

Алгоритмы и структуры данных на Python. Интерактивный  
курс

## Урок 8



# Графы

Алгоритм Дейкстры. Поиск  
кратчайшего пути в ширину.

# План

- Задача о семи мостах
- Что такое граф?
- Классификация графов
- Сферы применения



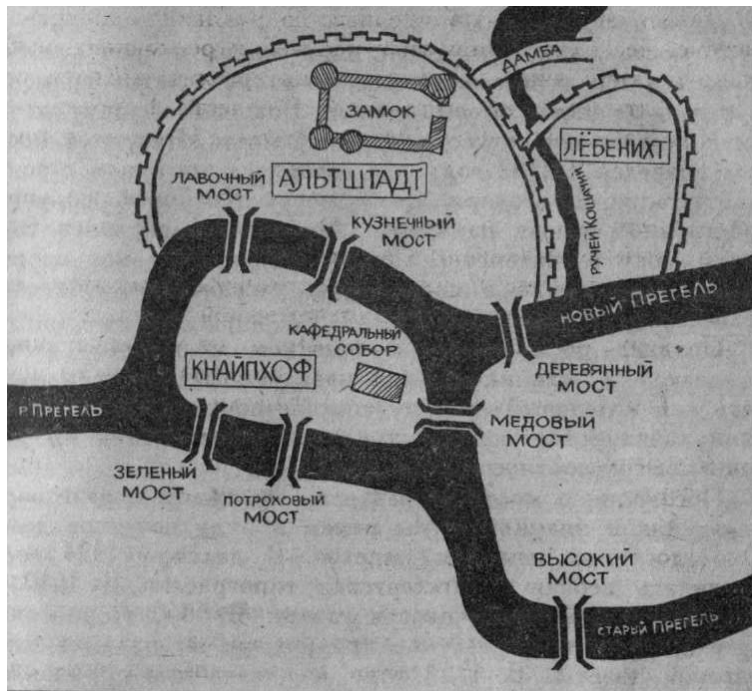
# Леонард Эйлер



В 1736 году Леонард Эйлер написал статью «Решение вопроса, связанного с геометрией положения», которая положила начало теории графов как математической дисциплине.



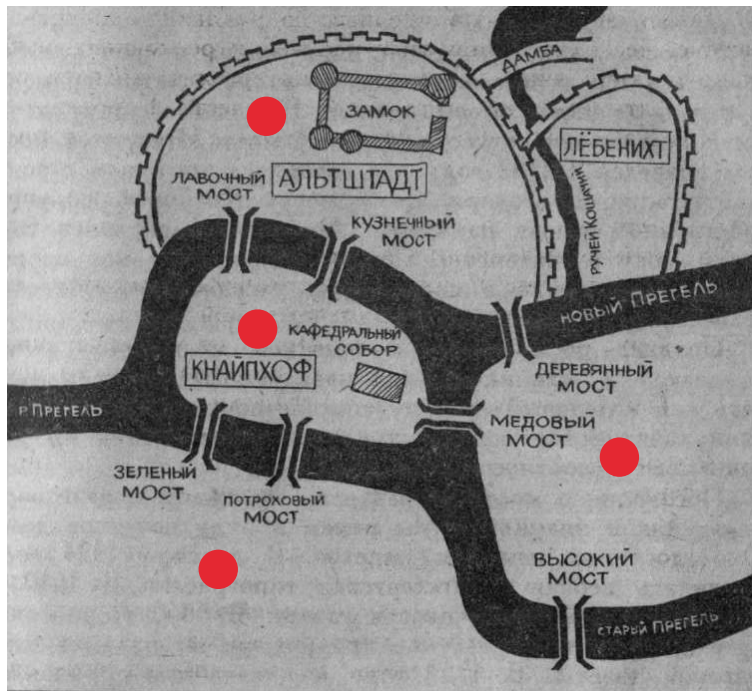
# Задача о семи мостах Кёнигсберга



Как пройти по всем городским мостам, не проходя ни по одному из них дважды?



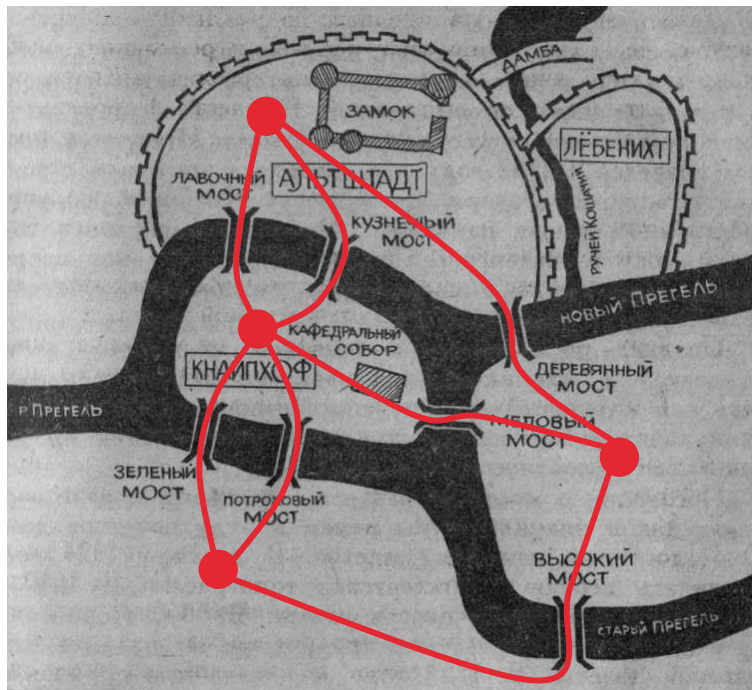
# Задача о семи мостах Кёнигсберга



Каждый участок суши - вершина графа.



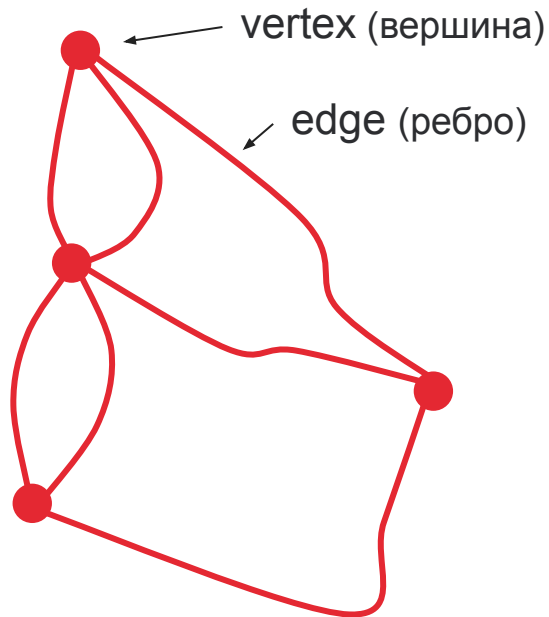
# Задача о семи мостах Кёнигсберга



Линий, которые соединяют  
вершины - рёбра графа.



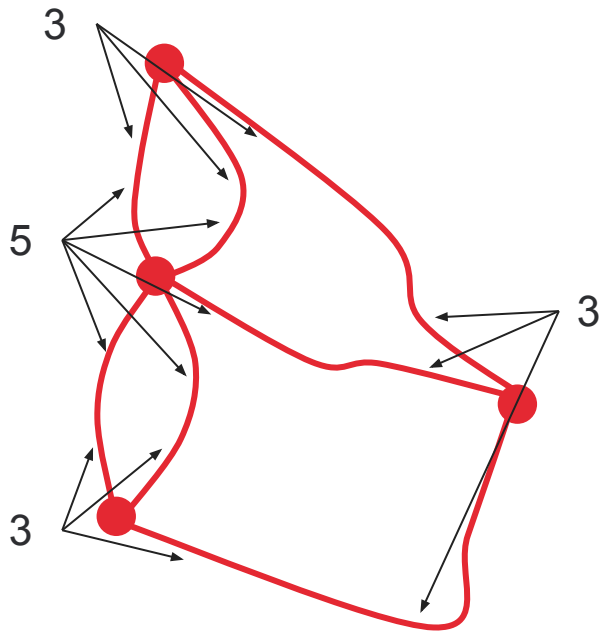
# Задача о семи мостах Кёнигсберга



Граф — абстрактный математический объект, представляющий собой множество вершин графа и набор рёбер, которые соединяют пары вершин.



# Задача о семи мостах Кёнигсберга

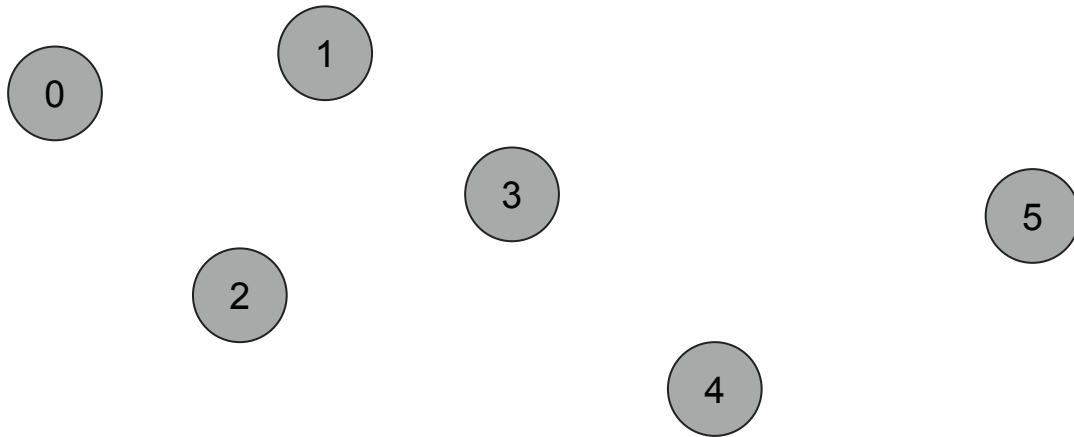


Для наличия Эйлера цикла в графе нужно, чтобы у каждой вершины её степень была чётной.

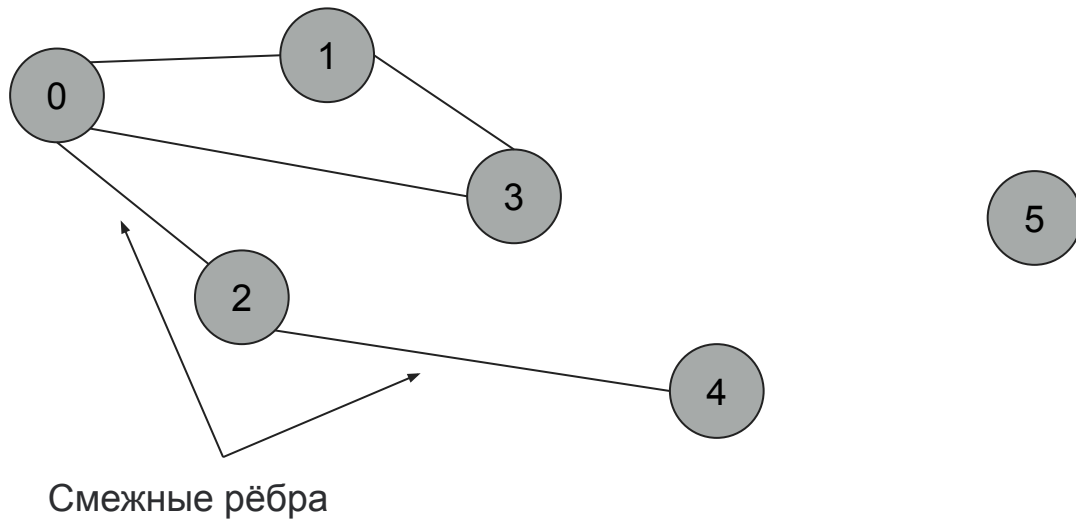




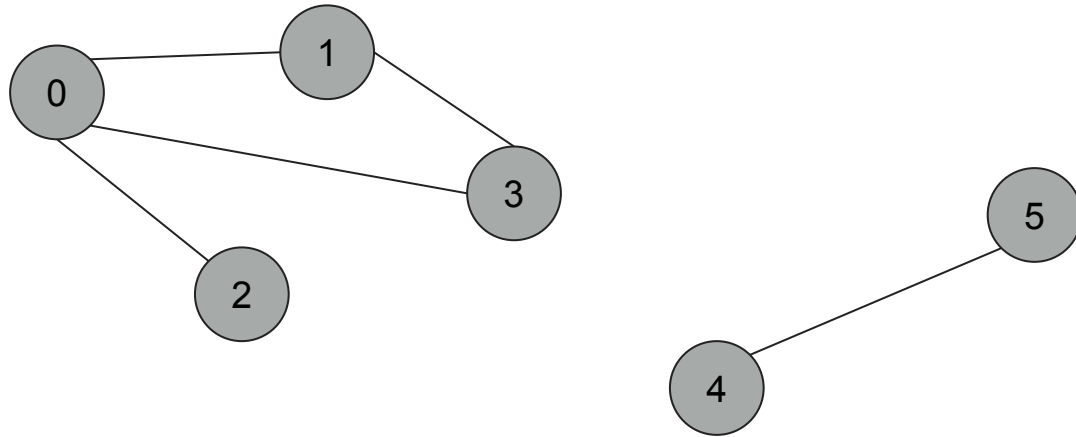
# Граф, состоящий из вершин



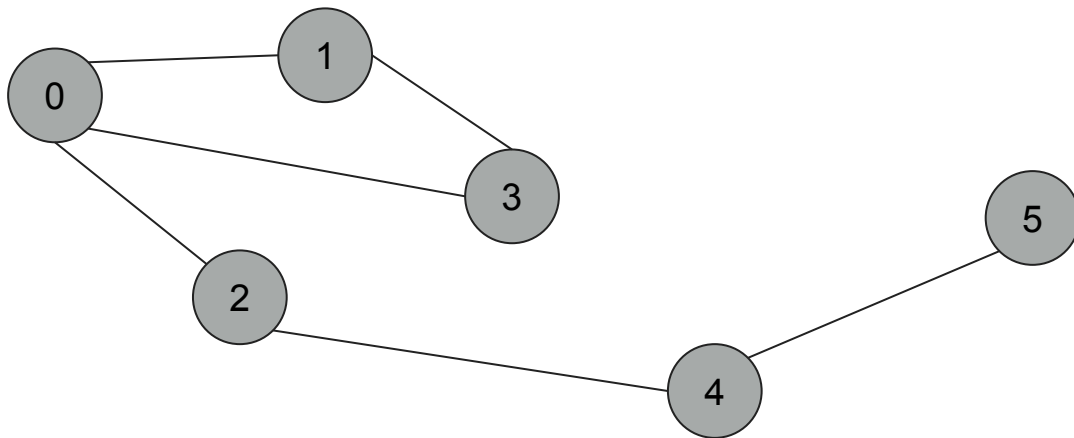
# Граф с изолированной вершиной



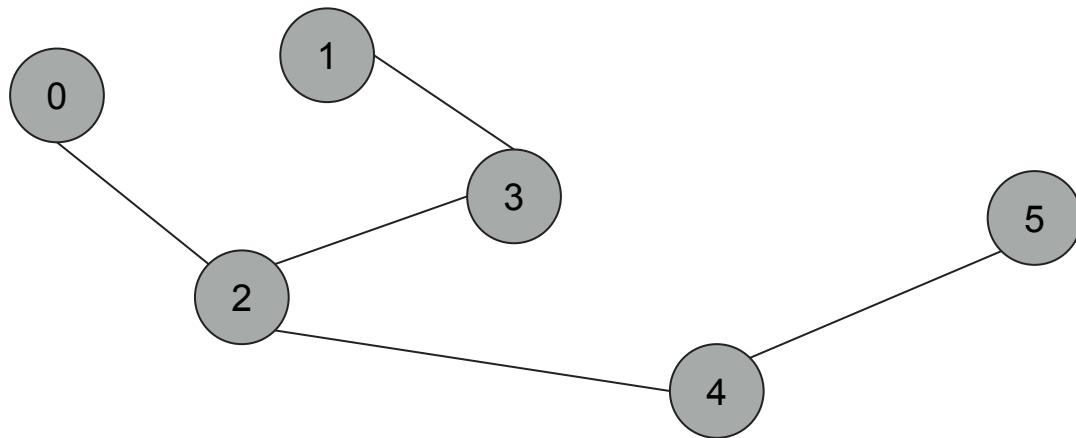
# Несвязный граф



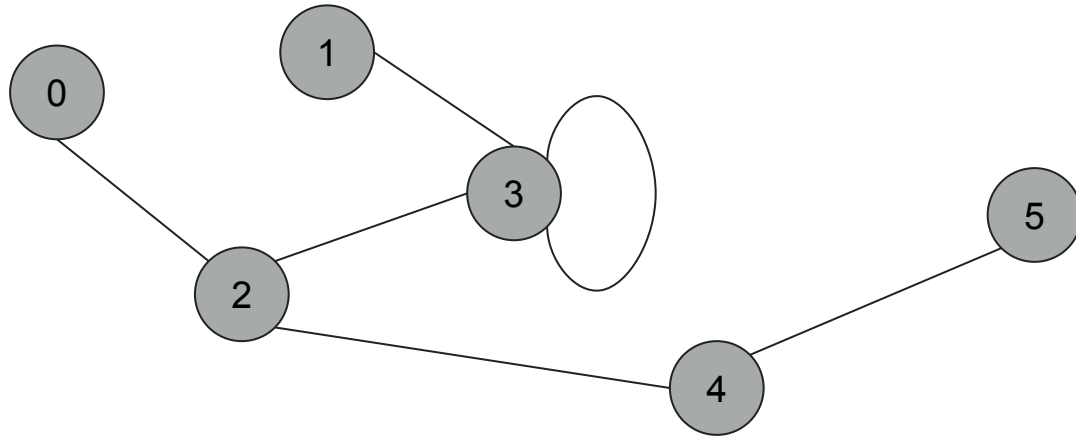
# Связный граф, имеющий цикл



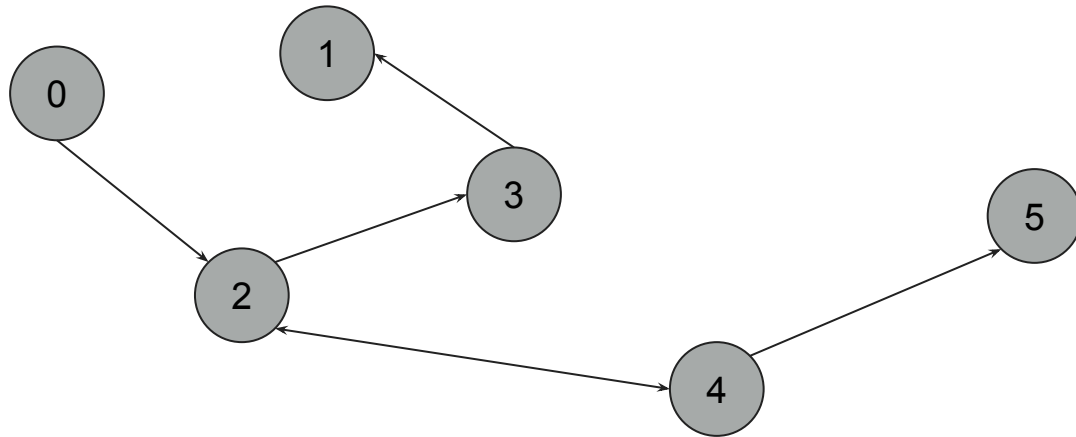
# Связный граф без циклов



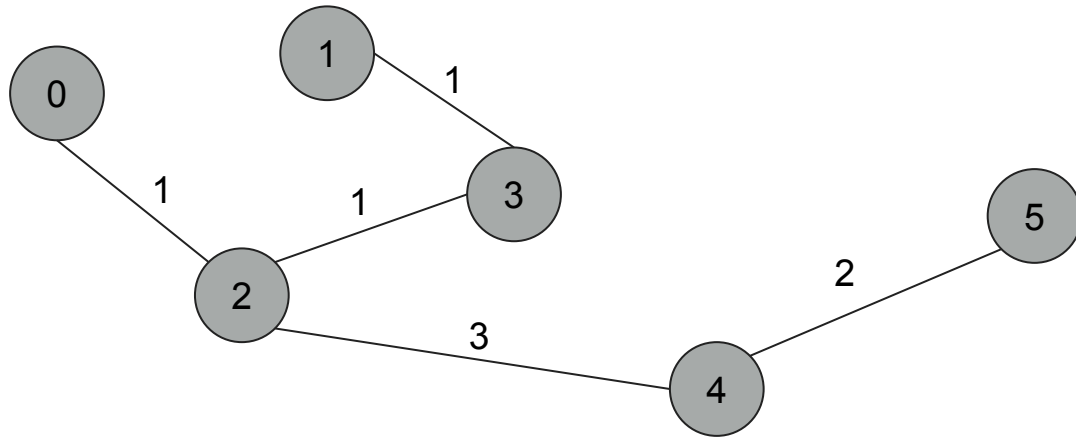
# Связный граф с петлёй



# Ориентированный граф

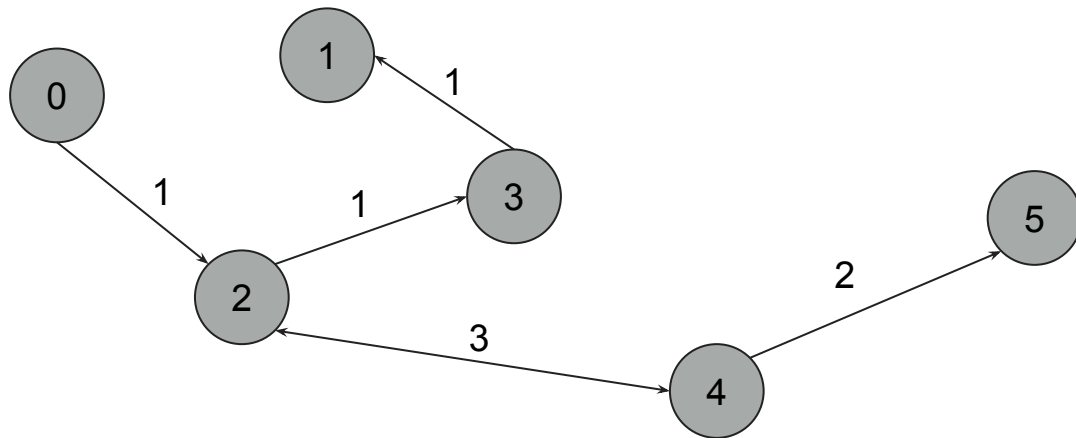


# Взвешенный граф





# Взвешенный ориентированный граф



# Задачи теории графов

- Задача о семи мостах Кёнигсберга
- Проблема четырёх красок
- Задача коммивояжёра



# Сферы применения теории графов

- Химия (описание структур, путей сложных реакций)
- Коммуникационные и транспортные системы (протоколы маршрутизации)
- Схемотехника (топология межсоединений элементов на печатной плате или микросхеме)
- Информатика и программирование
- Экономика
- Логистика



# Итоги:

## Теория

- Задача о семи мостах
- Определение графа
- Классификация графов
- Сферы применения

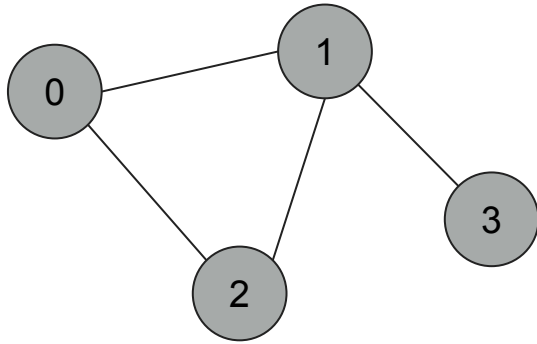


# План

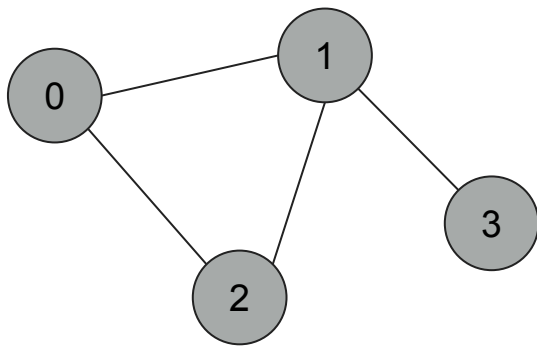
- Представление графов в Python
  - Матрицы смежности
  - Списки смежности
  - Списки рёбер



# Матрицы смежности



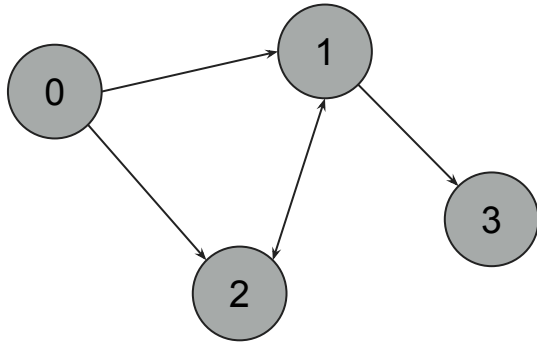
# Матрицы смежности



	0	1	2	3
0	0	1	1	0
1	1	0	1	1
2	1	1	0	0
3	0	1	0	0

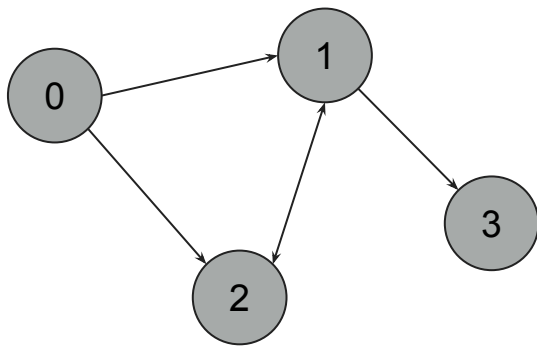


# Матрицы смежности





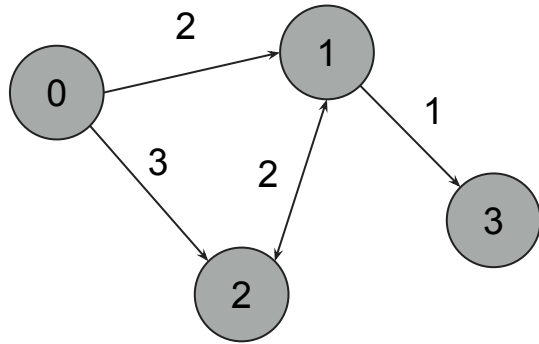
# Матрицы смежности



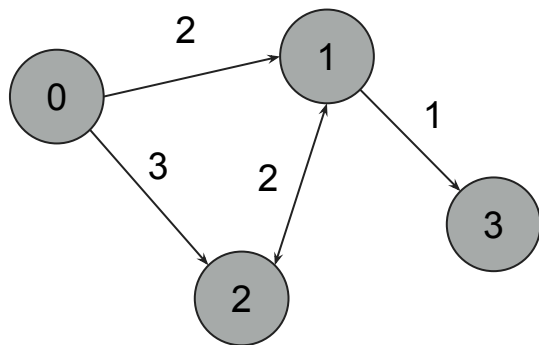
	0	1	2	3
0	0	1	1	0
1	0	0	1	1
2	0	1	0	0
3	0	0	0	0



# Матрицы смежности



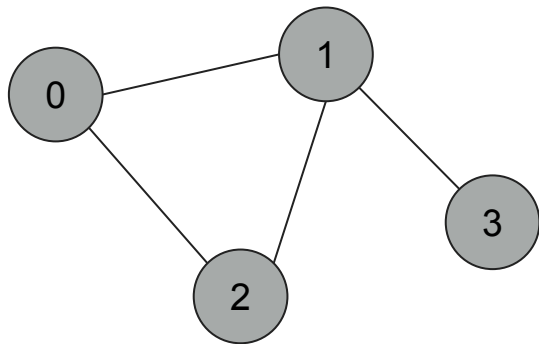
# Матрицы смежности



	0	1	2	3
0	0	2	3	0
1	0	0	2	1
2	0	2	0	0
3	0	0	0	0



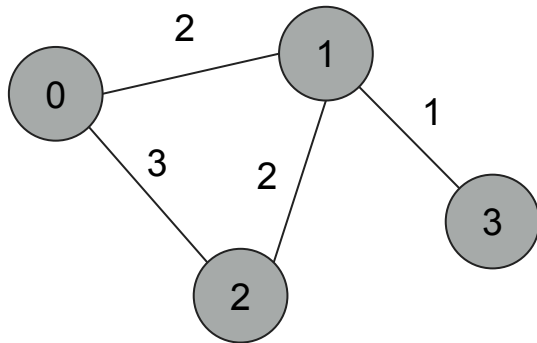
# Списки смежности



<b>0</b>	1	2	
<b>1</b>	0	2	3
<b>2</b>	0	1	
<b>3</b>	1		



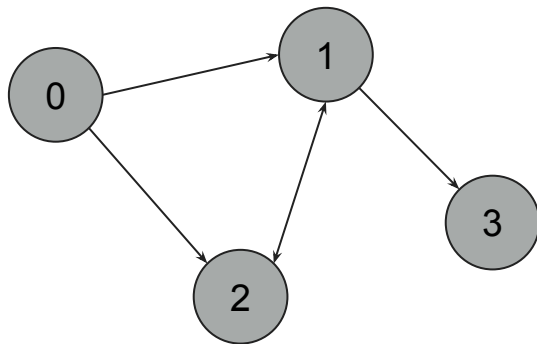
# Списки смежности



<b>0</b>	1, 2	2, 3	
<b>1</b>	0, 2	2, 2	3, 1
<b>2</b>	0, 3	1, 2	
<b>3</b>	1, 1		



# Список рёбер



0	1
0	2
1	2
1	3
2	1



# Итоги:

## Практика

- Представление графов в Python
  - Матрицы смежности
  - Списки смежности
  - Списки рёбер



# План

- Поиск кратчайшего пути в ширину (Breadth-First Search)



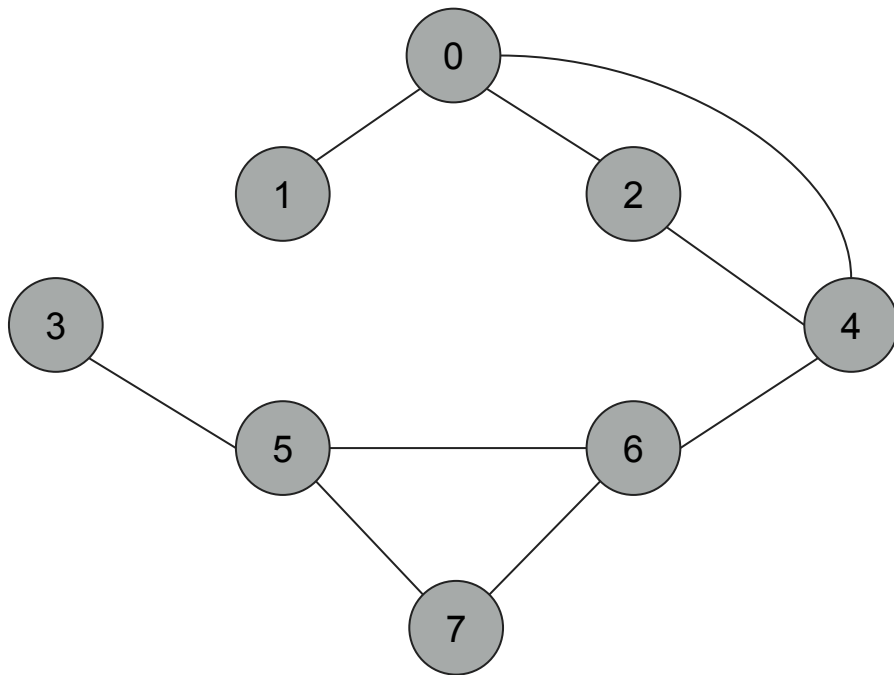


# Поиск кратчайшего пути в ширину

1. Поместить вершину, с которого начинается поиск, в пустую очередь.
2. Извлечь из начала очереди вершину.
  - а. Если вершина является целевой, то завершить поиск.
  - б. В противном случае, в конец очереди добавляются все смежные вершины, которые ещё не пройдены и не находятся в очереди.
3. Если очередь пуста, то все вершины графа были просмотрены, следовательно, целевой узел недостижим из начального; завершить поиск.



# Поиск кратчайшего пути в ширину



# Итоги:

## Теория и Практика

- Поиск кратчайшего пути в ширину (Breadth-First Search)

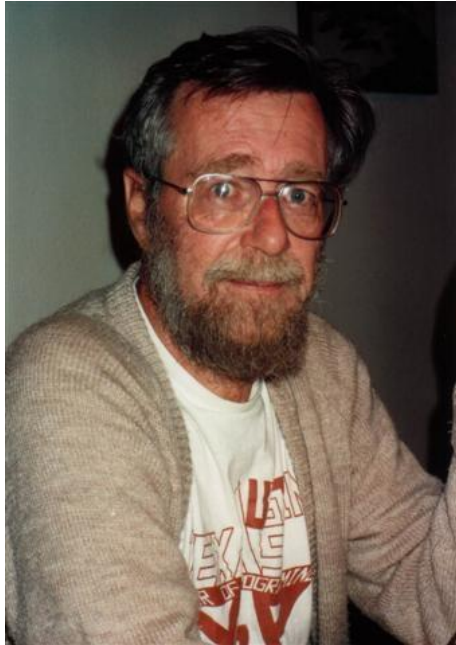


# План

- Поиск кратчайшего пути по алгоритму Дейкстры



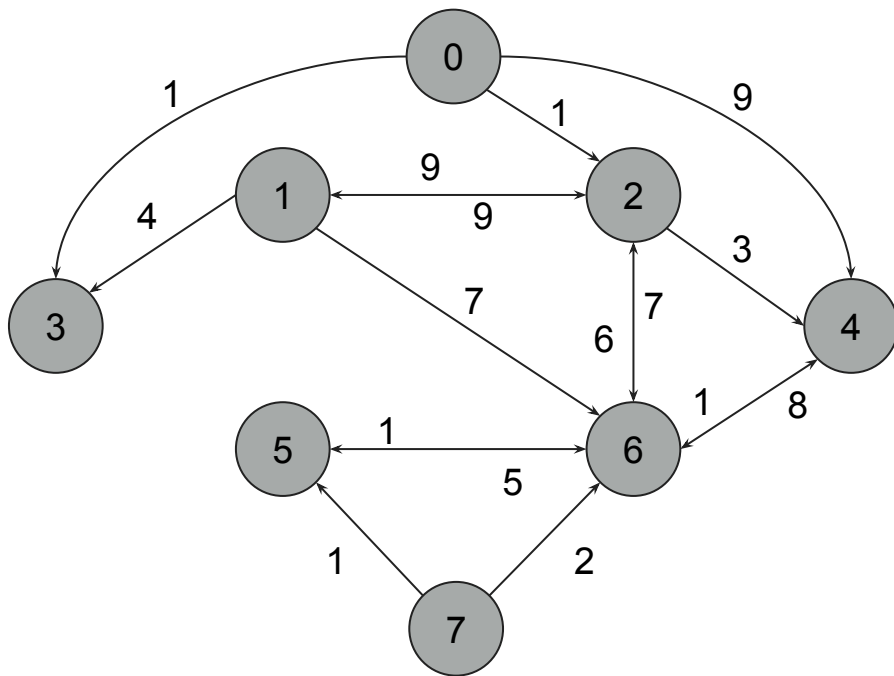
# Алгоритм Дейкстры



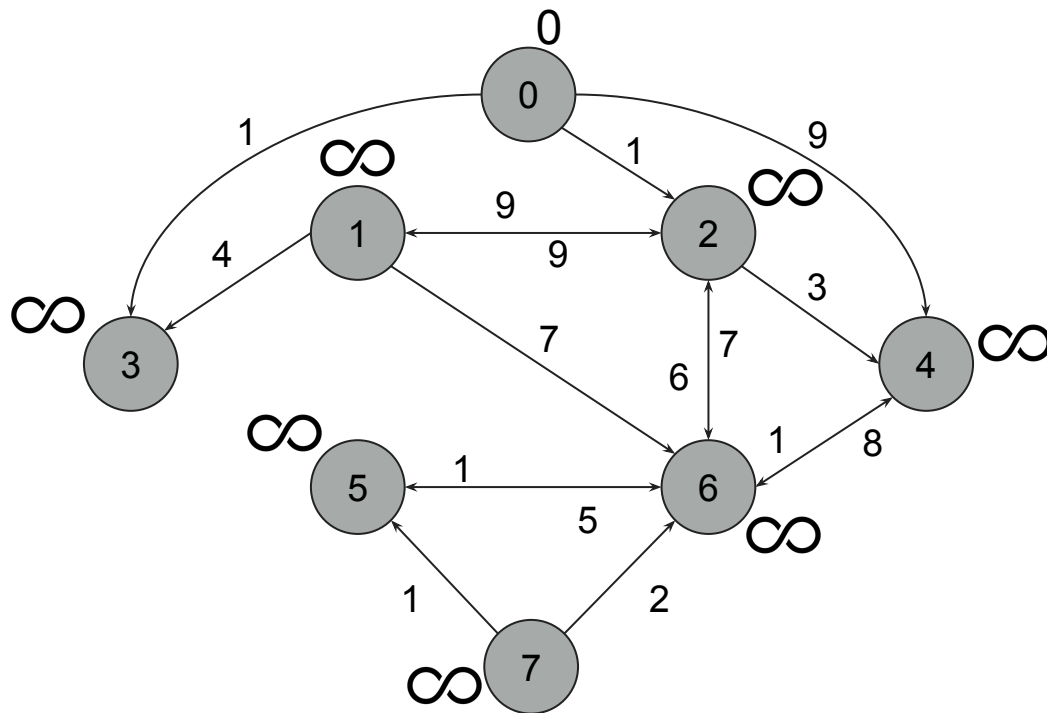
Нидерландский учёный Эдсгер Дейкстра изобрёл в 1959 году алгоритм нахождения кратчайшего пути от одной из вершин графа до всех остальных.



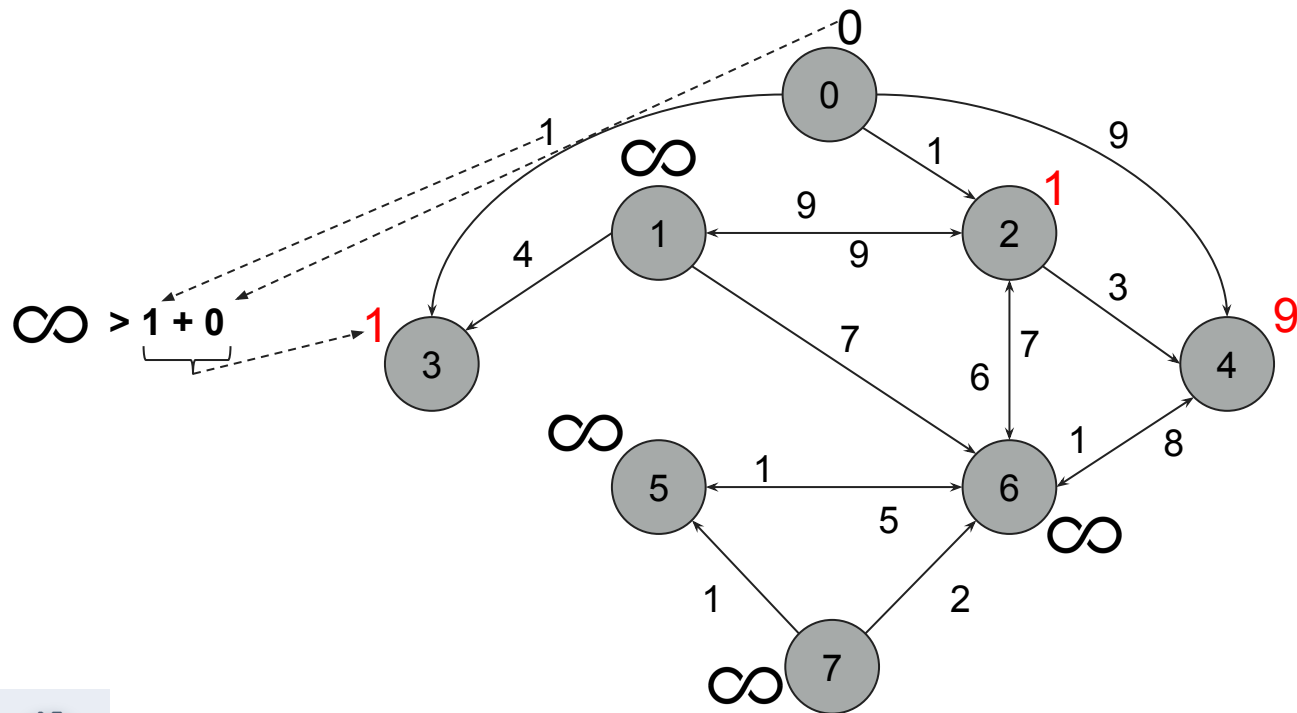
# Алгоритм Дейкстры



# Алгоритм Дейкстры

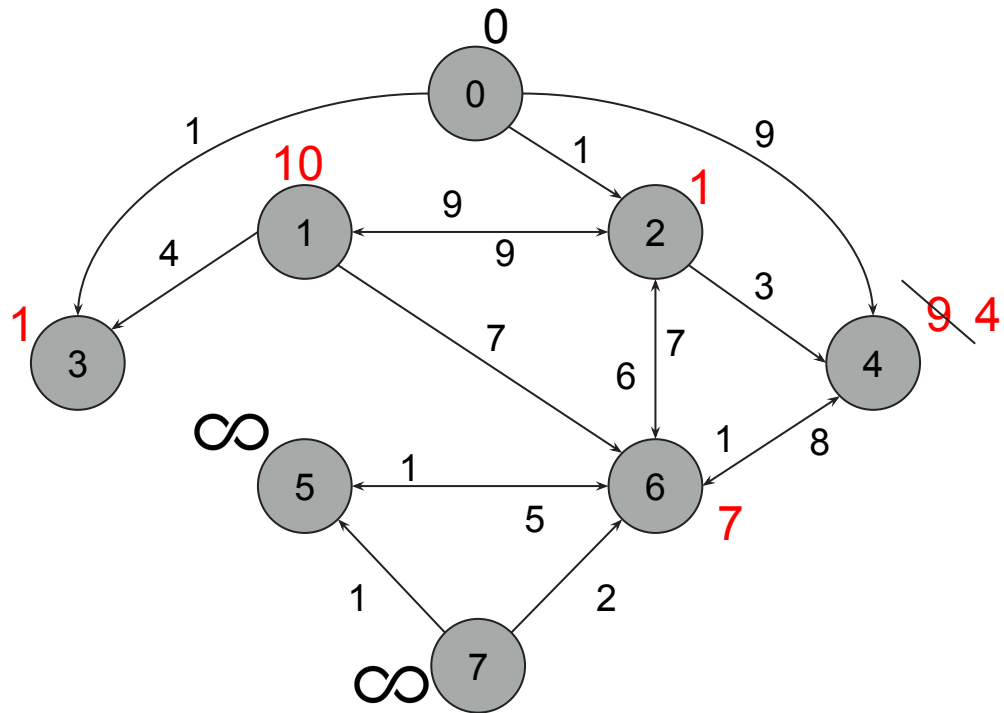


# Алгоритм Дейкстры

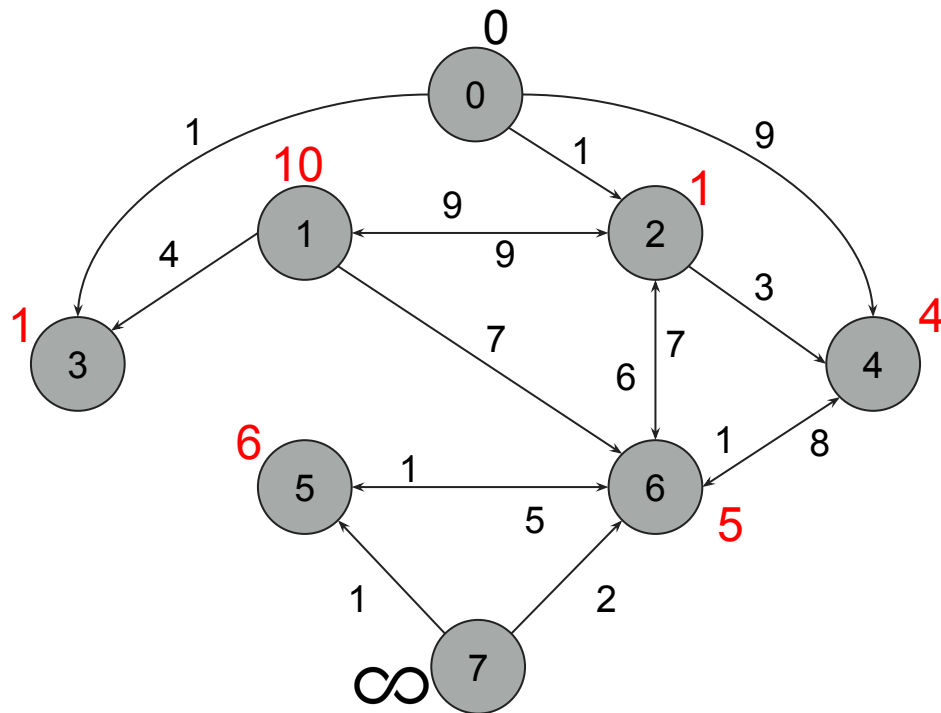




# Алгоритм Дейкстры



# Алгоритм Дейкстры



# Итоги:

## Теория и Практика

- Поиск кратчайшего пути по алгоритму Дейкстры



# Домашнее задание

1. На улице встретились  $N$  друзей. Каждый пожал руку всем остальным друзьям (по одному разу).  
Сколько рукопожатий было?

Примечание: Решите задачу при помощи построения графа.

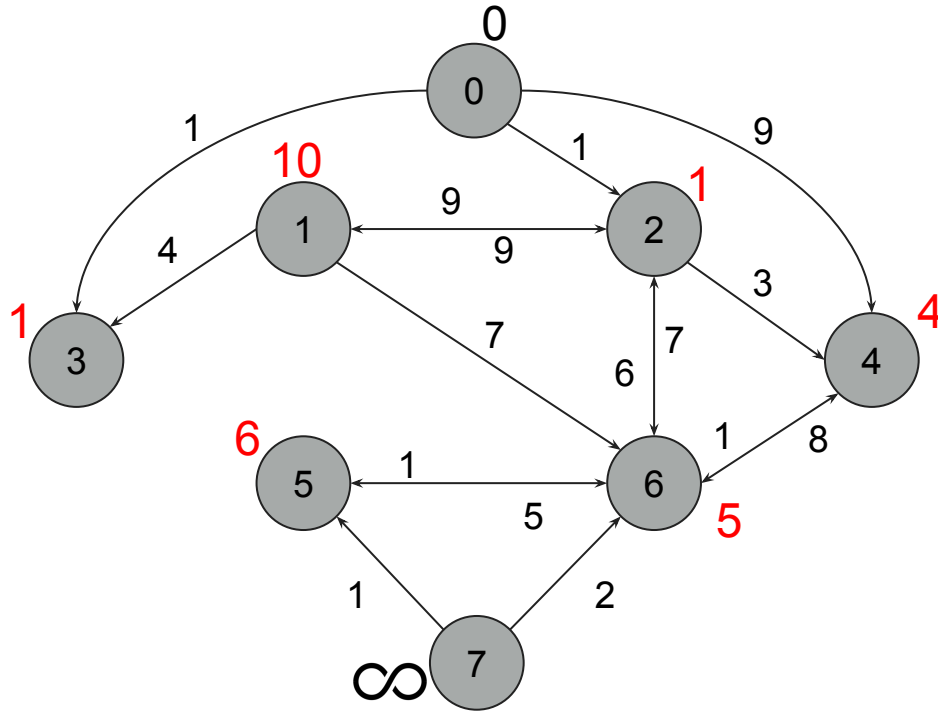


# Домашнее задание

2. Доработать алгоритм Дейкстры, чтобы он дополнительно возвращал список вершин, которые необходимо обойти.



# Алгоритм Дейкстры



0	0				
1	0	2	1		
2	0	2			
3	0	3			
4	0	2	4		
5	0	2	4	6	5
6	0	2	4	6	
7	нет пути				



# Домашнее задание

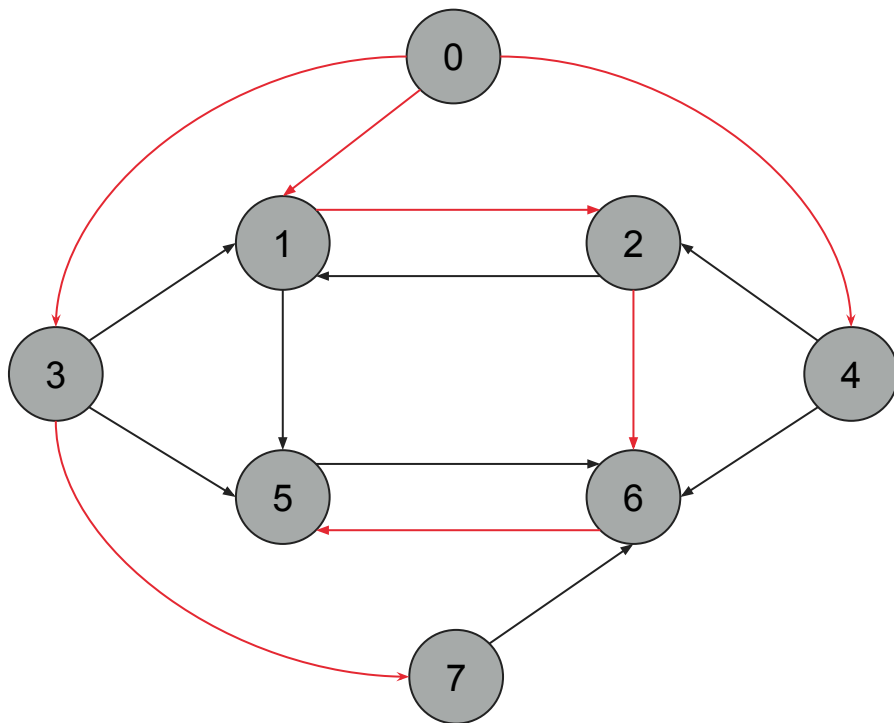
3. Написать программу, которая обходит не взвешенный ориентированный граф без петель, в котором все вершины связаны, по алгоритму поиска в глубину (Depth-First Search).

Примечания:

- граф должен храниться в виде списка смежности;
- генерация графа выполняется в отдельной функции, которая принимает на вход число вершин.



# Поиск в глубину





# План

- Разбор домашнего задания

