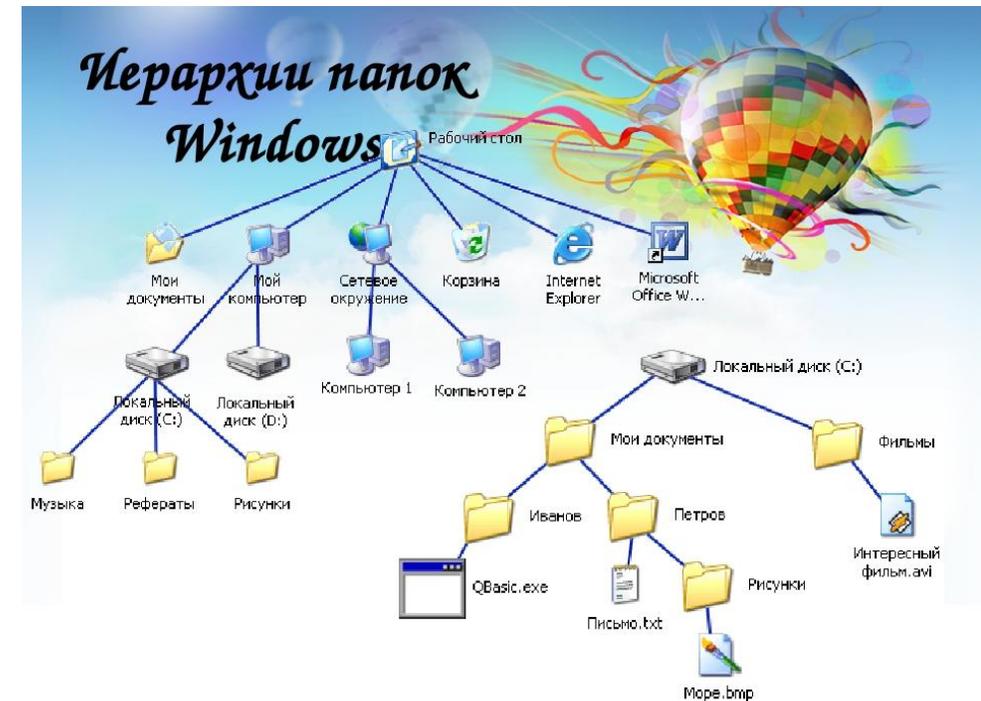


Рис. 1.14



Для хранения информации о вершинах и связях графа, соответствующего схеме на рис. 1.16, можно использовать таблицу (матрицу), показанную на рис. 1.17.

Единица на пересечении строки  $A$  и столбца  $B$  означает, что между вершинами  $A$  и  $B$  есть связь. Ноль указывает на то, что связи нет. Такая таблица называется **матрицей смежности**. Она симметрична относительно главной диагонали (серые клетки в таблице).

На пересечении строки  $C$  и столбца  $C$  стоит единица, которая говорит о том, что в графе есть **петля** — ребро, которое начинается и заканчивается в одной и той же вершине.

Можно поступить иначе: для каждой вершины перечислить все вершины, с которыми связана данная вершина. В этом случае мы получим **список смежности**. Для рассмотренного графа список смежности выглядит так:

$$(A (B, C), B (A, C, D), C (A, B, C, D), D (B, C))$$

Строго говоря, граф — это математический объект, а не рисунок. Конечно, его можно нарисовать на плоскости (например, как на рис. 1.16, б), но матрица смежности и список смежности не дают никакой информации о том, как именно следует располагать узлы друг относительно друга. Для таблицы на рис. 1.17 возможны, например, варианты, показанные на рис. 1.18.

	A	B	C	D
A	0	1	1	0
B	1	0	1	1
C	1	1	1	1
D	0	1	1	0

Рис. 1.17

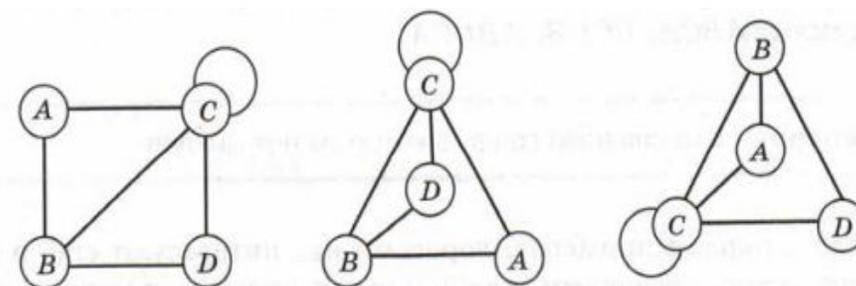


Рис. 1.18

# Графы

Подумайте, как можно структурировать такую информацию:

«От посёлка Васюки три дороги идут в посёлки Солнцево, Грибное и Ягодное. Между Солнцевым и Грибным и между Грибным и Ягодным также есть дороги. Кроме того, есть дорога, которая идет из Грибного в лес и возвращается обратно в Грибное».

Можно, например, нарисовать схему дорог (рис. 1.16, а). На рисунке 1.16, б населённые пункты для краткости обозначены латинскими буквами.

Для исследования таких схем используют графы.

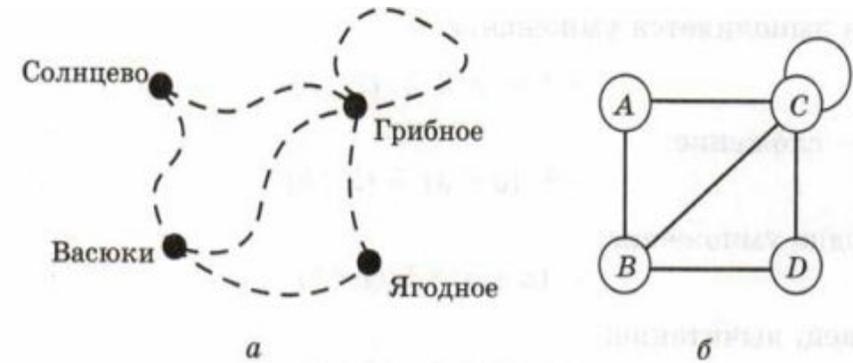
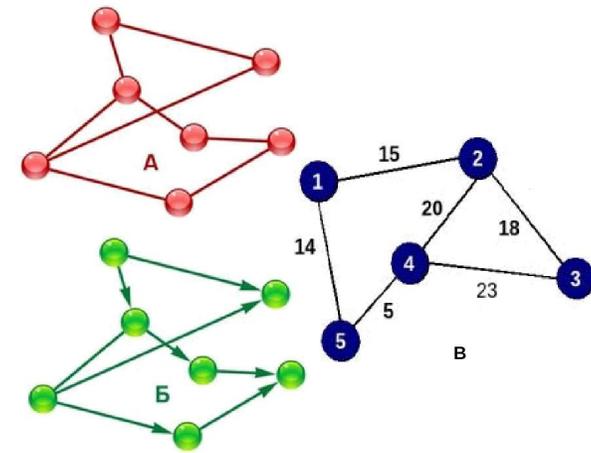


Рис. 1.16

---

**Граф** — это набор вершин и связей между ними (рёбер).

---



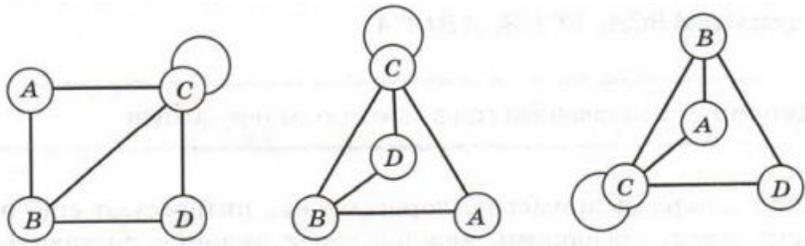


Рис. 1.18

В рассмотренном примере все вершины связаны, т. е. между любой парой вершин существует **путь** — последовательность рёбер, по которым можно перейти из одного узла в другой. Такой граф называется **связным**.

---

**Связный граф** — это граф, между любыми вершинами которого существует путь.

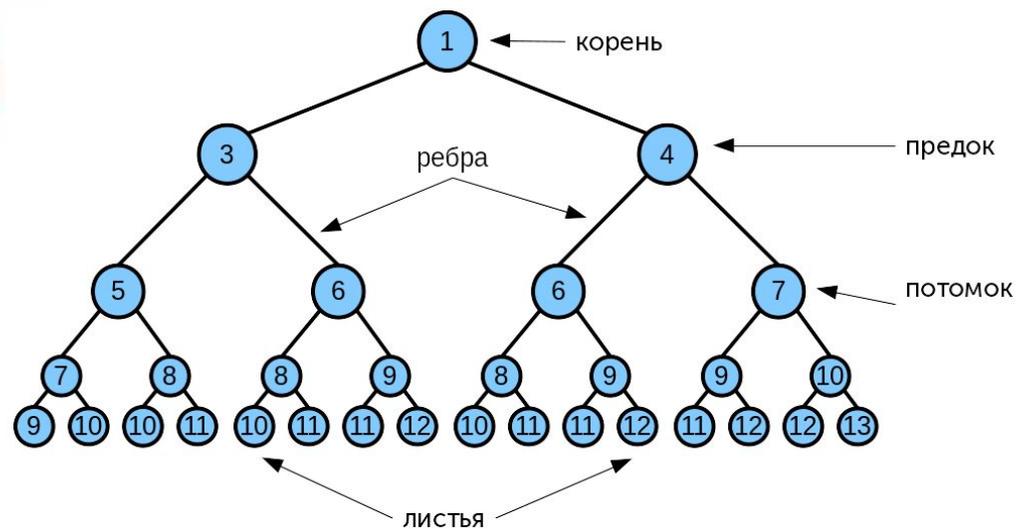
---

Вспоминая материал предыдущего пункта, можно сделать вывод, что дерево — это частный случай связного графа. Но у дерева есть одно важное свойство — в нём нет замкнутых путей (циклов). Граф на рис. 1.17 не является деревом, потому что в нём есть циклы: *ABCA*, *BCDB*, *ABDCA*.

---

**Дерево** — это связный граф, в котором нет циклов.

---



Если в первом примере с дорогами нас интересуют ещё и расстояния между поселками, каждой связи нужно сопоставить число (вес) (рис. 1.20).

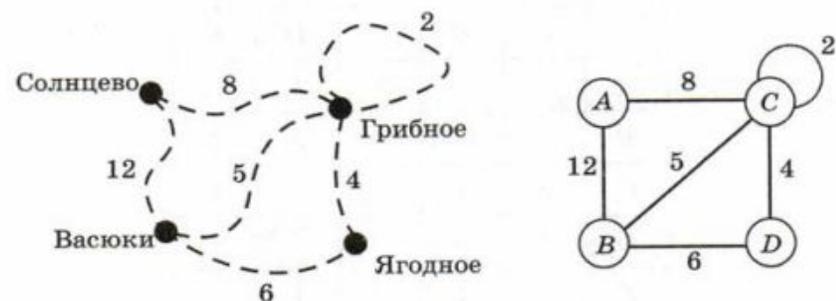


Рис. 1.20

Такой граф называется **взвешенным**, поскольку каждое ребро имеет свой вес. Весом может быть не только расстояние, но и, например, стоимость проезда или другая величина.

Как хранить информацию о таком графе? Ответ напрашивается сам собой — нужно в таблицу записывать не 1 или 0, а вес ребра. Если связи между двумя вершинами нет, на бумаге можно оставить ячейку таблицы пустой, а при хранении в памяти компьютера записывать в неё условный код, например  $-1$ . Такая таблица называется **весовой матрицей**, потому что содержит веса рёбер. В данном случае она выглядит, как показано на рис. 1.21.

	A	B	C	D
A		12	8	
B	12		5	6
C	8	5	2	4
D		6	4	

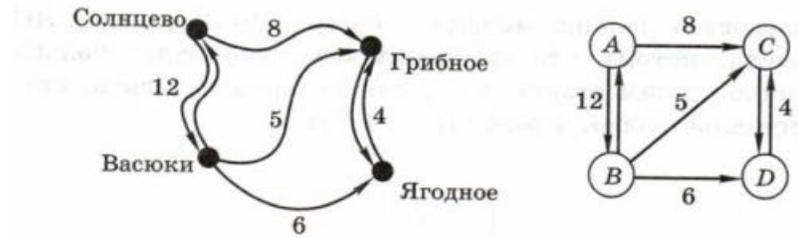
Рис. 1.21

Так же как и матрица смежности, весовая матрица симметрична относительно главной диагонали. Нижняя ячейка в столбце A и верхняя в столбце D говорят о том, что между вершинами A и D нет ребра.

Наверное, вы заметили, что при изображении деревьев, которые описывают иерархию (подчинение), мы ставили стрелки от верхних уровней к нижним. Это означает, что для каждого ребра указывается направление, и двигаться можно только по стрелкам, но не наоборот. Такой граф называется **ориентированным** (или коротко **орграфом**). Он может служить, например, моделью системы дорог с односторонним движением. Матрица смежности и весовая матрица для орграфа уже не обязательно будут симметричными.

На схеме на рис. 1.27 всего две дороги с двусторонним движением, по остальным можно ехать только в одну сторону.

Рёбра в орграфу называют дугами. Дуга, в отличие от ребра, имеет начало и конец.



	A	B	C	D
A		12	8	
B	12		5	6
C				4
D			4	

Рис. 1.27

### Списки, графы, деревья и таблицы

Таблица	Неориентированный	Ориентированный
Невзвешенный		
Взвешенный		



# Контрольные вопросы

1. Что такое структурирование информации? Зачем оно нужно?
2. Приведите примеры множеств.
3. Какими способами можно задать множество? Что такое пустое множество?
4. Чем отличаются множество и линейный список?
5. Что такое иерархия? Приведите примеры.
6. Как называется иерархическая структура в информатике?
7. Что такое корень, лист, родитель, сын, предок, потомок в структуре «дерево»?
8. Какой граф называется связным?
9. Что такое петля? Как по матрице смежности определить, есть ли петли в графе?
10. Что такое орграф?
11. В чем разница между понятиями «ребро» и «дуга»?
12. Что такое взвешенный граф?