Лекция 8
Отношения, графы, деревья: основные определения

Дополнительно:

учебное пособие Нестеренко Т.В., Чурина Т.Г. Основы алгоритмизации и программирования (часть 2). Динамические структуры данных, алгоритмы на графах.

Стр. 34-42, 45-47

Определение. Пусть a и b — объекты.

Через (a, b) обозначим *упорядоченную пару,* состоящую из объектов a и b, взятых в этом порядке.

Упорядоченные пары (a, b) и (c, d) называются *равными,* если $a = c$ и $b = d$.

В противоположность этому $\{a, b\} = \{b, a\}$.

Определение. *Декартовым произведением* множеств A и B, обозначаемым через $A \times B$, называют множество $\{(a, b) \mid a \in A \text{ и } b \in B\}$.

Пример. Пусть $A = \{1, 2\}$ и $B = \{2, 3, 4\}$. Тогда

$A \times B = \{(1, 2), (1, 3), (1, 4), (2, 2), (2, 3), (2, 4)\}$

Отношения

Определение. Пусть A и B —множества.

Отношением из A в B называется любое подмножество множества AxB.

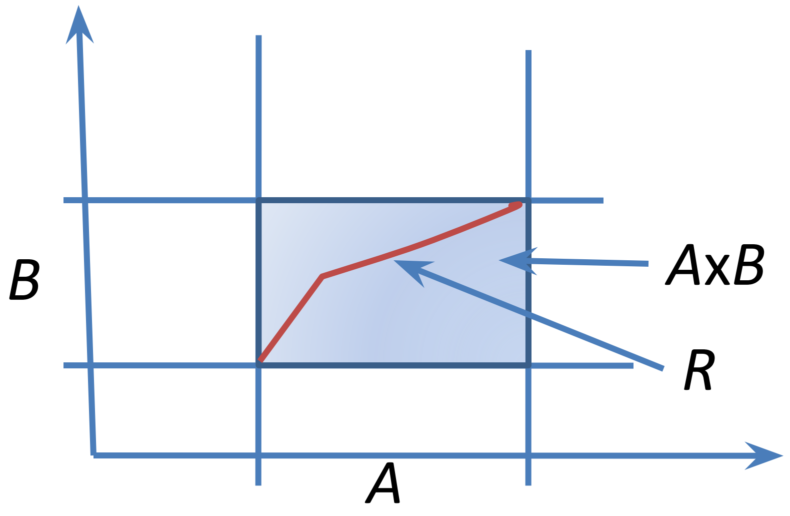
Если $A = B$, то говорят, что отношение задано, или определено, *на A* (или просто, что это — *отношение на множестве* A).

Если R — отношение из A в B и (а, b)$\in R$, то пишут aRb.

A — *область определения* отношения R,

B —множество его значений.

Определение. Отношение $\{(b, a) \mid (a, b) \in R\}$ называется *обратным* к отношению R и часто обозначается через R^{-1}.



Свойства отношений

Определение. Пусть A—множество и R — отношение на A. Отношение R называется

- *рефлексивным,* если aRa для всех a из A,
- *симметричным,* если aRb влечет bRa для a и b из A,
- *транзитивным,* если aRb и bRc влекут aRc для a, b и c из A. Элементы a, b и c не обязаны быть различными.

Рефлексивное, симметричное и транзитивное отношение называется *отношением эквивалентности*.

Важное свойство любого отношения эквивалентности R, определенного на множестве A, заключается в том, что оно разбивает множество A на непересекающиеся подмножества, называемые *классами эквивалентности*.

Графы

В математической теории графов и информатике ***граф*** — это совокупность двух конечных множеств: множества точек, которые называются вершинами, и множества пар вершин, которые называются ребрами .

Для разных областей применения виды графов могут различаться направленностью, ограничениями на количество связей и дополнительными данными о вершинах или рёбрах.

Графы

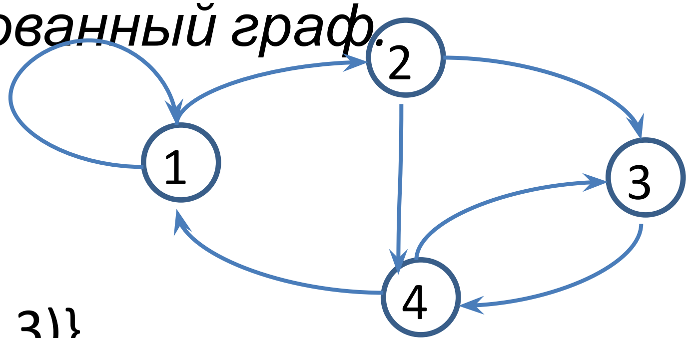
Система – это целое, состоящее из объектов, взаимосвязанных между собой (человек, книга, обучение в школе и т.д.).

Граф – это средство для наглядного представления состава и структуры системы. Обычно - это набор узлов (вершин) и связей между ними (ребер).

Неупорядоченный граф G — это пара *(A, R)*, где *A* — множество элементов, называемых *вершинами* (или *узлами)*, а *R* — отношение на множестве *A.*

Если *R* — несимметричное отношение, то *G* — *ориентированный граф;*

R — симметричное, то *G* —не*ориентированный граф.*



Пример.

A={1, 2, 3, 4} ,

R = {(1, 1), (1, 2), (2, 3), (2, 4), (3, 4), (4, 1), (4, 3)}.

Пара (а, *b*)$\in$*R* называется *дугой* (или *ребром*) графа *G*.

Говорят, что дуга *выходит* из вершины *а*
и *входит* в вершину *b.*

Если (*а*, *b*) — дуга, то говорят,
что вершина *а* *предшествует* вершине *b,*
а вершина *b* *следует* *за* вершиной *a.*

Вершина b смежна с вершиной a, если дуга выходит из a и входит в b .

Графы

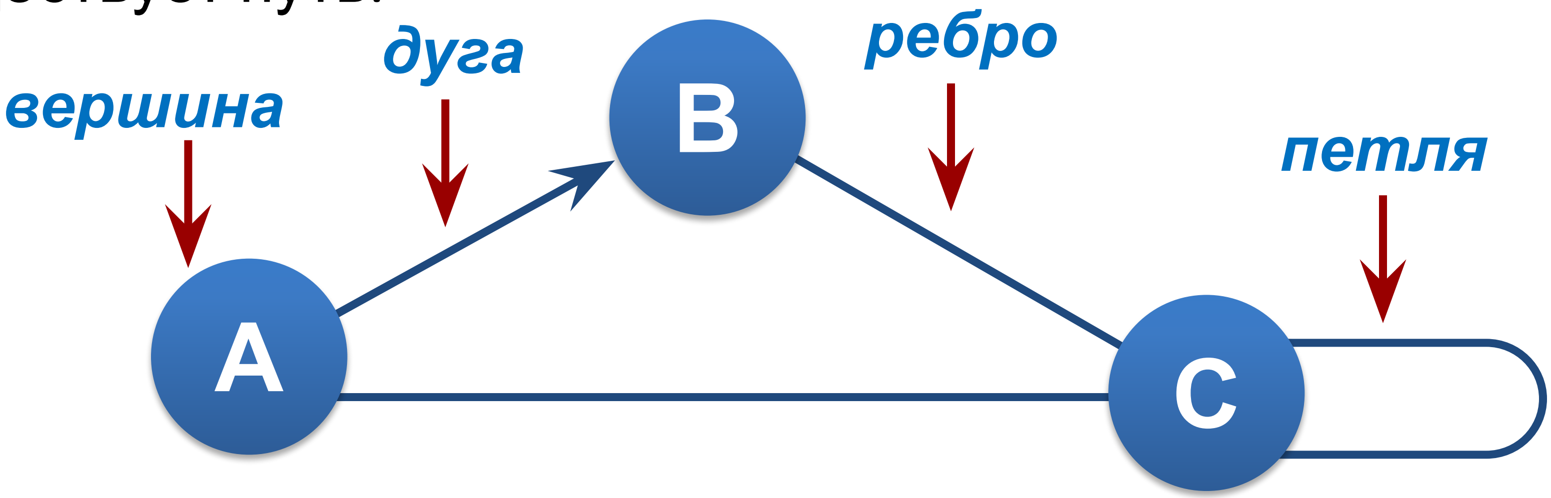
Граф состоит из ***вершин***, связанных линиями.

Направленная линия (со стрелкой) называется ***дугой***.

Линия ненаправленная (без стрелки) называется ***ребром***.

Линия, выходящая из некоторой вершины и входящая в неё же, называется ***петлей***.

Связный граф – это граф, между любыми вершинами которого
существует путь.

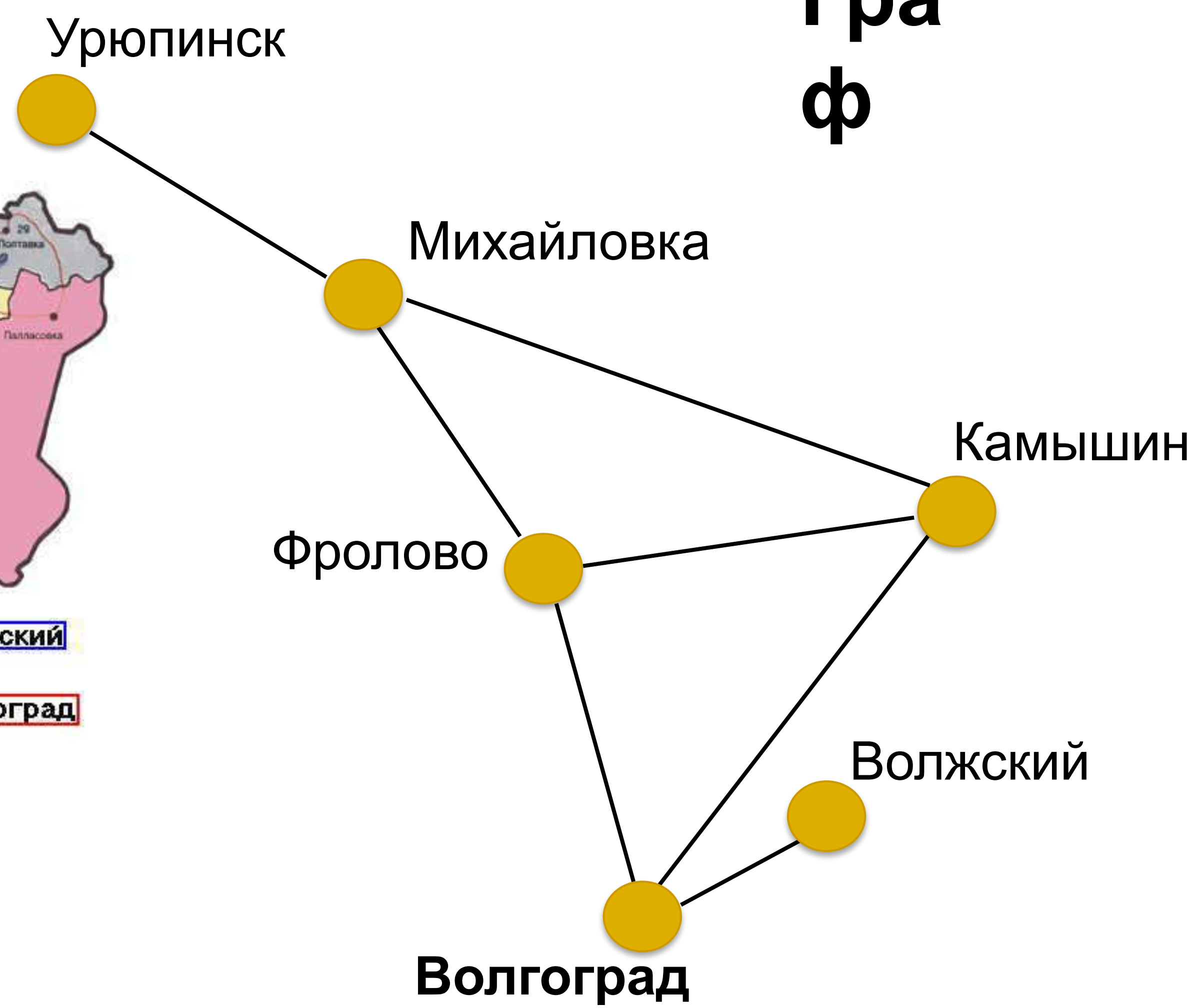


Обработка данных в виде графов

- Графы формально описывают множество близких ситуаций.
- Самым привычным примером служит карта автодорог, на которой изображены перекрестки и связывающие их дороги. Перекрестки являются вершинами графа, а дороги — его ребрами.
- Графы могут быть ориентированы (подобно улицам с односторонним движением) или взвешены, когда каждой дороге приписана стоимость путешествия по ней (если, например, дороги платные).

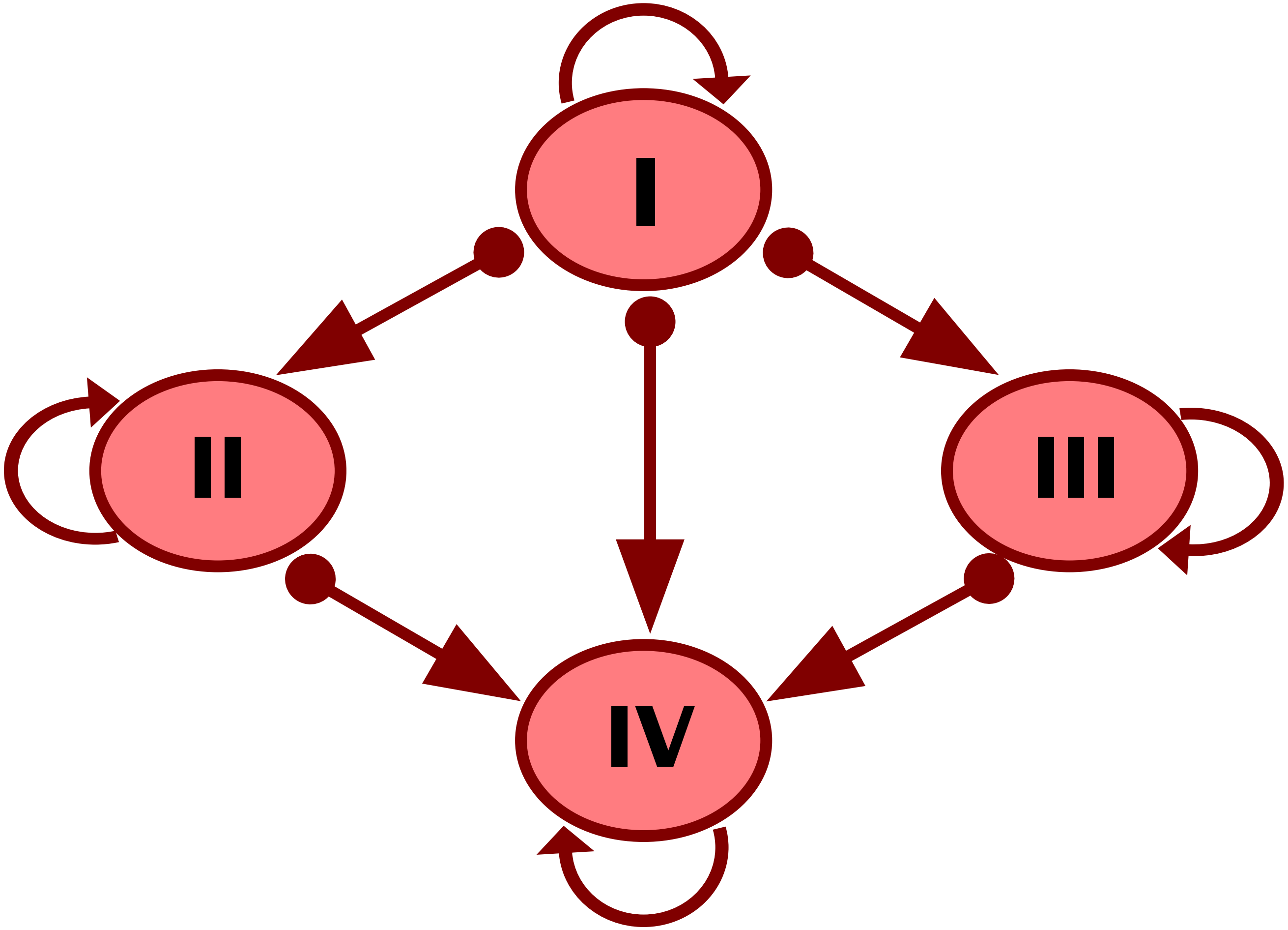
Обработка данных в виде графов

Гра ф



Карта Волгоградской области

ПЕРЕЛИВАНИЕ КРОВИ



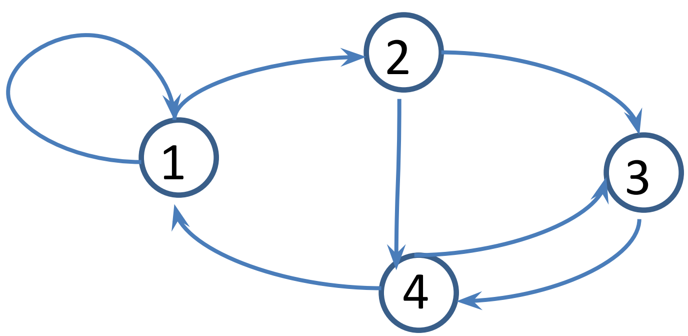
Определения.

Последовательность вершин $(a_0, a_1, \ldots, a_n)$, $n \geq 1$, называется *путем* (или *маршрутом*) *длины* n из вершины a_0 в вершину a_n, если для каждого $1 \leq i \leq n$ существует дуга, выходящая из вершины a_{i-1} и входящая в вершину a_i.

Если существует путь из вершины a_0 в вершину a_n, то говорят, что a_n *достижима* из a_0.

Циклом называется путь $(a_0, a_1, \ldots, a_n)$, в котором $a_0 = a_n$.

Граф называется *сильно связным*, если для любых двух разных вершин a и b существует путь из a в b.



(1, 2, 4,3) - путь из 1 вершины в 3

(1, 2, 3, 4, 1) – цикл, проходящий через вершины 1, 2, 3, 4

Неориентированный граф

Неориентированный граф - граф, вершины которого соединены ребрами.

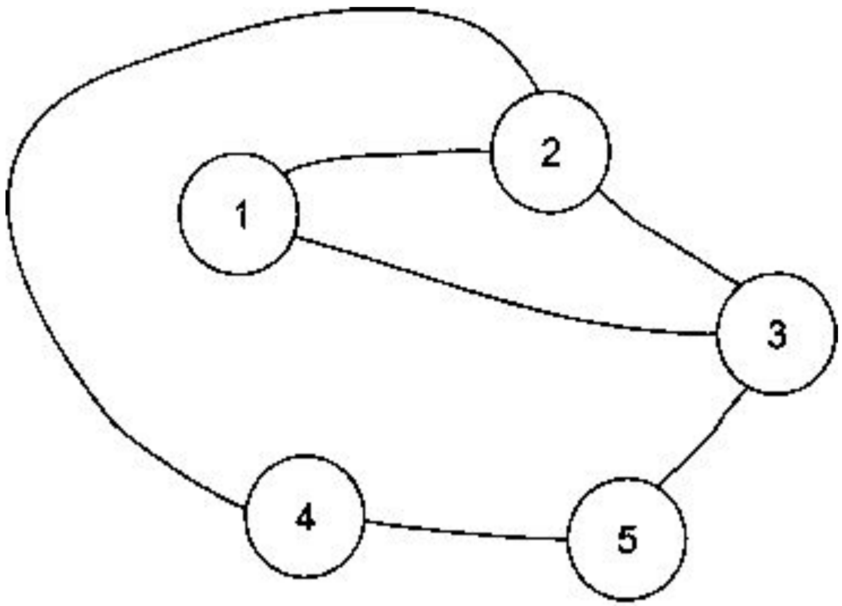
С помощью таких графов могут быть представлены схемы двухсторонних (симметричных) отношений.

<u>Цепь</u> – путь по вершинам и ребрам, включающий любое ребро графа **не более одного раза**.

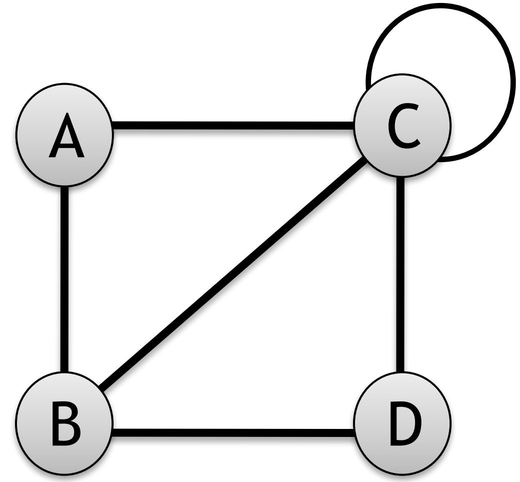
<u>Цикл</u> – цепь, начальная и конечная вершины которой совпадают.

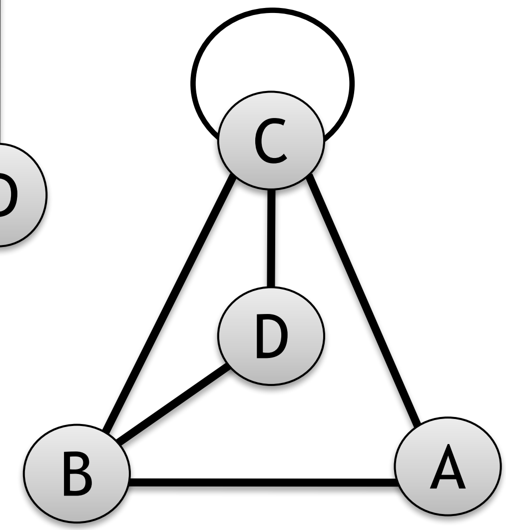
Граф с циклом называют ***сетью***. Это набор узлов (вершин) и связей между ними (ребер).

Неориентированный граф



Граф $G = (\{1, 2, 3, 4, 5\}, \{\{1, 2\}, \{1, 3\}, \{2, 4\}, \{3, 5\}, \{2, 3\}, \{4, 5\}\})$

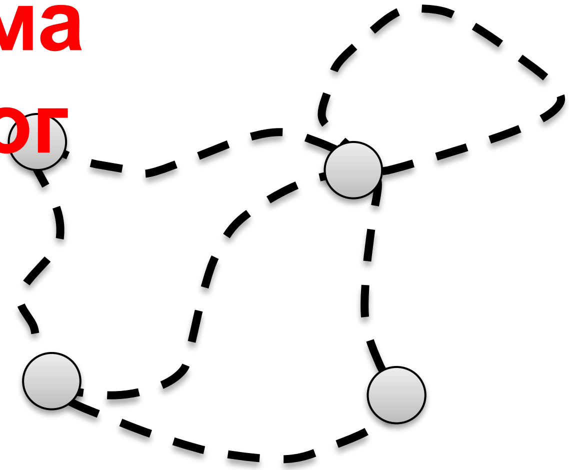




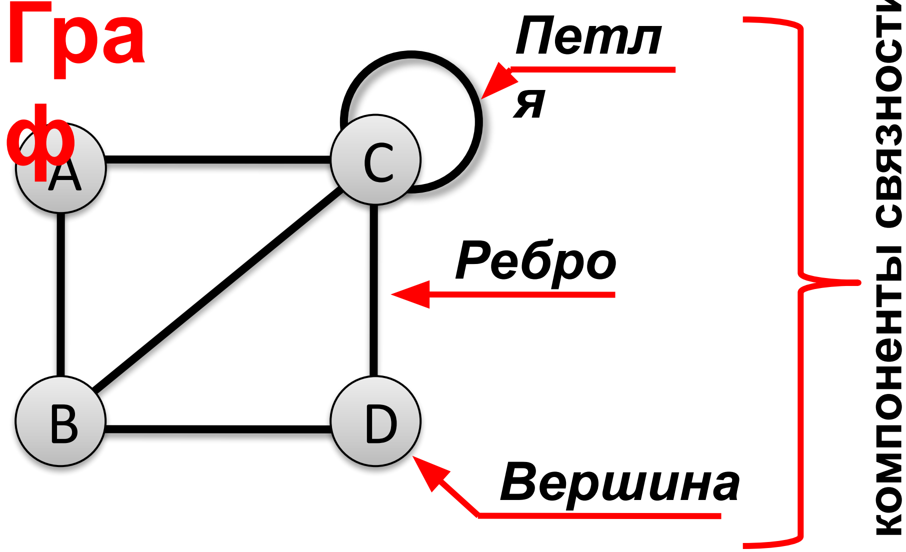
- Граф может быть ориентированным или нет.
- Ребра неориентированного графа, чаще всего называемого просто графом, можно проходить в обоих направлениях.
- В этом случае ребро — это неупорядоченная пара вершин, его концов.

Обработка данных в виде графов

Схема дорог



Граф



Список смежности неоринтированного Графа

(**A**(B, C),
B(A, C, D),
C(A, B, **C**, D),
D(B, C))

Матрица смежности неоринтированного графа

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

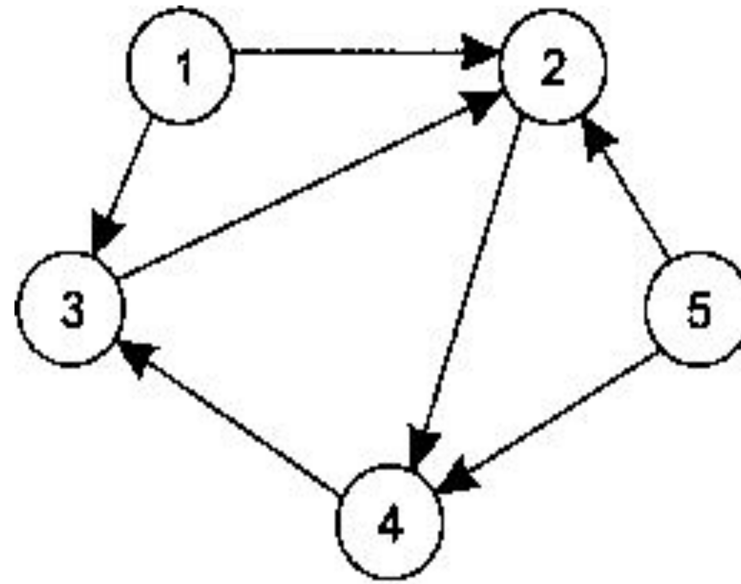
Петля

Ориентированный граф (орграф)

Ориентированный граф (орграф) - граф, вершины которого соединены дугами.

С помощью таких графов могут быть представлены схемы односторонних отношений.

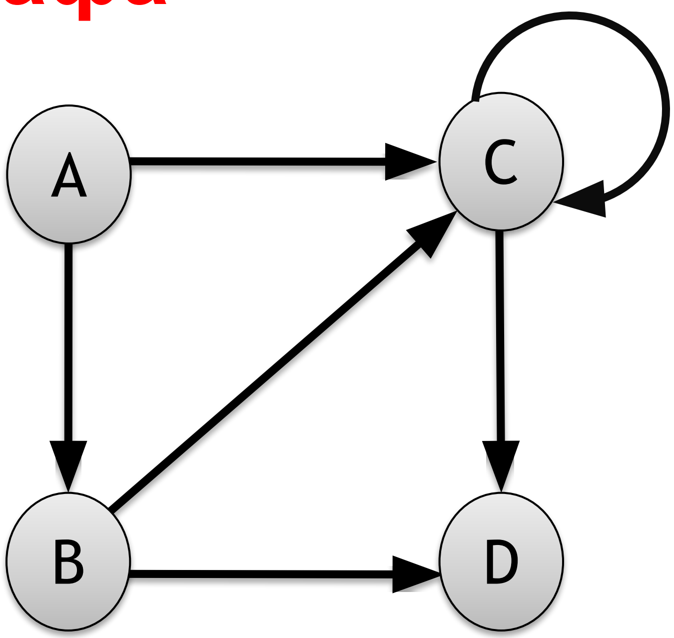
В ориентированном графе, или орграфе, ребра представляют собой упорядоченные пары вершин: первая вершина — это начало ребра, вторая — его конец



Ориентированный граф G = ({ 1, 2, 3, 4, 5 }, { (1, 2), (1, 3), (2, 4), (3, 2), (4, 3), (5, 2), (5, 4) })

Ориентированный граф (орграф)

Матрица смежности оринтированного графа



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

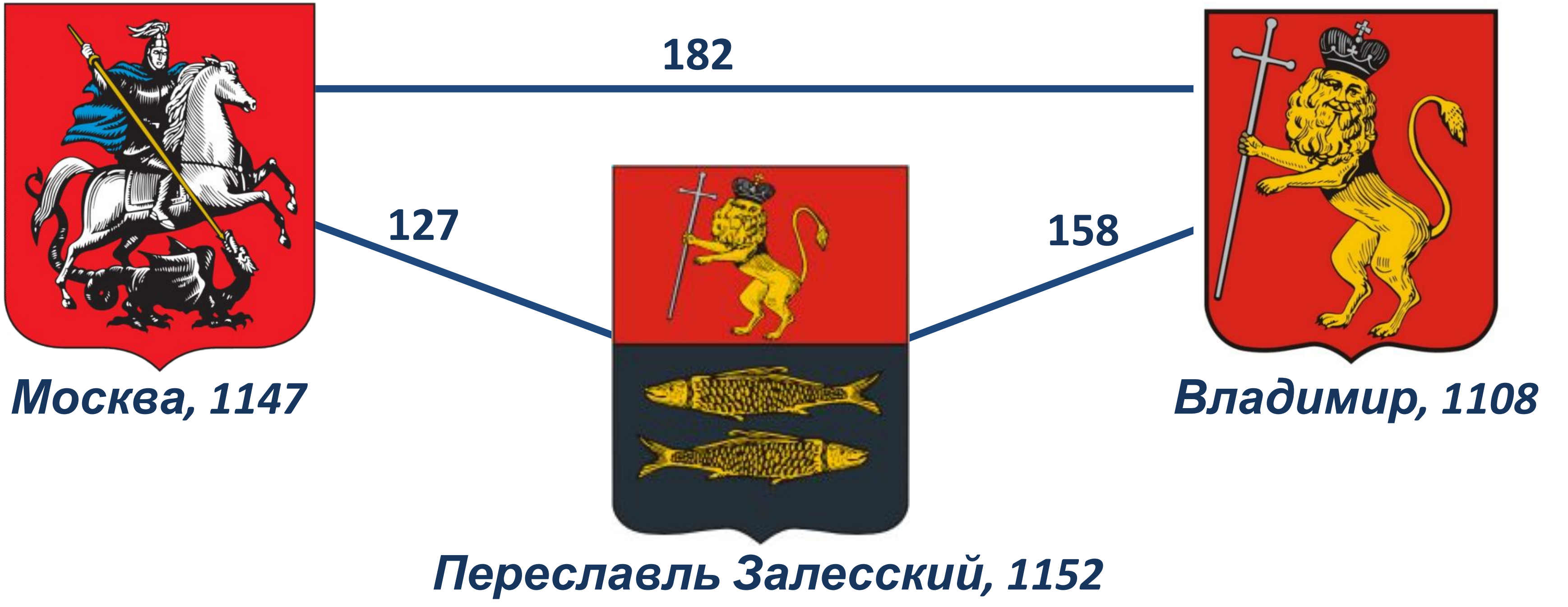
Может быть несимметрична!

Полный граф

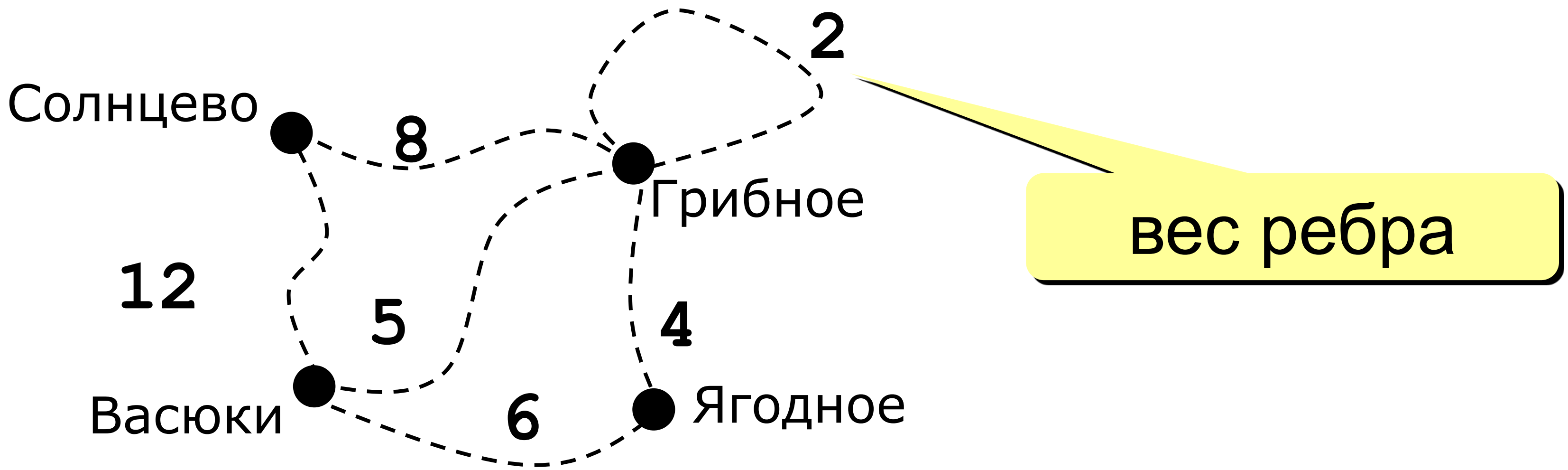
- это граф, в котором каждая вершина соединена со всеми остальными.
- Количество ребер в полном графе без петель с N вершинами равно (N2 - N)/2.
- В полном ориентированном графе разрешается переход из любой вершины в любую другую. Поскольку в графе переход по ребру разрешается в обоих направлениях.

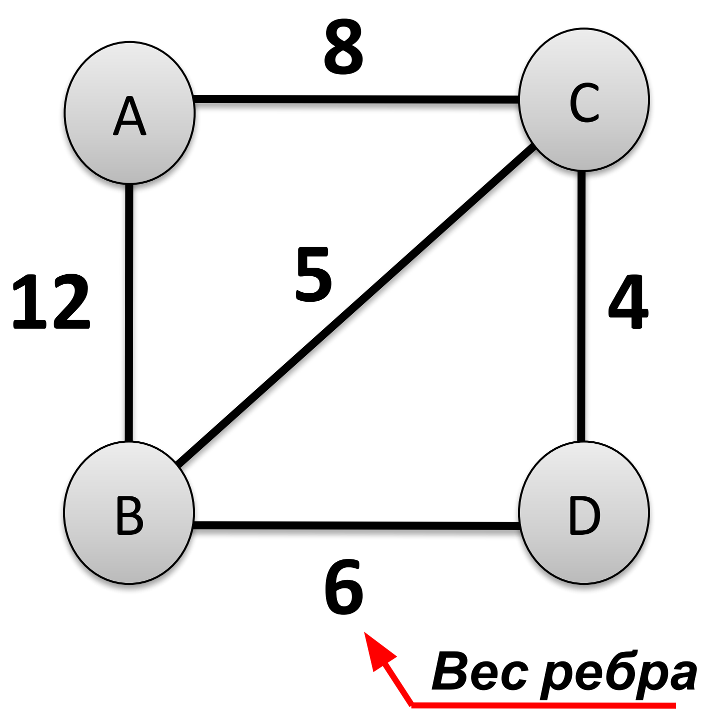
Взвешенный граф

Взвешенный граф - граф, у которого вершины или рёбра (дуги) несут дополнительную информацию (вес).



Взвешенный граф

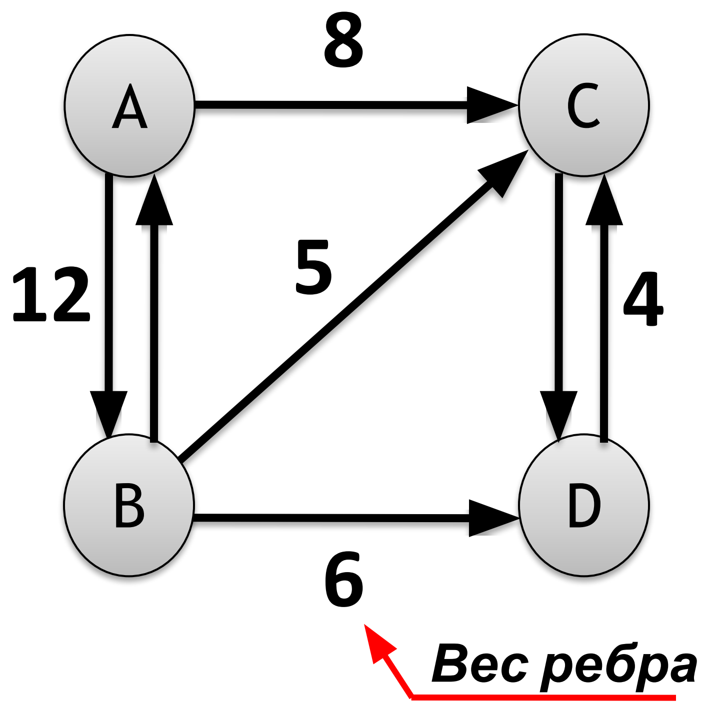




Весовая матрица

	A	B	C	D
A		12	8	0
B	12		5	6
C	8	5		4
D	0	6	4	

Взвешенный орграф



Весовая матрица

Она может быть несим

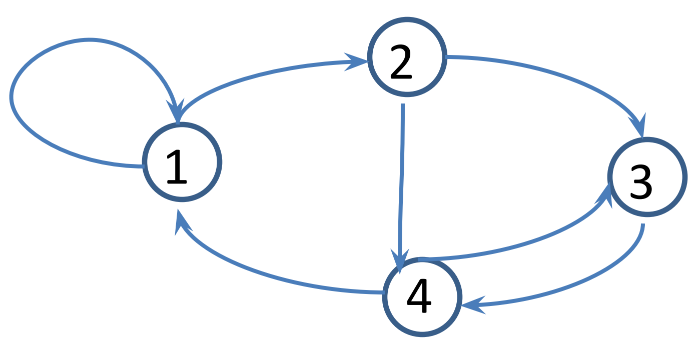
	A	B	C	D
A		12	8	
B	12		5	6
C				4
D			4	

Степень вершины

Степенью по входу (*полустепенью входа*) вершины ***а*** назовем число входящих в нее дуг,

а *степенью по выходу* (*полустепенью исхода*)— число выходящих из нее дуг.

Если граф неориентированный, то *степень* вершины – это количество ребер, связанных с ней.



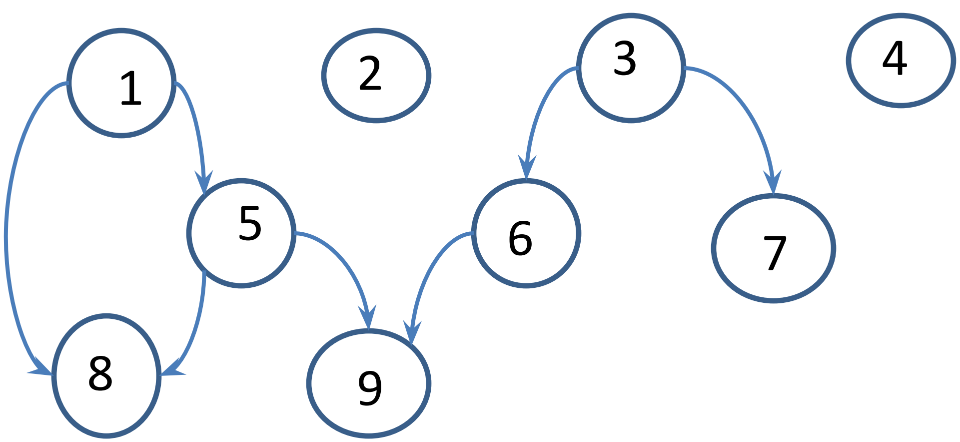
Для вершины 2:
полустепень входа = 1
полустепень исхода = 2

Ациклические графы

Ациклическим графом называется (ориентированный) граф, не имеющий циклов.

Вершину, степень по входу которой равна 0, назовем *базовой*.

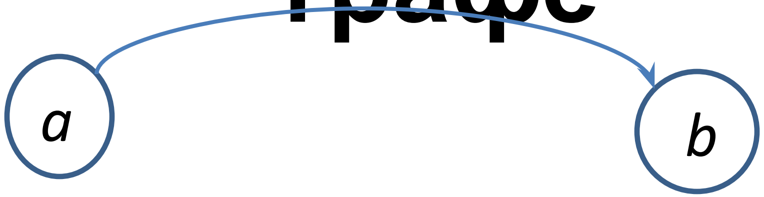
Вершину, степень по выходу которой равна 0, назовем *листом* (или *концевой* вершиной).



Базовые вершины: 1, 2, 3, 4

Листья – 2, 4, 8, 9, 7

Дуга и путь в ациклическом графе



Если (*a*, *b*) – дуга в ациклическом графе,

то *a* – прямой предок *b*,

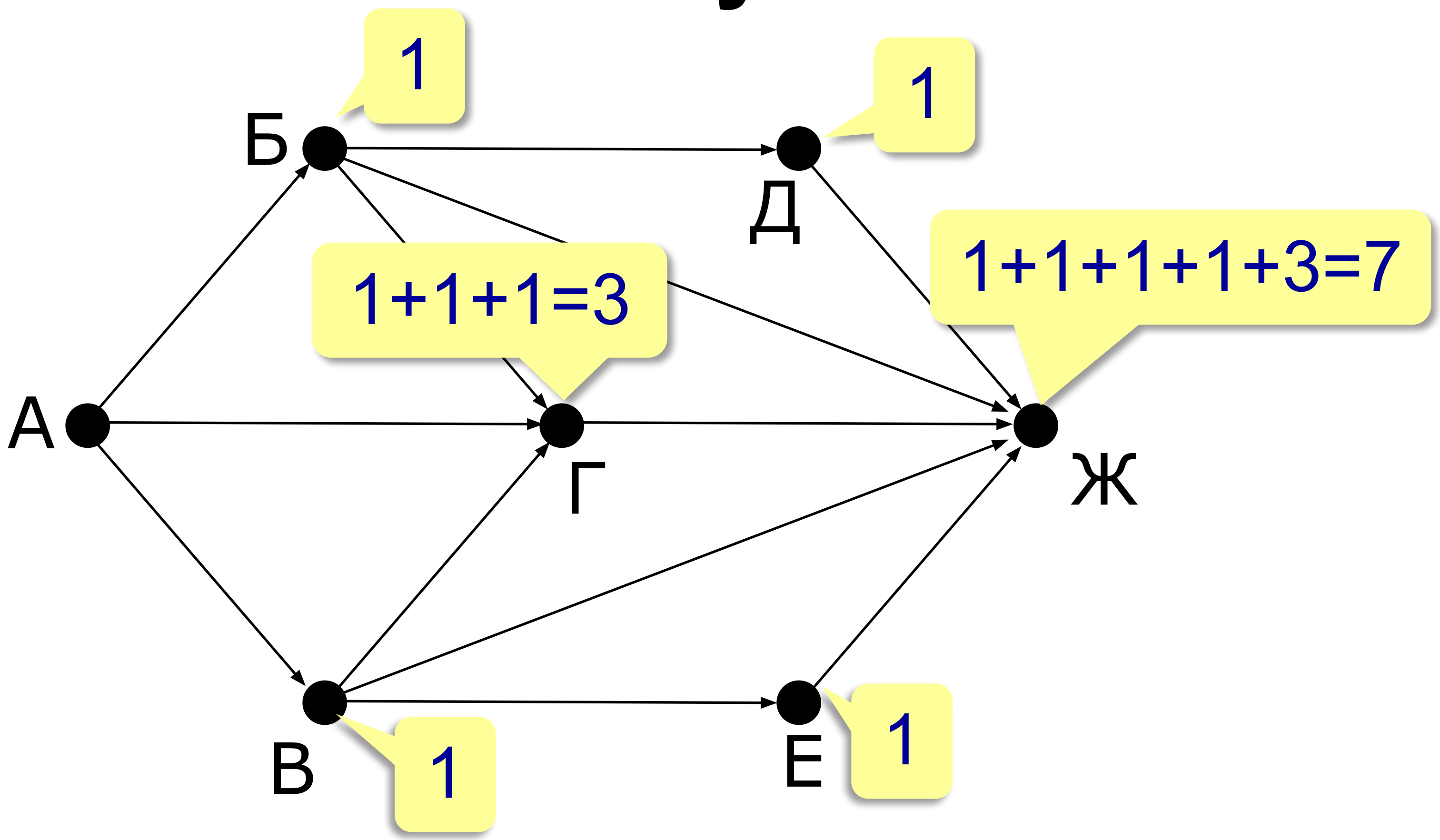
b – прямой потомок *a*.

Если в ациклическом графе существует путь из *a* в *b*,

то *a* – предок *b*,

b – потомок *a*.

Количество путей из А в Ж



$$N_Ж = N_Д + N_Б + N_Г + N_В + N_Е$$

Работа с графом

- При работе с графами часто приходится выполнять некоторое действие по одному разу с каждой из вершин графа, например, когда некоторую порцию информации следует передать каждому из компьютеров в сети. При этом нерационально посещать какой-либо компьютер дважды.

- Аналогичная ситуация возникает, если нужно собирать, а не распространять информацию.

- Подобный обход можно совершать двумя способами. При обходе в глубину проход по выбранному пути осуществляется настолько глубоко, насколько это возможно, а при обходе по уровням происходит равномерное движение вдоль всех возможных направлений.

Обход в глубину

- При обходе в глубину происходит посещение первого узла, а затем передвижение вдоль ребер графа, пока не будет достигнут тупик. Узел неориентированного графа является тупиком, если уже посещены все примыкающие к нему узлы. В ориентированном графе тупиком также оказывается узел, из которого нет выходящих ребер.
- После попадания в тупик следует вернуться назад вдоль пройденного пути, пока не будет обнаружена вершина, у которой есть еще не посещенный сосед, а затем двигаться в новом направлении. Процесс оказывается завершенным после того, как произошел возврат в отправную точку, а все примыкающие к ней вершины уже посещены.

Обход по уровням

- При обходе графа по уровням после посещения первого узла происходит обход всех соседних с ним вершин.
- При втором проходе посещаются все вершины на расстоянии «двух ребер» от начальной.
- При каждом новом проходе обходятся вершины, расстояние от которых до начальной на единицу больше предыдущего.
- В графе могут быть циклы, поэтому не исключено, что одну и ту же вершину можно соединить с начальной двумя различными путями.
- При обходе этой вершины впервые проходится самый короткий до нее путь, и посещать ее второй раз нет необходимости. Поэтому, чтобы предупредить повторное посещение, приходится либо вести список посещенных вершин, либо сопоставить каждой вершине флажок,

Типы моделей на графах

1. **Иерархия** (дерево). Принцип связи – «один ко многим». Фактически - это расположение частей или элементов целого в порядке от высшего к низшему.

2. **Сеть**. Принцип связи – «многие ко многим».

Сетевая модель (сетевой график) — ориентированный граф, вершины которого отображают состояния (характеристики) некоторого объекта (например, строительного объекта, дорожной сети и т.д.), а дуги — работы (процессы), связанные с этим объектом. Каждой дуге соответствует показатель (время, расстояние и т.д.), характеризующий работу (процесс).

Типы моделей на графах

1. **Иерархия** (дерево). Принцип связи – «один ко многим». Фактически - это расположение частей или элементов целого в порядке от высшего к низшему.

2. **Сеть**. Принцип связи – «многие ко многим».

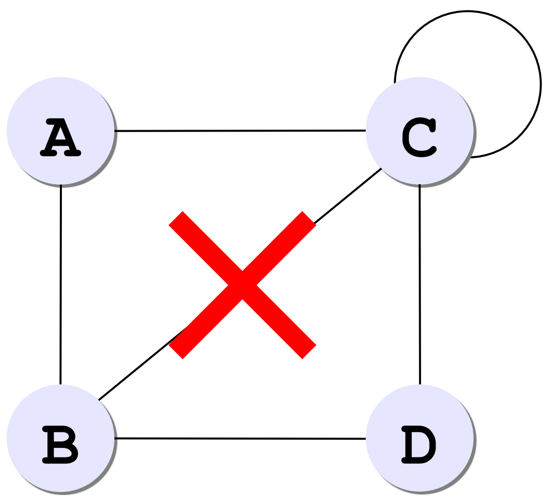
Дерево – это граф иерархической структуры. Между любыми двумя его вершинами существует единственный путь. Дерево не содержит циклов и петель.

Дерево – это связный граф без циклов (замкнутых путей).

Типы моделей на графах

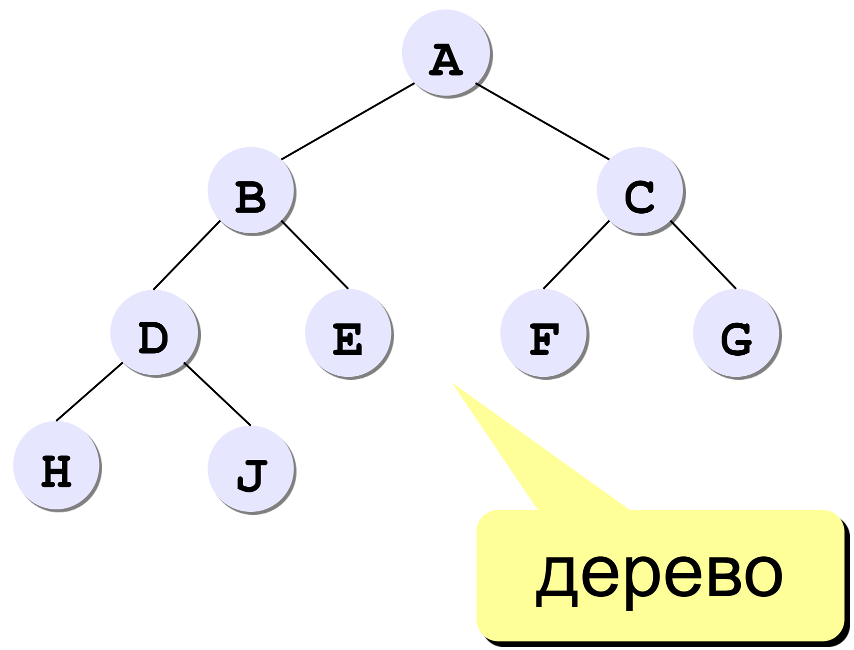
Дерево – это граф иерархической структуры. Между любыми двумя его вершинами существует единственный путь. Дерево не содержит циклов и петель.

Дерево – это связный граф без циклов (замкнутых путей).



ABC ABDC

BCD CCC…



Дерево
(частный вид ациклического графа)

Определение. *(Ориентированным) деревом T* называется (ориентированный) граф *G = (A,R)* со специальной вершиной *r*$\in$ *A*, называемый *корнем*, у которого

- степень по входу вершины *r* равна 0,
- степень по входу всех остальных вершин дерева *T* равна 1,
- каждая вершина *a*$\in$*A* достижима из *r*.

Дерево *T* обладает следующими свойствами:

- *T—ациклический граф,*
- ***для каждой вершины дерева T существует единственный путь, ведущий из корня в эту вершину.***

Корень – главная вершина дерева.
Предок – объект верхнего уровня.
Потомок – объект нижнего уровня.
Листья – вершины, не имеющие потомков.

Дерево
(частный вид ациклического графа)

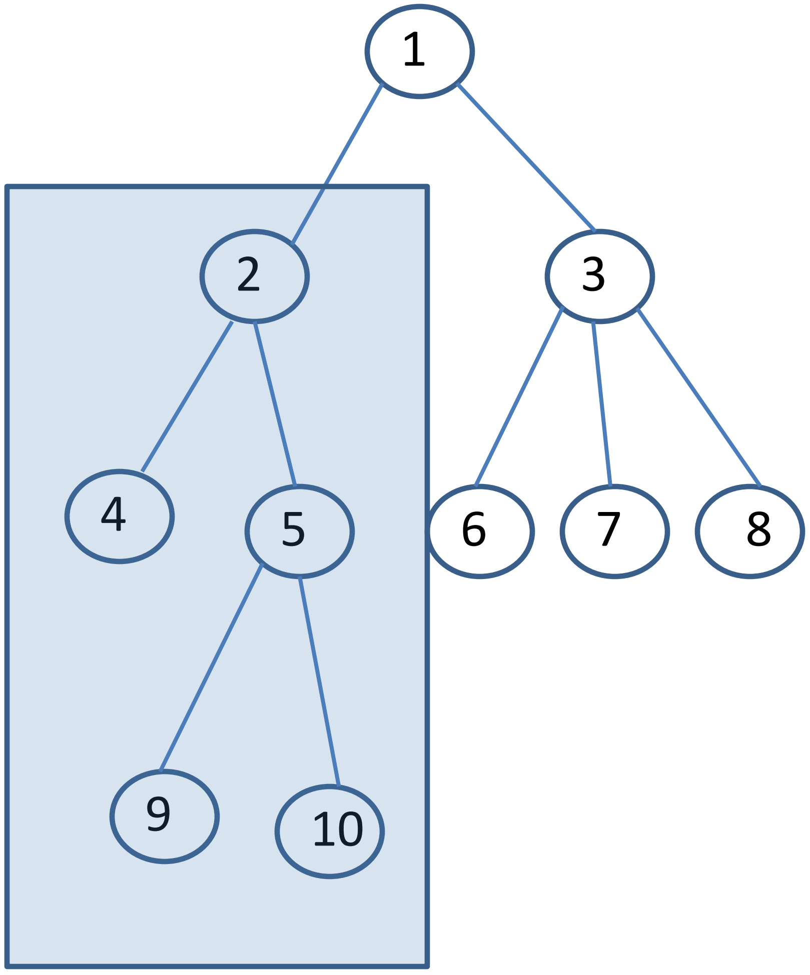
- Корень - это единственный узел, не имеющий непосредственного предка. Это главная вершина дерева.
- Имеется множество типов деревьев. Важнейшим с точки зрения информатики подмножеством структуры типа дерева является подмножество бинарных деревьев поиска.
- У бинарного дерева каждый узел имеет не более двух дочерних узлов, причем левый и правый узлы различаются.
- Каждый узел содержит несколько полей: поля значения, хранящегося в узле, и полей, указывающих на левый и правый потомки данного узла, а также на родительский узел.

Поддеревом дерева $T = (A, R)$ называется любое дерево $T' = (A', R')$, у которого

1) A' не пусто и содержится в A,
2) $R' = (A' \times A') \cap R$,
3) ни одна вершина из множества $A \setminus A'$ не является потомком вершины из множества A'.

Другими словами, поддерево – это дерево с корнем в выделенной вершине и все вершины и дуги, достижимые из нее.

Ориентированный граф, состоящий из нескольких деревьев, называется *лесом*.



Пусть $T=(A, R)$ – дерево, $(a, b) \in R$, тогда

a – *отец* b, а b – *сын* a.

Глубина или уровень вершины – длина пути от корня до этой вершины.

Высота вершины – длина максимального пути от этой вершины до листа.

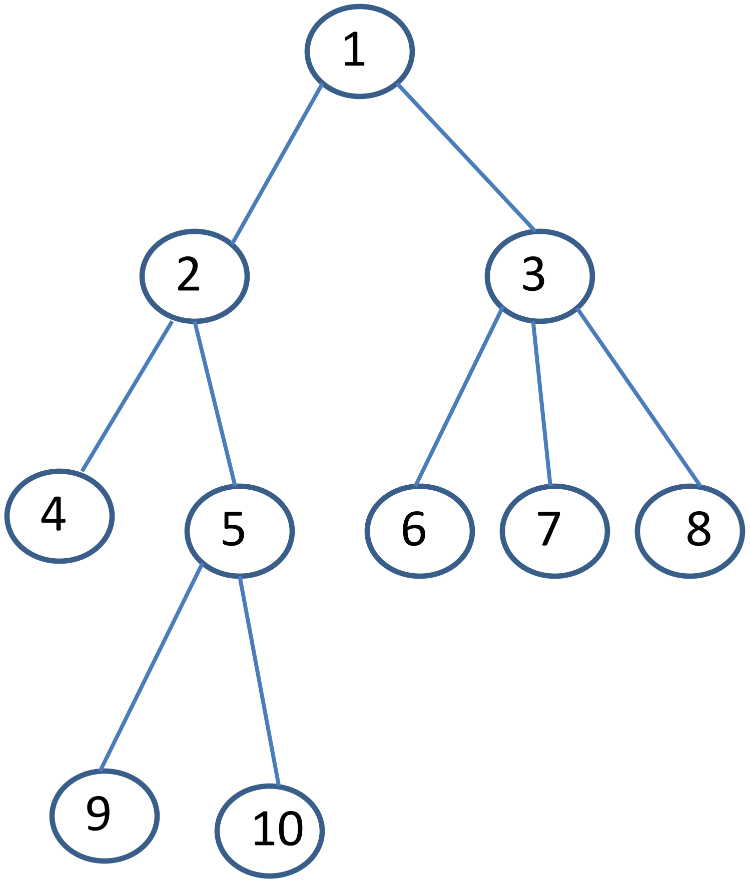
Высота дерева – длина максимального пути от корня до листа.

Глубина корня = 0.

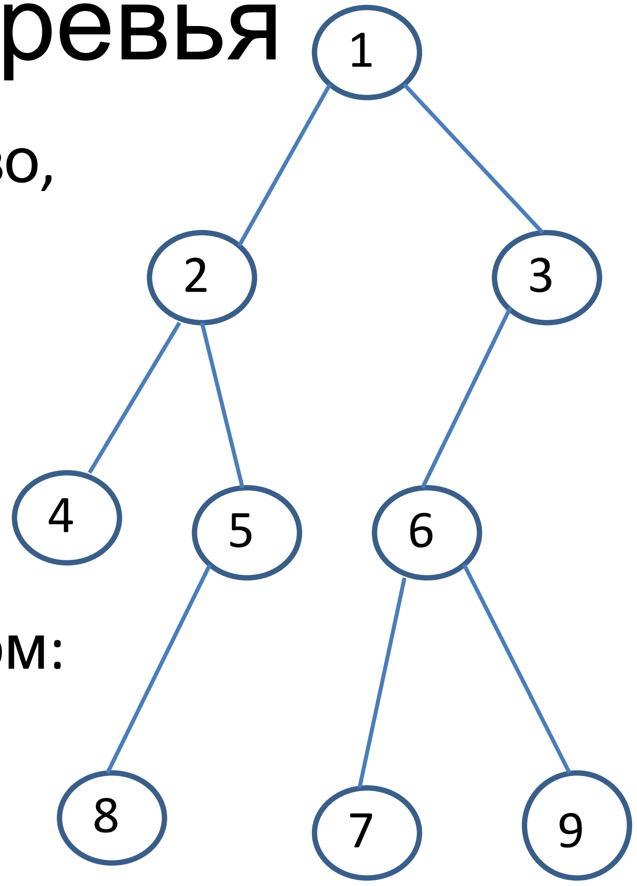
Например, глубина вершины 2 = 1, высота вершины 2 = 2.

Высота вершины 1 = высота дерева =3.

Глубина вершины 1 = 0



Бинарные деревья

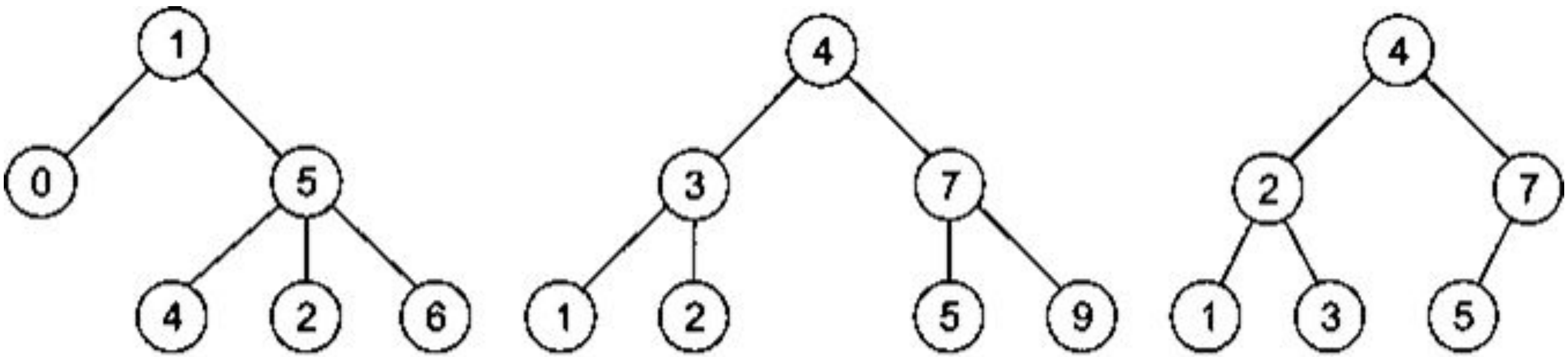


Упорядоченное дерево – это дерево, в котором множество сыновей каждой вершины упорядочено слева направо.

Бинарное дерево – это упорядоченное дерево, в котором:

1) любой сын – либо левый либо правый,
2) любой узел имеет не более одного левого и не более одного правого сына.

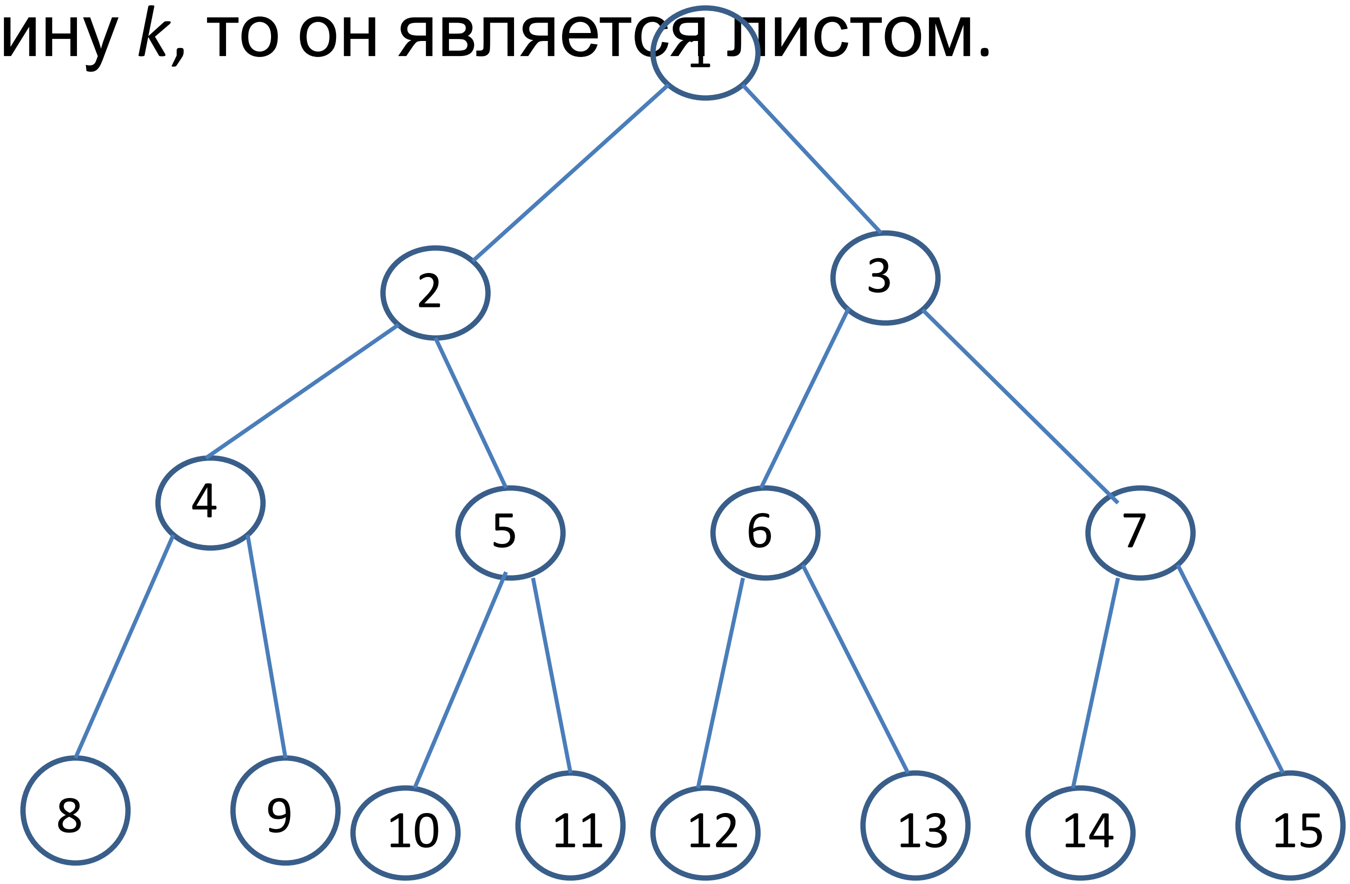
Бинарное дерево



а б в

- Дерево на рис. **а** не является бинарным, так как у узла 5 есть три дочерних узла.
- Дерево на рис. **б** является бинарным, но не является деревом поиска, так как узел 2 является правым дочерним узлом по отношению к узлу 3, нарушая свойство дерева поиска.
- Дерево на рис. **в** представляет собой корректное бинарное дерево поиска.

Бинарное дерево называется *полным*, если существует некоторое целое *k*, такое что любой узел глубины меньше *k* имеет как левого, так и правого сына, а если узел имеет глубину *k*, то он является листом.



Представление полных бинарных деревьев с помощью массива

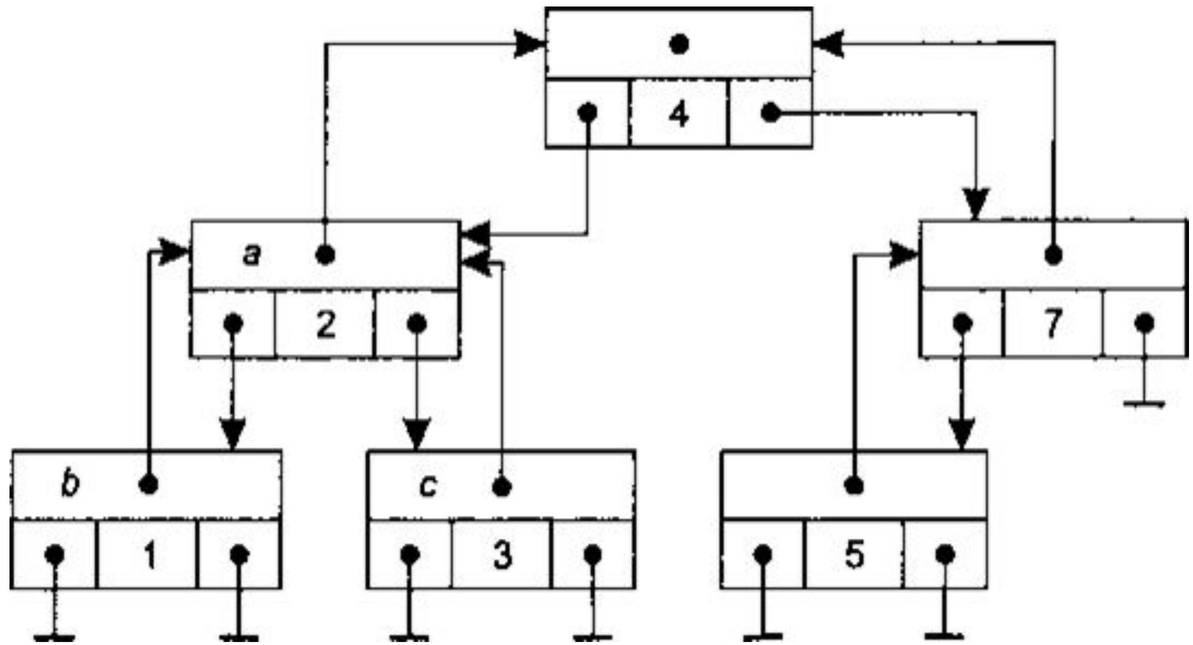
Пусть T[2^{k+1}-1] – массив для хранения вершин дерева, k- высота дерева.

В T[0] хранится корень дерева.

Левый сын узла *i* расположен в позиции 2 * *i* + 1, правый сын – в позиции 2 * *i* + 2.

Отец узла, находящегося в позиции *i* >0, расположен в позиции $\lfloor (i-1)/2 \rfloor$.

Бинарное дерево поиска



- изображено возможное представление бинарного дерева поиска с использованием полей указателей (например, двусвязный список).
- На этом рисунке, цифры представляют собой значения, хранящиеся в каждом из элементов дерева, а стрелки показывают, как при помощи указателей каждый узел дерева связан со своими правым и левым поддеревьями, а также с родительским узлом.

Задача №1

Таблица стоимости перевозок устроена следующим образом: числа, стоящие на пересечениях строк и столбцов таблиц, означают стоимость проезда между соответствующими соседними станциями.

Если пересечение строки и столбца ***пусто***, то станции ***не являются соседними***. Укажите таблицу, для которой выполняется условие: «Минимальная стоимость проезда из А в В не больше 6».

Стоимость проезда по маршруту складывается из стоимостей проезда между соответствующими соседними станциями.

1)

	A	B	C	D	E
A			3	1	
B			4		2
C	3	4			2
D	1				
E		2	2		

2)

	A	B	C	D	E
A			3	1	1
B			4		
C	3	4			2
D	1				
E	1		2		

3)

	A	B	C	D	E
A			3	1	4
B			4		2
C	3	4			2
D	1				
E	4	2	2		

4)

	A	B	C	D	E
A				1	
B			4		1
C		4		4	2
D	1		4		
E		1	2		

Решение задачи №1

1. Для каждой таблицы нарисуем соответствующий ей взвешенный граф

1)

	A	B	C	D	E
A			3	1	
B			4		2
C	3	4			2
D	1				
E		2	2		

2)

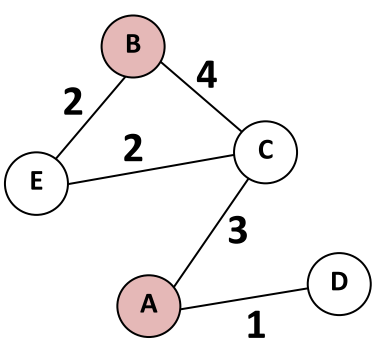
	A	B	C	D	E
A			3	1	1
B			4		
C	3	4			2
D	1				
E	1		2		

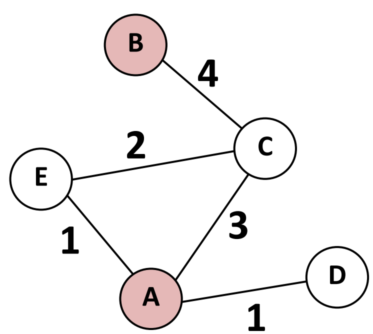
3)

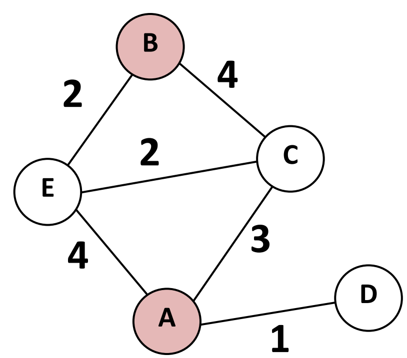
	A	B	C	D	E
A			3	1	4
B			4		2
C	3	4			2
D	1				
E	4	2	2		

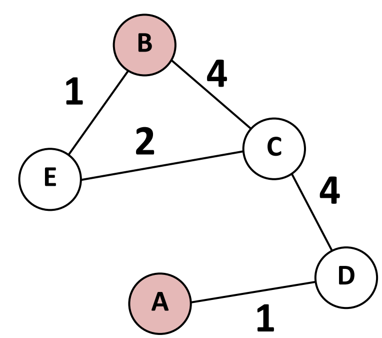
4)

	A	B	C	D	E
A				1	
B			4		1
C		4		4	2
D	1		4		
E		1	2		









Решение задачи №1

2. Теперь по схемам определяем кратчайшие маршруты для каждой таблицы:

1: $A \xrightarrow{3} C \xrightarrow{4} B$ или $A \xrightarrow{3} C \xrightarrow{2} E \xrightarrow{2} B$, стоимость 7

2: $A \xrightarrow{3} C \xrightarrow{4} B$ или $A \xrightarrow{1} E \xrightarrow{2} C \xrightarrow{4} B$, стоимость 7

3: $A \xrightarrow{4} E \xrightarrow{2} B$, стоимость 6

4: $A \xrightarrow{1} D \xrightarrow{4} C \xrightarrow{2} E \xrightarrow{1} B$, стоимость 8

3. Условие *«не больше 6»* выполняется только для таблицы 3

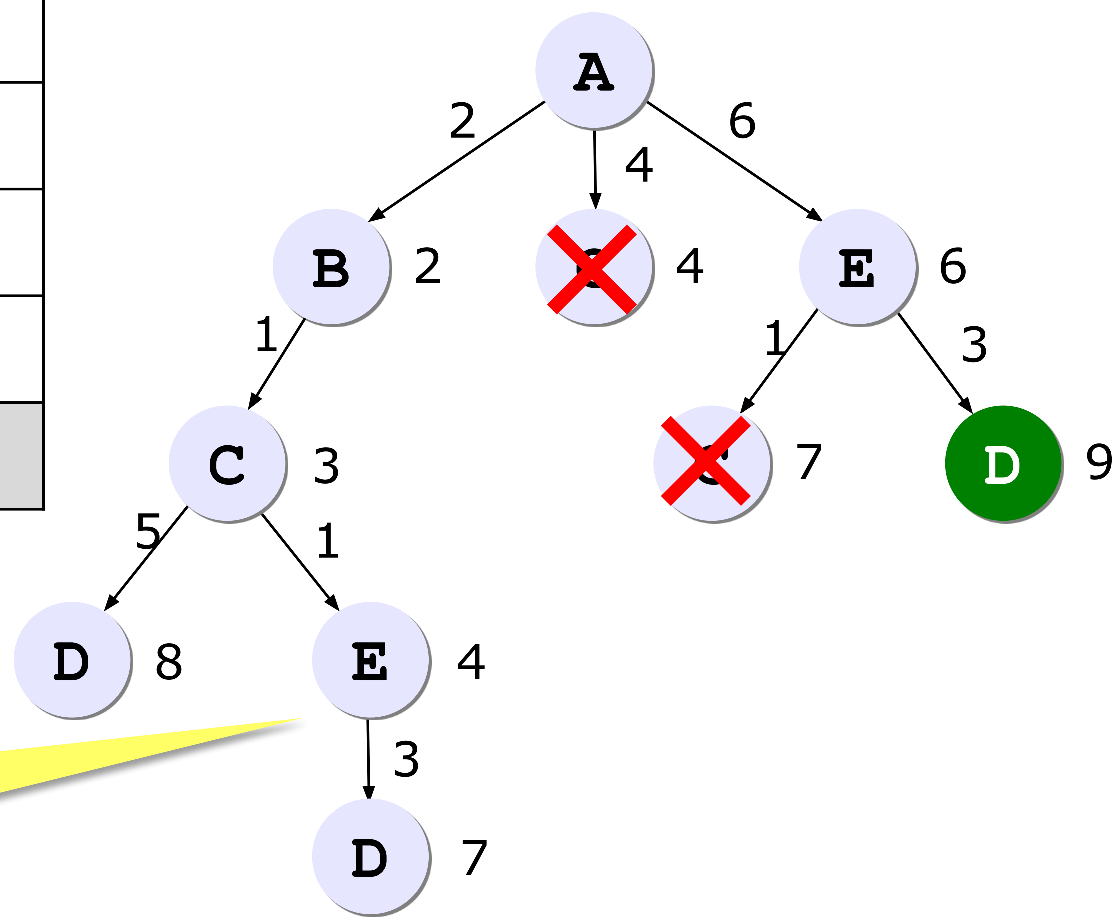
Таким образом, правильный ответ – 3.

Задача №2

Кратчайший путь (перебор)

Определите кратчайший путь между пунктами A и D.

	A	B	C	D	E
A		2	4		6
B	2		1		
C	4	1		5	1
D			5		3
E	6		1	3	



дерево возможных путей