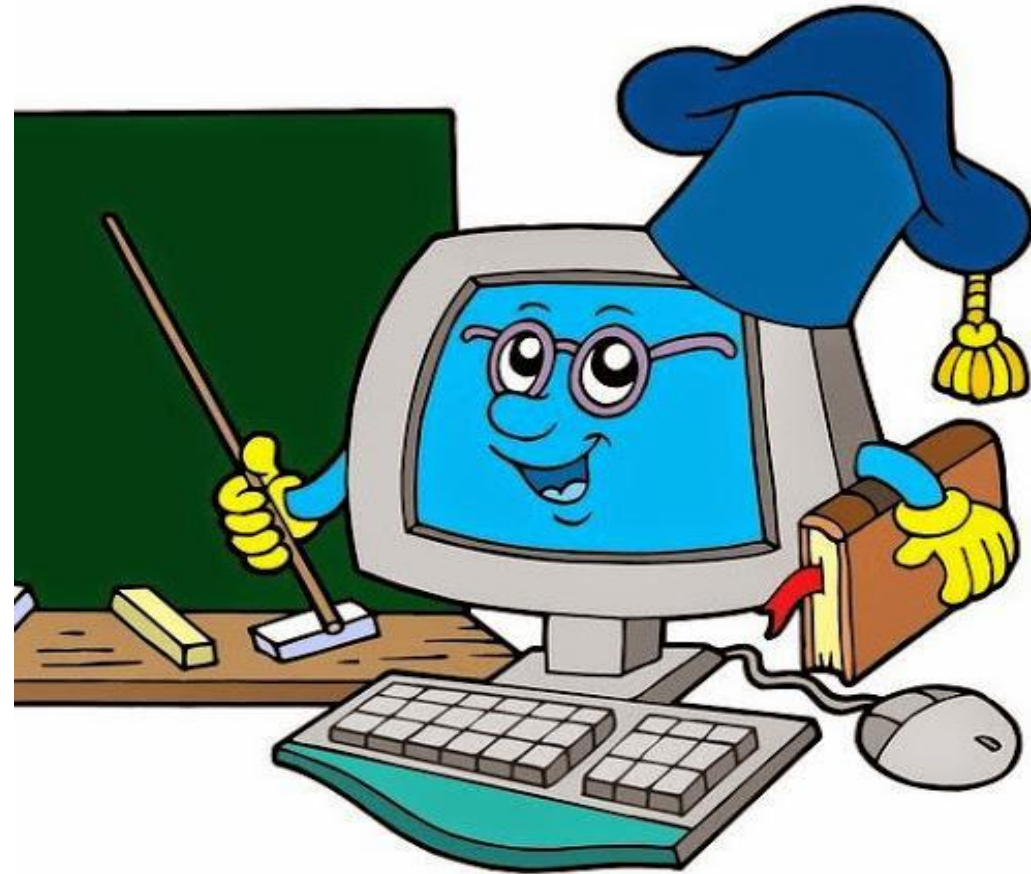


Решение алгоритмических задач связанных с анализом графов. Использование графов деревьев, списков, при описании объектов и процессов окружающего мира

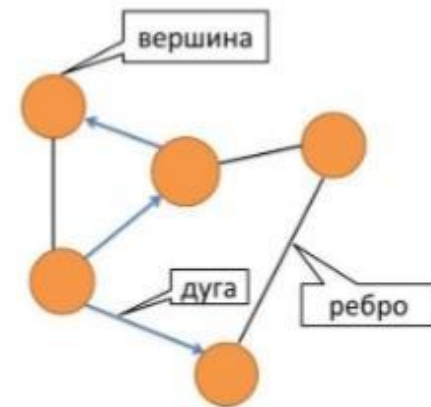


Немного истории...

1736 год, г.Кёнигсберг. Через город протекает река Прегеля. В городе - семь мостов, расположенных так, как показано на рисунке выше. С давних времен жители Кенигсберга бились над загадкой: можно ли пройти по всем мостам, пройдя по каждому только один раз? Эту задачу решали и теоретически, на бумаге, и на практике, на прогулках - проходя по этим самым мостам. Никому не удавалось доказать, что это неосуществимо, но и совершить такую «загадочную» прогулку по мостам никто не мог.

Разрешить проблему удалось знаменитому математику Леонарду Эйлеру. Причем, он решил не только эту конкретную задачу, но придумал общий метод решения подобных задач. При решении задачи о Кенигсбергских мостах Эйлер поступил следующим образом: он "сжал" сушу в точки, а мосты "вытянул" в линии. Такую фигуру, состоящую из точек и линий, связывающих эти точки, называют *ГРАФОМ*.

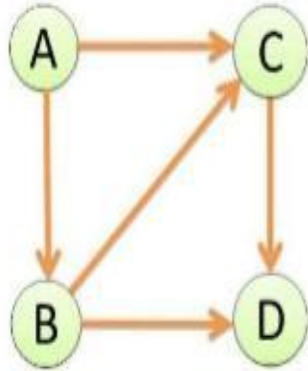
Граф – это совокупность непустого множества вершин и связей между вершинами. Кружки называются вершинами графа, линии со стрелками – дугами, без стрелок – ребрами.



Виды графов:

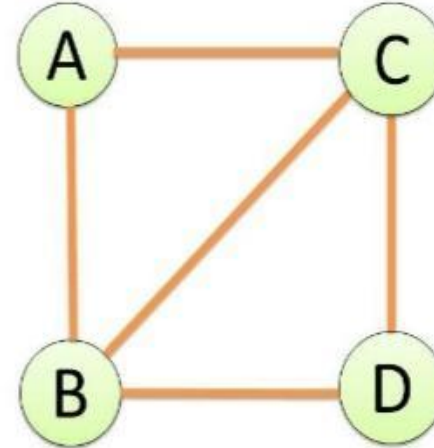
1. **Ориентированный граф** (кратко **орграф**) — рёбрам которого присвоено направление.
2. **Неориентированный граф** - это **граф**, в котором нет направления линий.
3. **Взвешенный граф** – дуги или ребра имеют вес (дополнительная информация).

Ориентированный граф (орграф)

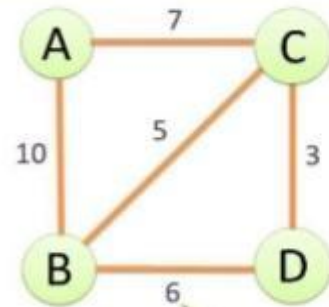


	A	B	C	D
A	0	1	1	0
B	0	0	1	1
C	0	0	1	1
D	0	0	0	0

Неориентированный граф



Взвешенный граф



	A	B	C	D
A		10	7	0
B	10		5	6
C	7	5		3
D	0	6	3	

Вес ребра

Граф - дерево

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



Классификация компьютеров

Граф иерархической структуры - **Дерево**

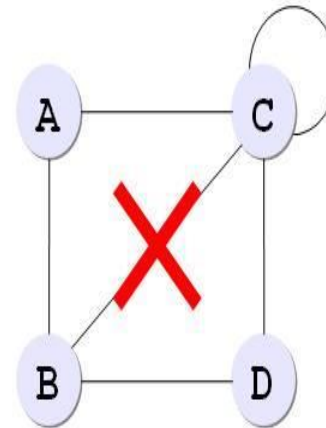


Между любыми двумя его вершинами существует единственный путь.

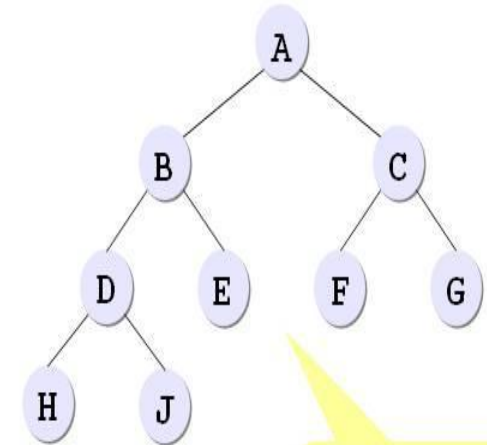
Деревья не содержат циклов и петель

Дерево – это граф?

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



ABC ABDC
BCD CCC...

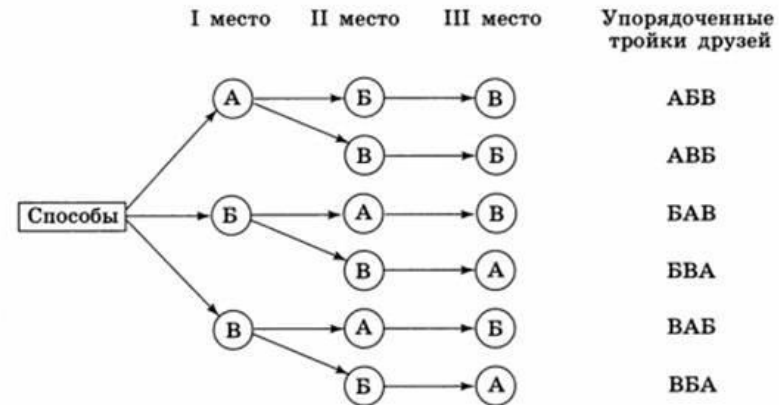


Дерево

Пример решения задачи с помощью граф - дерево

- **Граф-дерево.**

Пример 4. Антон, Борис и Василий купили 3 билета на футбольный матч на 1,2,3-е места первого ряда. Сколькими способами они могут занять имеющиеся три места?

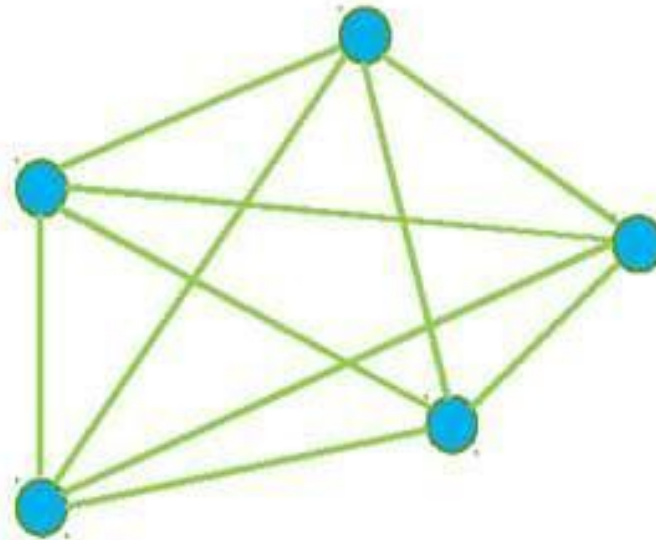


В итоге получаем 6 способов.

Ответ: 6.

ПРИМЕНЕНИЕ ТЕОРИИ ГРАФОВ

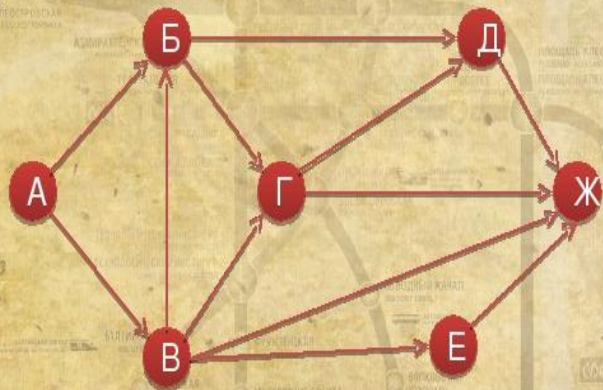
ПРИ РЕШЕНИИ ЗАДАЧ



Решение задач на графах

Задача 2

На рисунке - схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город Ж?



1. А-Б-Д-Ж
2. А-Б-Г-Д-Ж
3. А-Б-Г-Ж
4. А-В-Б-Д-Ж
5. А-В-Б-Г-Д-Ж
6. А-В-Б-Г-Ж
7. А-В-Г-Д-Ж
8. А-В-Г-Ж
9. А-В-Ж
10. А-В-Е-Ж

Ответ: 10 путей

В своей работе я рассматривала и решала задачи с помощью графов.

В первенстве класса по настольному теннису 6 участников: Андрей, Борис, Виктор, Галина, Дмитрий и Елена. Первенство проводят по круговой системе – каждый из участников играет с каждым из остальных один раз. К настоящему моменту некоторые игры уже проведены: Андрей сыграл с Борисом, Галиной, Еленой; Борис – с Андреем, Галиной; Виктор – с Галиной, Дмитрием, Еленой; Галина – с Андреем, Виктором и Борисом. Сколько игр проведено к настоящему моменту и сколько еще осталось?

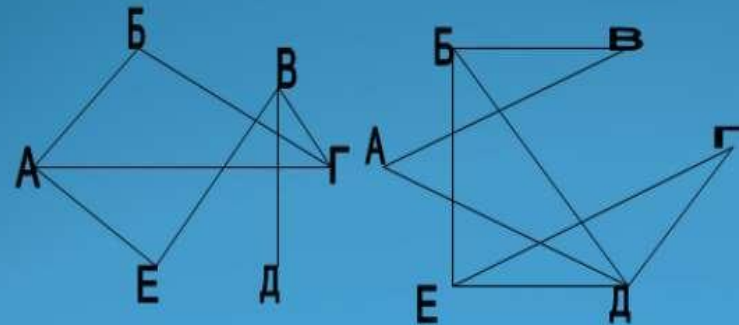


Рис.1

Рис.2

Решение: Построим граф (рис.1).

Сыграно 7 игр.

На рис. 2 граф имеет 8 ребер, следовательно, осталось провести 8 игр.

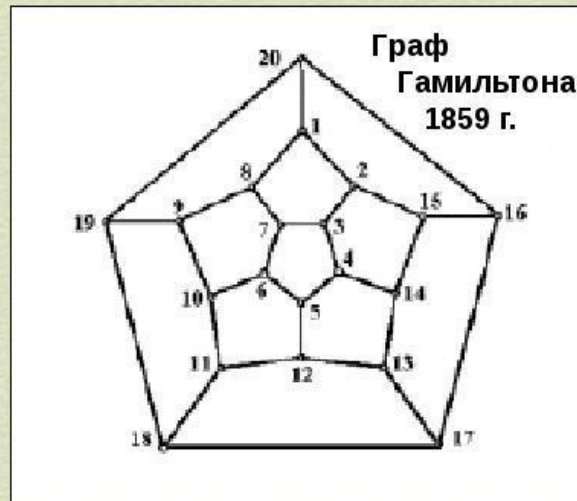
Граф Гамильтон

Теория графов

Задача: пройти через все вершины по одному разу (номера вершин указывают последовательность шага прохождения каждого ребра не обязательно).



**Уильям Роуэн Гамильтон
1805-1865
Ирландский математик и физик**



Алгоритмы.

Алгоритм – это предназначенное для конкретного исполнителя точное описание последовательности действий, направленных на решение поставленной задачи.

Алгоритм

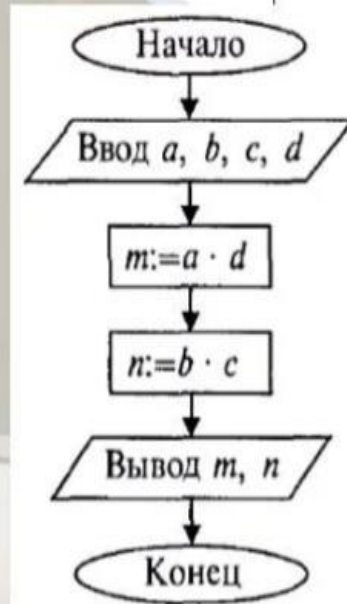
- **Алгоритм** – это предназначенное для конкретного исполнителя точное описание последовательности и действий, направленных на решение поставленной задачи.



Виды алгоритмов

Линейные алгоритмы

- Для описания алгоритмов чаще всего используют язык блок-схем.
- Эллипсами обозначают начало и конец программы.
- Параллелограммы – операции ввода и вывода данных.
- Прямоугольники – операции присваивания.
- Ромбы – ветвления алгоритма.
- На рисунке показана блок-схема алгоритма деления дробей.
- Вначале задаются входные параметры: a , b , c , и d .
- Последовательно вычисляются числитель и знаменатель искомой дроби.
- Выводятся числитель m и знаменатель n искомой дроби.



Ветвление - алгоритмическая конструкция, в которой в зависимости от результата проверки условия (да или нет) предусмотрен выбор одной из двух последовательностей действий (ветвей).

Алгоритмы, в основе которых лежит структура «ветвление», называют **разветвляющимися**.

Полная форма ветвления

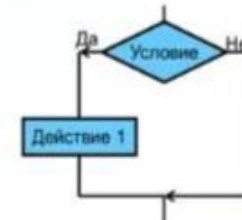
если <условие>
то <действие 1>
иначе <действие 2>
все



if (условие) then
действие_1
else
действие_2;

Неполная форма ветвления

если <условие>
то <действие 1>
все



if (условие) then
действие_1;

Операции сравнения

- $A < B$ A меньше B
- $A \leq B$ A меньше или равно B
- $A = B$ A равно B
- $A > B$ A больше B
- $A \geq B$ A больше или равно B
- $A \neq B$ A не равно B

Если в условном операторе после **then** или после **else** нужно выполнить несколько операторов, то используют операторные скобки – конструкцию вида:

```
begin  
  <последовательность операторов>  
end;
```


Решение задач с помощью линейных и разветвляющихся алгоритмов

Задача на линейный алгоритм

Вводятся объём и масса тела. Определить плотность материала этого тела.

Входные данные:

V - объём тела (real);
m - масса тела (real);

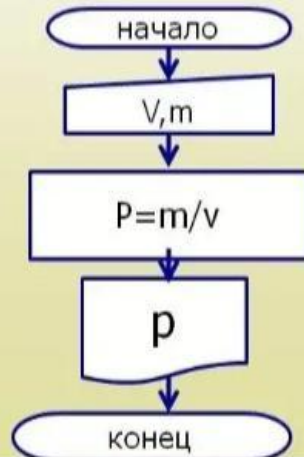
Выходные данные:

p - плотность тела (real).

Паскаль-программа

```
Program z1;  
uses crt;  
Var V,m,p:real;  
BEGIN  
Clrscr;  
Write('Введите значения объёма и  
массы тела:');  
ReadLn(v,m);  
P:=m/V;  
Write('Плотность тела  
объёмом',V:7:1,' и массой',m:7:1,  
равна',p:8:2);  
ReadLn;  
END.
```

БЛОК-СХЕМА



Пример разветвляющегося алгоритма «Решение квадратного уравнения»

На алгоритмическом языке

В виде блок-схемы:

начало

ввод a, b, c

если a =

то «Уравнение не является квадратным»

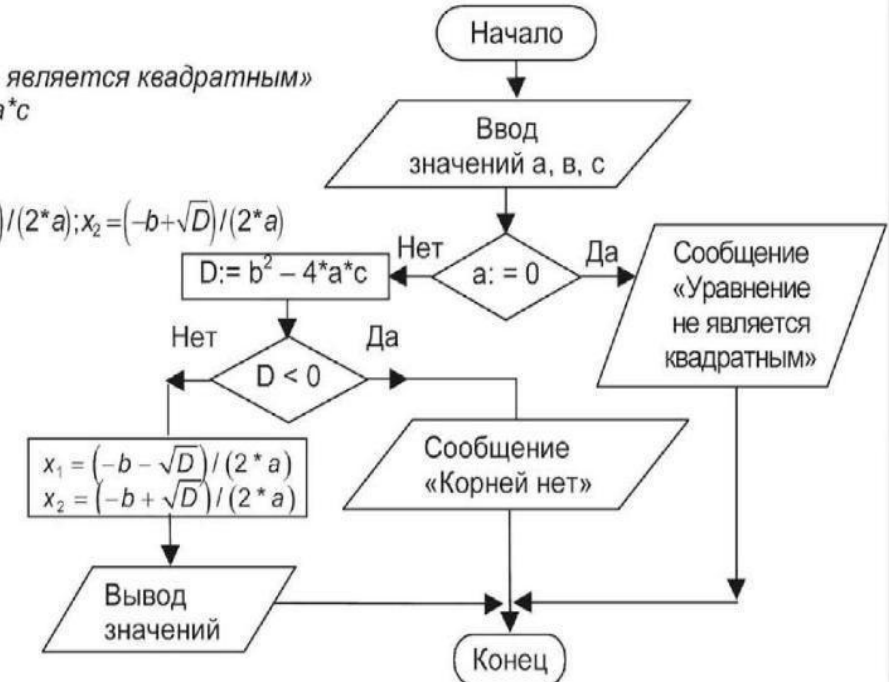
иначе $D := b^2 - 4*a*c$

если $D < 0$

то «Корней нет»

иначе $x_1 = (-b - \sqrt{D}) / (2*a); x_2 = (-b + \sqrt{D}) / (2*a)$

конец



Пояснения к алгоритму: После ввода значений коэффициентов производится проверка значения a. При $a = 0$ уравнение становится линейным, – выполнение алгоритма прекращается. В противном случае (когда a отлично от 0) производится вычисление дискриминанта, а затем – его проверка. Если $D < 0$, то уравнение не имеет корней, выполнение алгоритма прекращается. В противном случае ($D > 0$) производится вычисление корней x_1 и x_2 . Пользователю предъявляется полученный результат и выполнение алгоритма на этом заканчивается.

*СПАСИБО ЗА
ВНИМАНИЕ!*

