

Лицей – интернат естественных наук



# *Элементы теории графов. Способы обходов графов.*

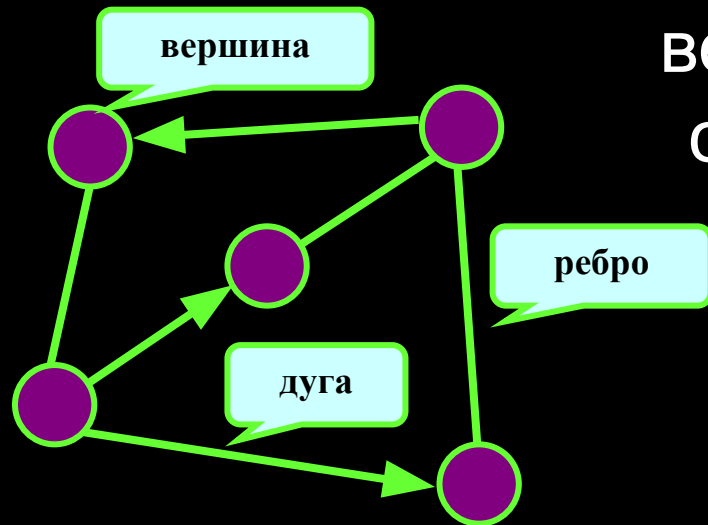


**Ищенко М. Н.**

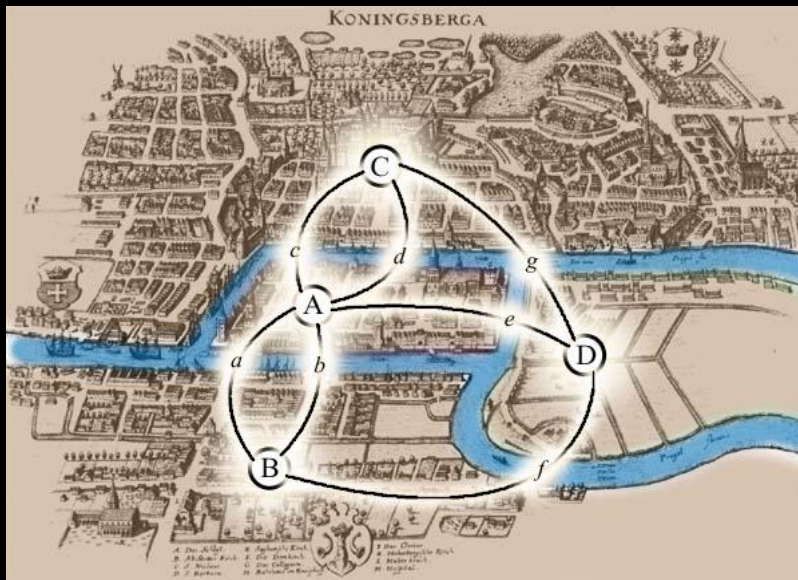
*В основе теории лежит понятие графа.*

# Граф

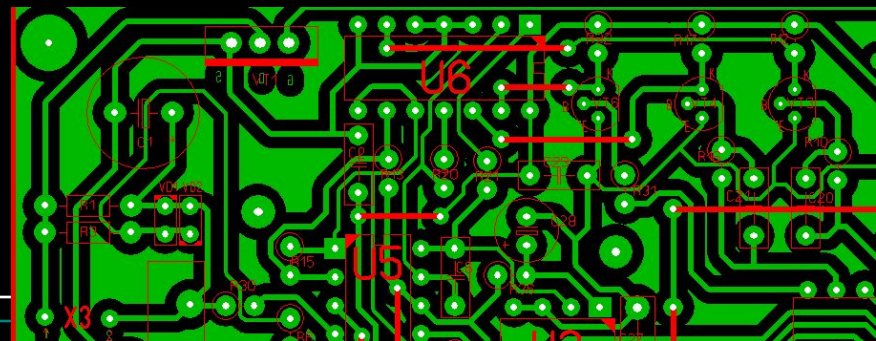
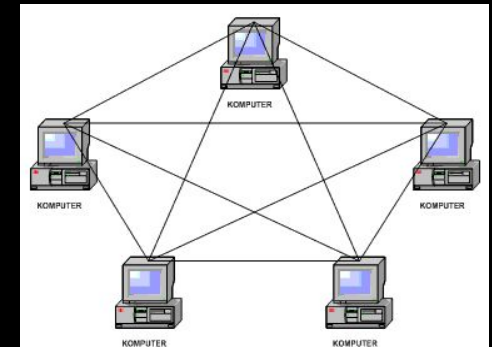
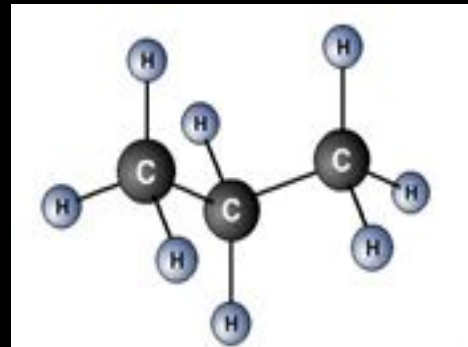
- совокупность конечного числа точек, называемых вершинами графа, и попарно соединяющих некоторые из этих вершин линий, называемых ребрами или дугами графа.



Первая работа по теории графов, принадлежащая известному швейцарскому математику Л. Эйлеру, появилась в 1736 г., связанная с решением известной головоломки о мостах Кёнигсберга. Толчок к развитию теории графов получила на рубеже XIX и XX столетий, когда резко возросло число работ в области топологии и комбинаторики.



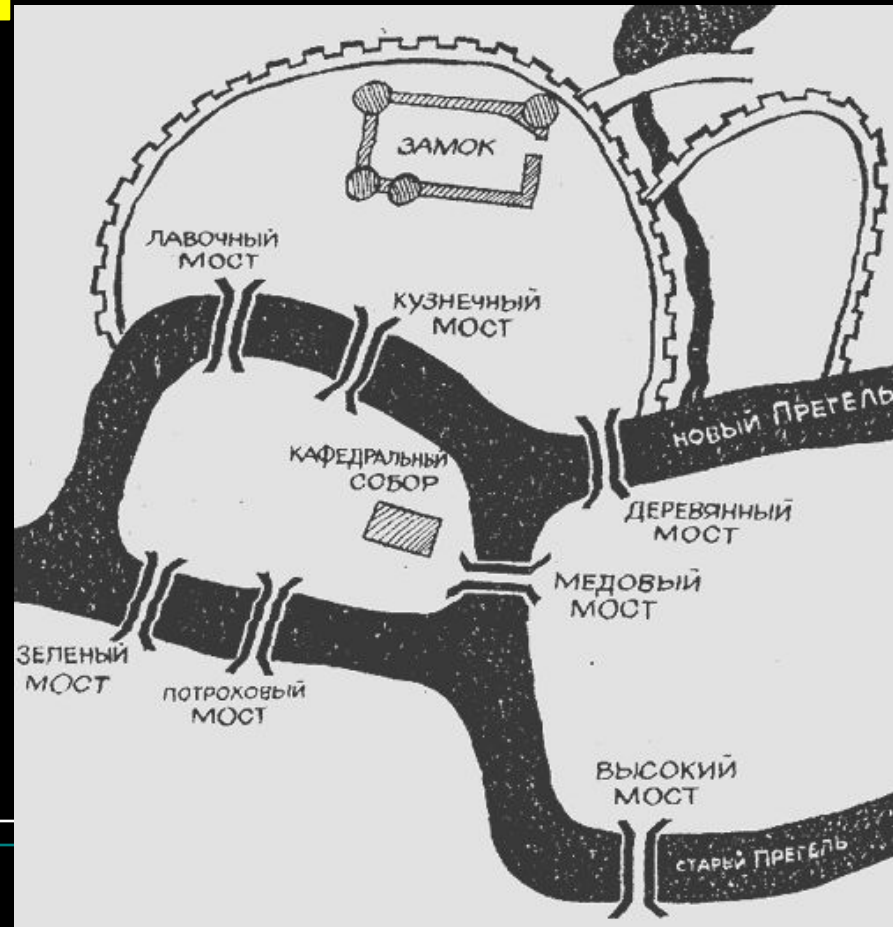
В настоящее время графы эффективно используются в теории планирования и управления, теории расписаний, социологии, экономике, биологии, медицине, географии. Широкое применение находят графы в таких областях, как программирование, электроника, в решении вероятностных и комбинаторных задач, нахождения кратчайшего расстояния и др. Математические головоломки тоже являются частью теории графов.



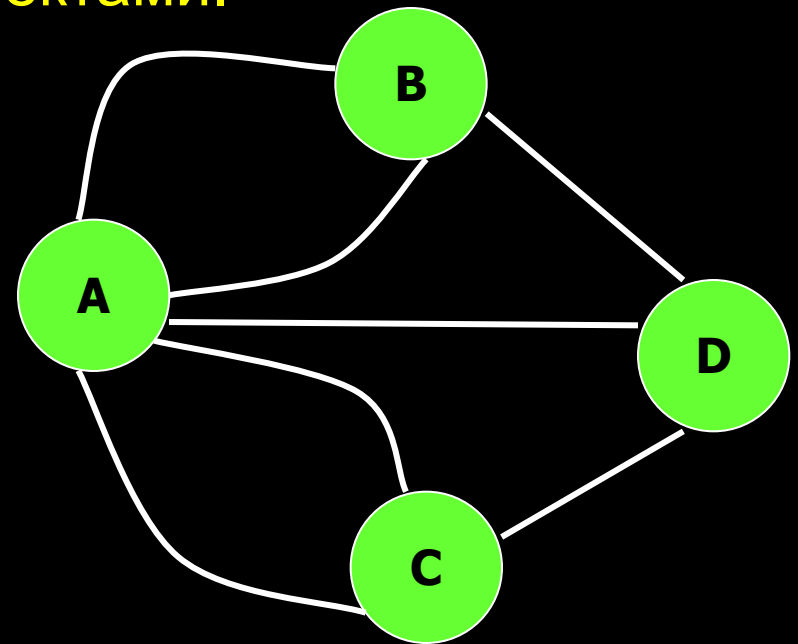
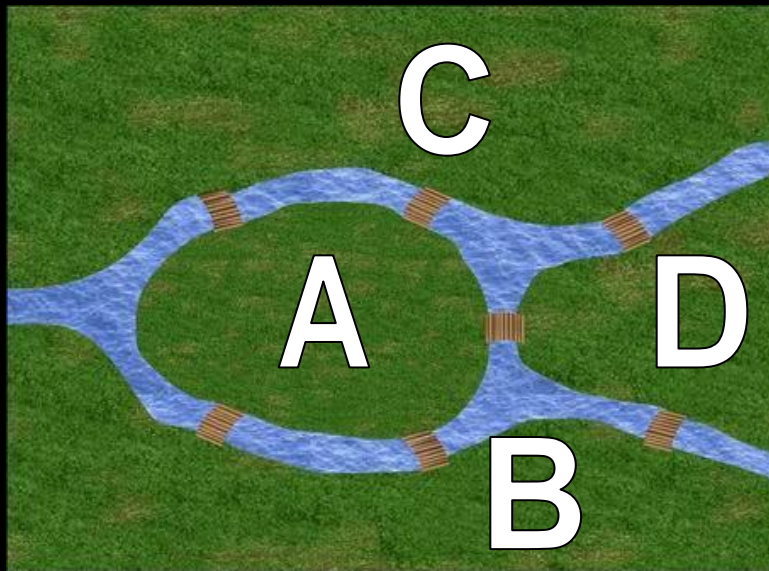
Благодаря использованию графов можно упростить решение задач.

# Кёненсбергские мосты

«В Кенигсберге есть остров, называемый Кнейпгоф. Река, омывающая его, делится на два рукава, через которые перекинута семь мостов. Можно ли обойти все эти мосты, не побывав ни на одном из них более раза?»



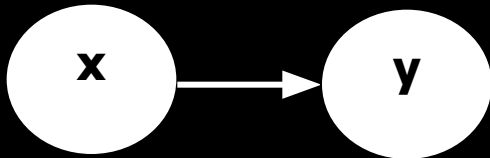
На практике вершины графа можно использовать для представления объектов, а дуги — для отношений между объектами.



Л. Эйлеру удалось ответить на этот вопрос. Представим, что мосты, соединяющие части города – это рёбра графа, а части города – это вершины графа.

# Основные понятия

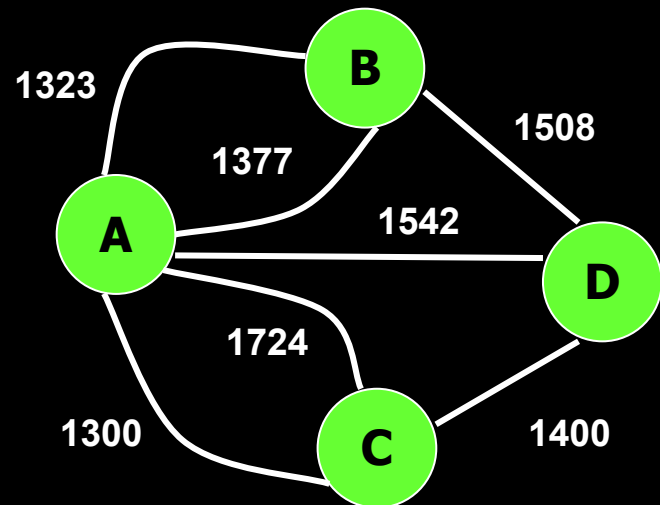
Ориентированный  
граф



Неориентированный  
граф

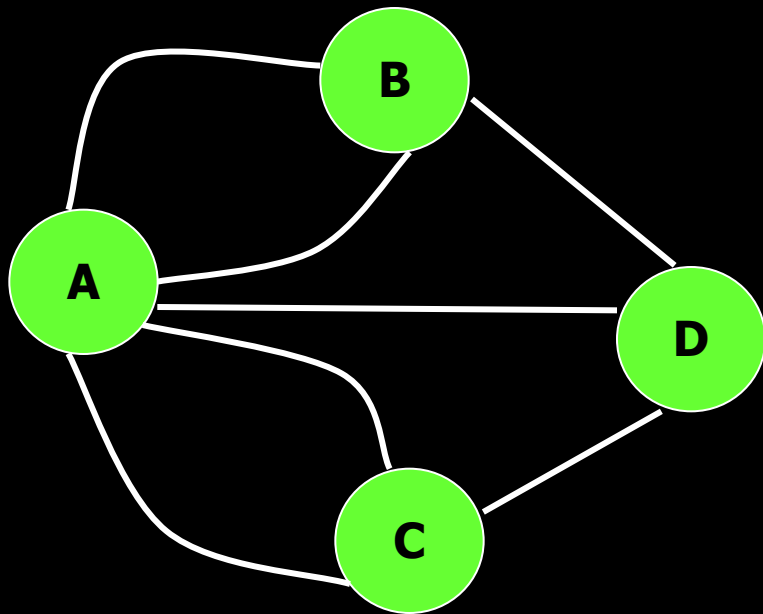


Взвешенный  
граф

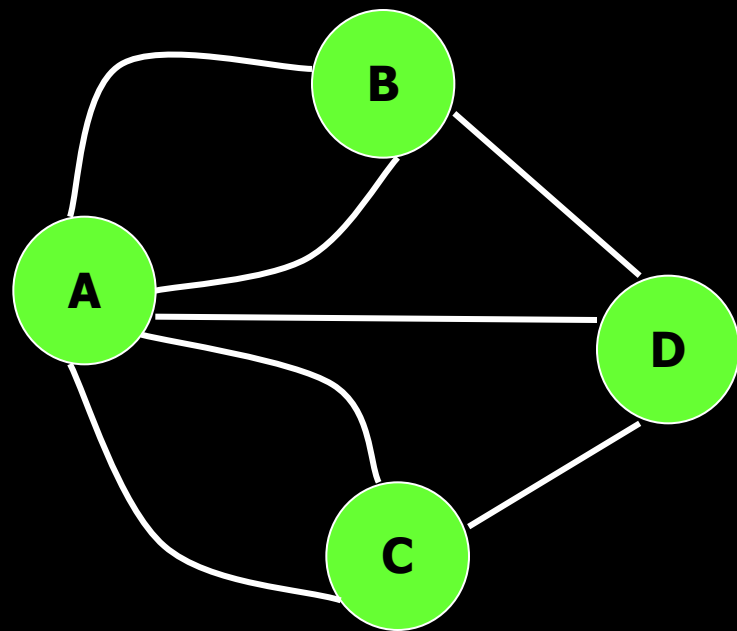


# Основные понятия

Смежные  
вершины



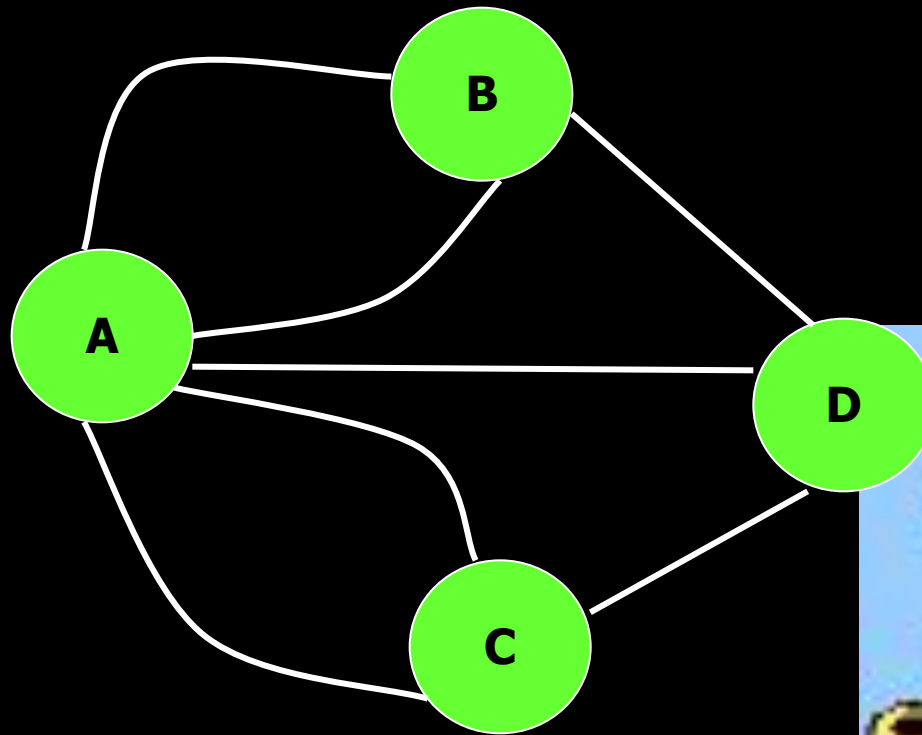
Смежные  
рёбра





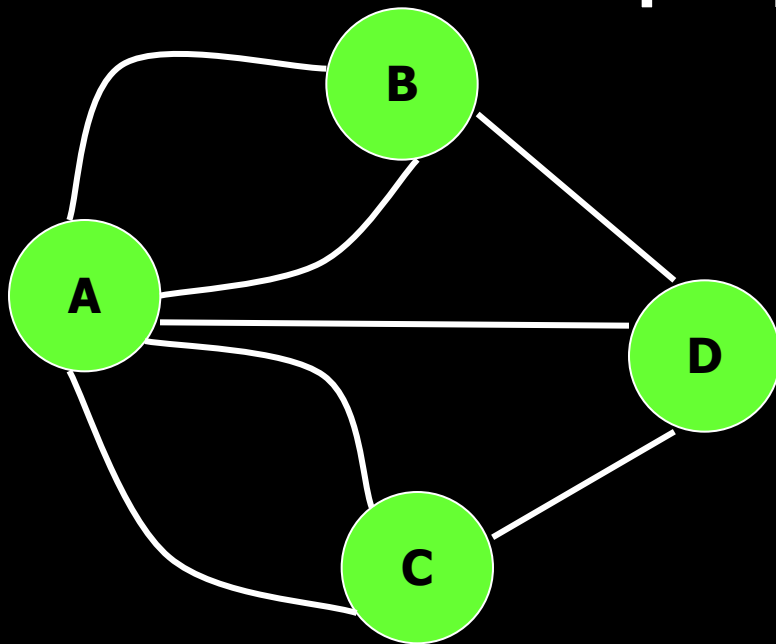
# Основные понятия

## Инциденция



# Основные понятия

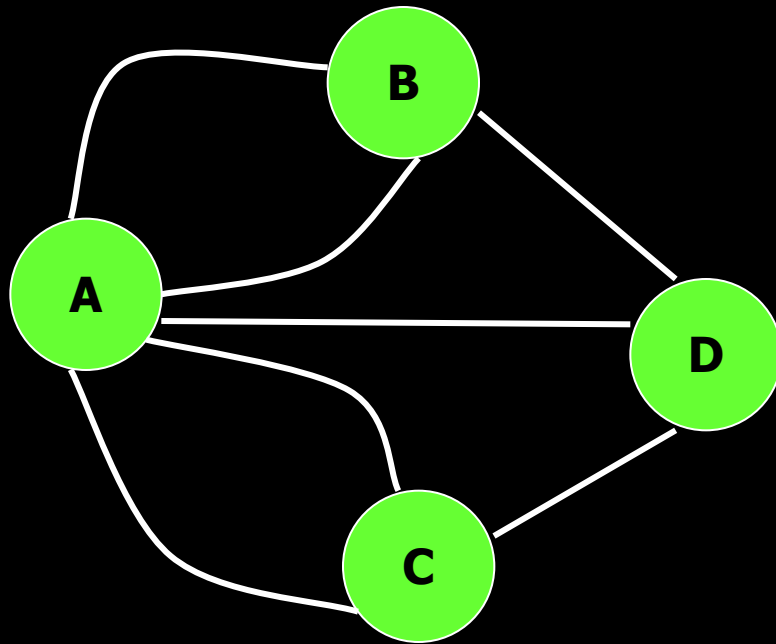
Степень вершины  
в неориентированном  
графе



Степень вершины A  
равна

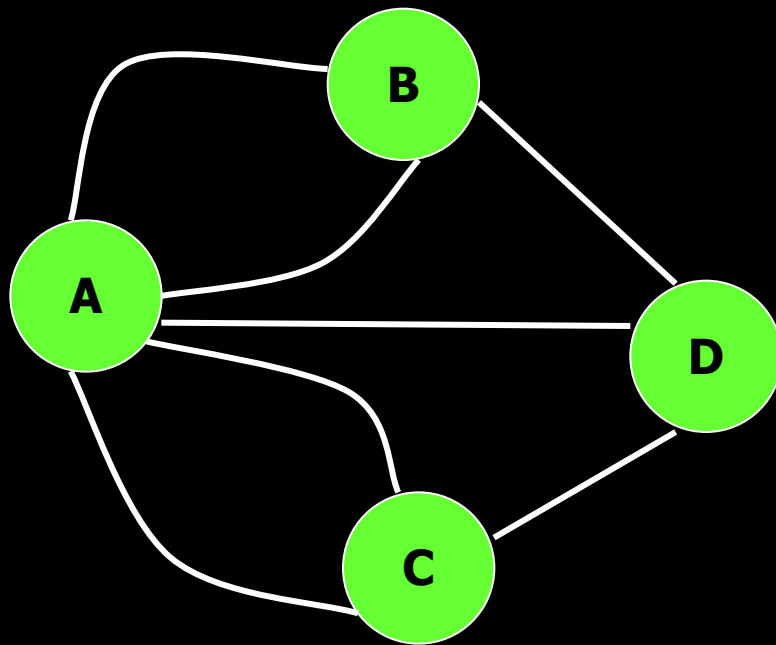
**5**

# Кёненсбергские мосты



Задача сводится к тому, чтобы начертить граф одним росчерком, не отрывая карандаша от бумаги и не проводя ни одной линии дважды. Но это сделать невозможно, т.к. граф кёнигсбергских мостов имеет четыре нечётные вершины, следовательно, невозможно пройти по всем мостам, не проходя ни по одному из них дважды.

# Пути (маршруты) в графах



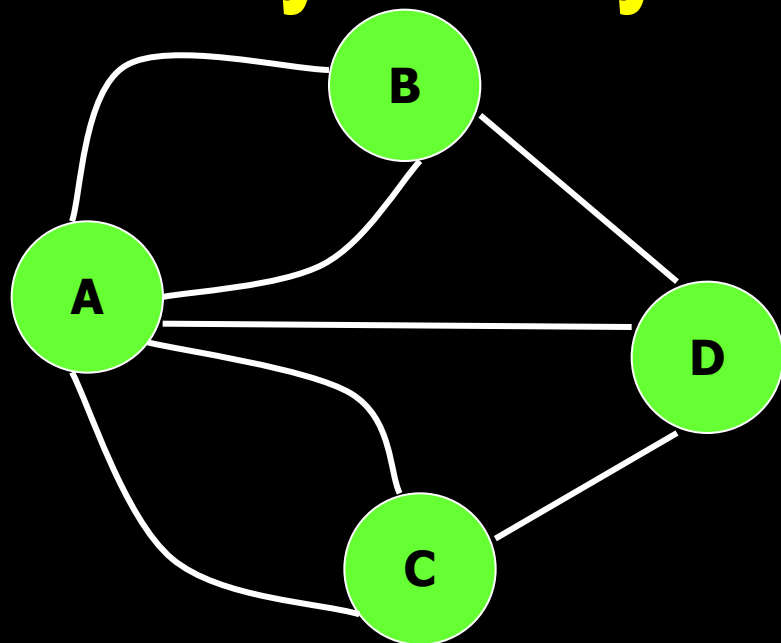
Путь из A в D:

**A B D**

Длина пути из A в D:

**2**

# Замкнутый путь



А В А D C А

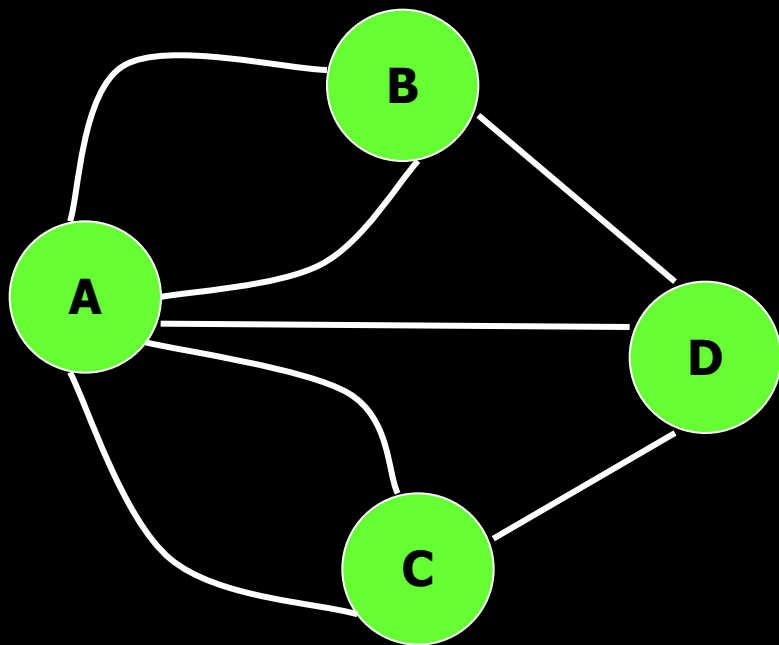
# Цикл

А В D C А



# Способы представления графов

## Матрица смежности

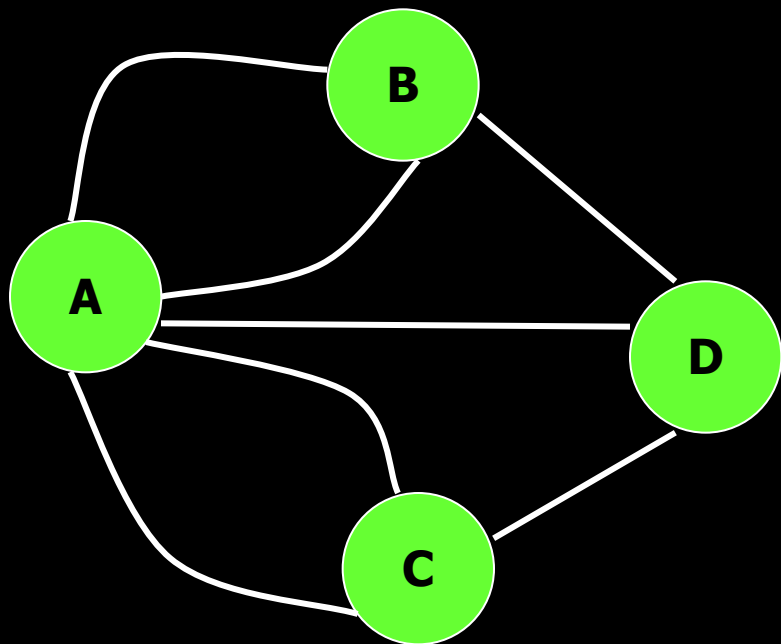


|   | A | B | C | D |
|---|---|---|---|---|
| A | 0 | 1 | 1 | 1 |
| B | 1 | 0 | 0 | 1 |
| C | 1 | 0 | 0 | 1 |
| D | 1 | 1 | 1 | 0 |

# Способы представления графов

## графов

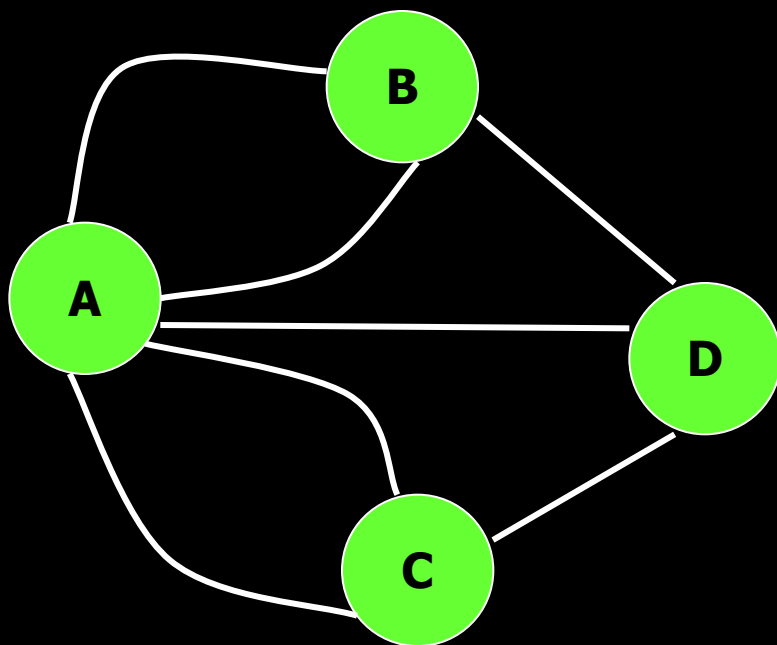
### Матрица инциденций



|   | A-B | A-C | A-D | B-D | C-D |
|---|-----|-----|-----|-----|-----|
| A | 1   | 1   | 1   | 0   | 0   |
| B | 1   | 0   | 0   | 1   | 0   |
| C | 0   | 1   | 0   | 0   | 1   |
| D | 0   | 0   | 1   | 1   | 1   |

# Способы представления графов

## Список смежности

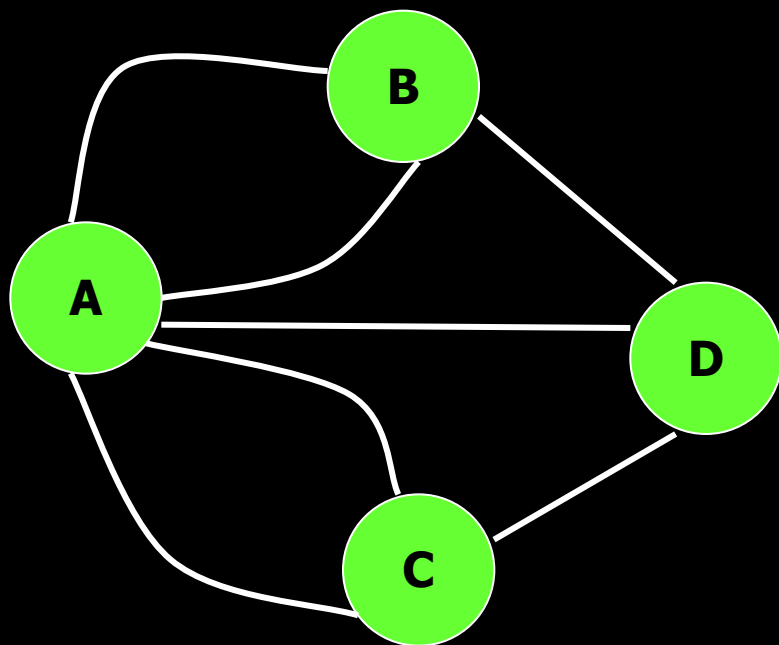


|    |         |
|----|---------|
| A: | B, D, C |
| B: | A, D    |
| C: | A, D    |
| D: | A, B, C |



# Обходы графов

## Поиск в глубину



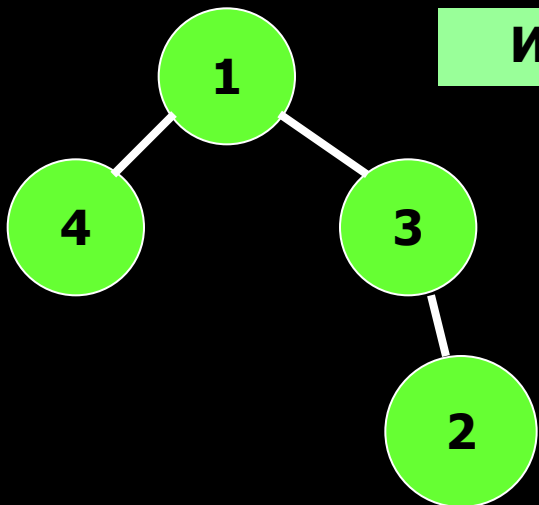
**A B D C**



```

Program graf;
Var n,v,u: integer;
    gr: array [1..30, 1..30] of integer;
    nov: array [1..15] of boolean;
procedure dfs (v: integer);
    var u: integer;
Begin
  Readln;
  Write (v, ' ');
  nov [v]:=false;
  For u:=1 to n do
  If (gr[v,u]=1) and (nov[u]) then dfs (u);
  End;
Begin
  n:=4;
  for v:=1 to n do
    begin
nov [v]:= true;
  Writeln;
  For u:=1 to n do begin nov [u]:=true;
  Write ( ' gr [ ' ,v,u, ' ]=' );
  Read (gr [v,u]);

```



**Исходный массив**

|   |   |   |   |   |
|---|---|---|---|---|
|   | 1 | 2 | 3 | 4 |
| 1 | 0 | 0 | 1 | 1 |
| 2 | 0 | 0 | 1 | 0 |
| 3 | 1 | 1 | 0 | 0 |
| 4 | 1 | 0 | 0 | 0 |

**Размерность массива**  
n = 4

**Результат**

1 3 2 4

```

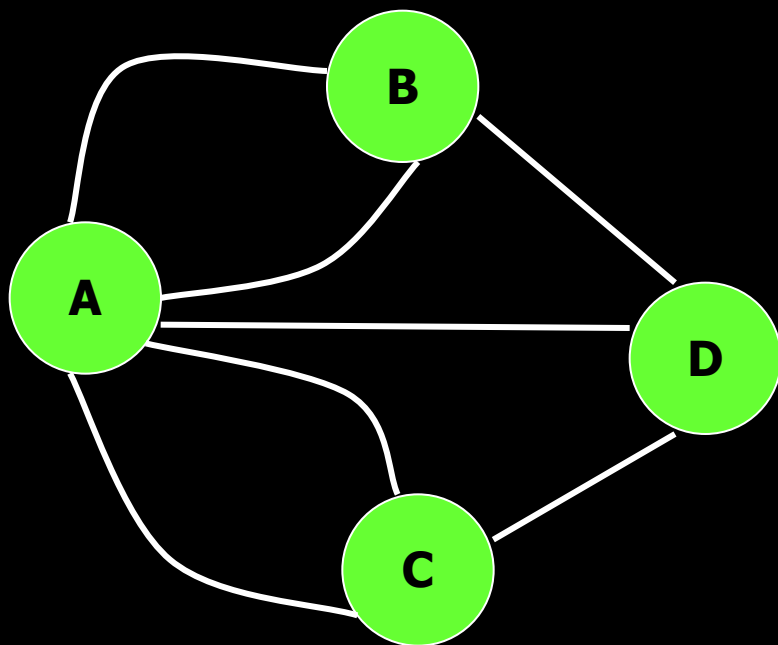
End;
End;
For v:=1 to n do begin
  IF nov [v] then
  dfs (v);
  End;
  Readln;
End.

```



# Обходы графов

## Поиск в ширину



A B C D



*Спасибо за внимание!*

