



# Теория алгоритмов

Лекция 4

Графы. Способы представления и основные алгоритмы обработки

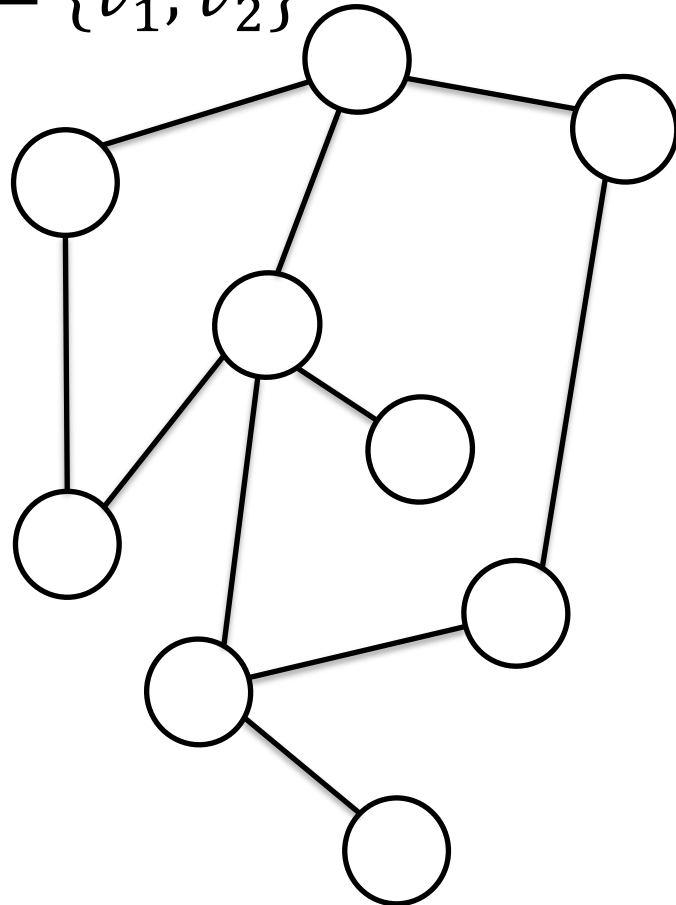
The screenshot shows a software interface for a Petri net simulator. On the left, a Solution Explorer displays a project structure with files like `datawriter.h`, `event.h`, `gates.h`, `logiclevel.h`, `netlist.h`, `netlist_loader.h`, `meth.h`, `simulation_data.h`, `simulator.h`, `datawriter.cpp`, `netlist.cpp`, `netlist_loader.cpp`, `simulation_data.cpp`, and `simulator.cpp`. The main window displays a Petri net diagram with nodes (circles) and transitions (squares). Nodes are labeled with numbers: 1, 6, 8, 11, 13, 15, 17, 22, 25, 27. Transitions are labeled with numbers: 1, 2, 3, 4, 6, 7, 8, 9. Some nodes and transitions are highlighted in red or black. The diagram is overlaid on a code editor showing the implementation of the Petri net in C++ within a file named `gates.cpp`. The code includes a header `#include "gates.h"` and defines a class `gate` with methods `operate()` and `process()`. The `operate()` method contains logic for handling transitions and updating the state of the Petri net.

## Что такое граф?

$$G: (V, E)$$

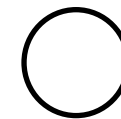
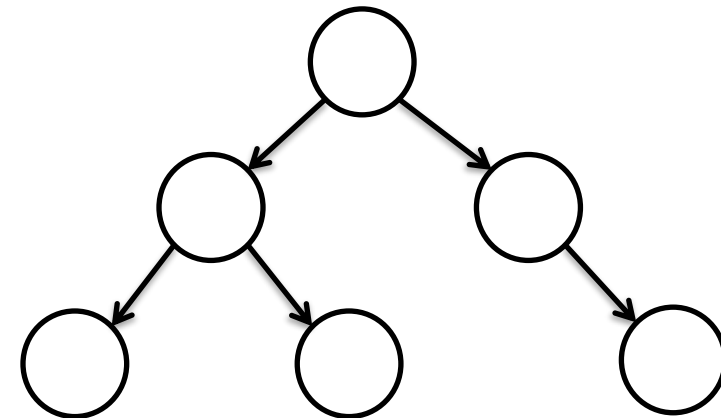
$$v_i \in V, \quad V \neq \emptyset$$

$$e_i \in E : e = \{v_1, v_2\}$$



**Порядок графа** – число вершин графа

**Степень вершины** – число рёбер, инцидентных вершине

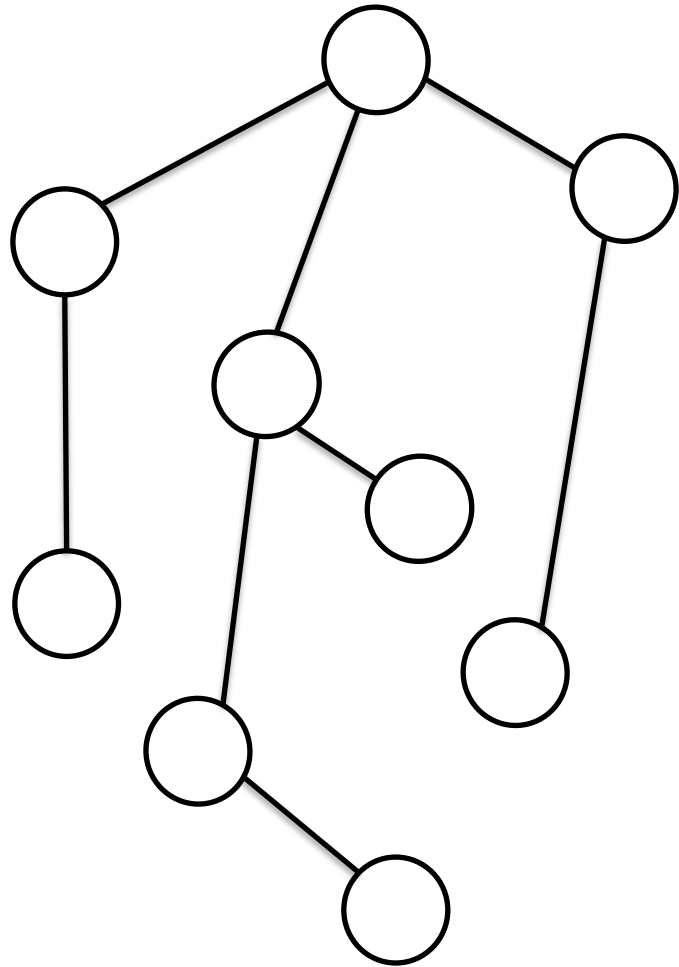


Вершина графа

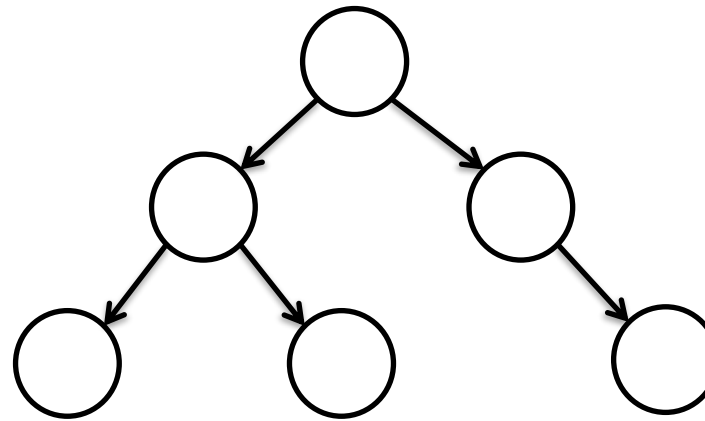


Ребро графа

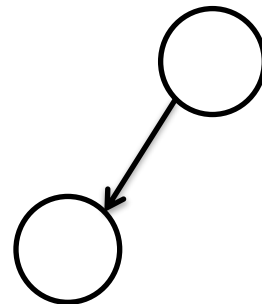
## Теория графов: основные определения



Неориентированный граф



Ориентированный граф  
(орграф)

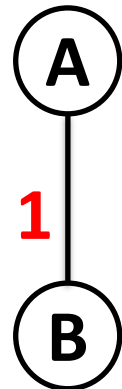
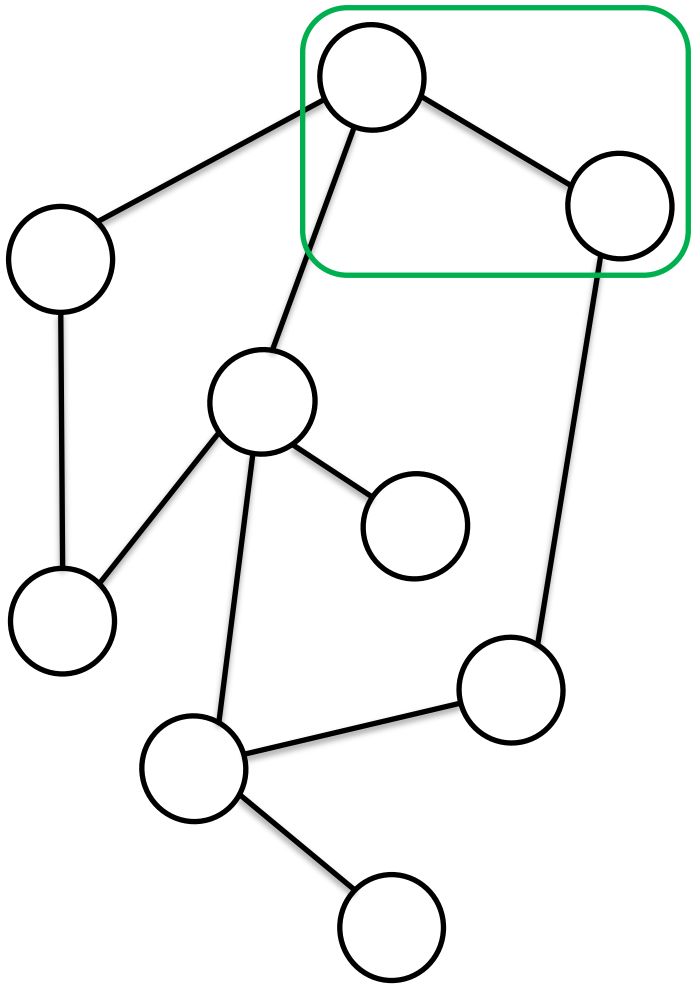


Предок

Ориентированное ребро

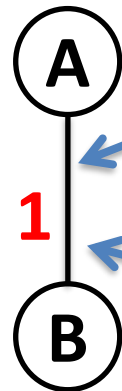
Потомок

## Теория графов: основные определения (2)



Вершины A и B -  
смежные  
вершины

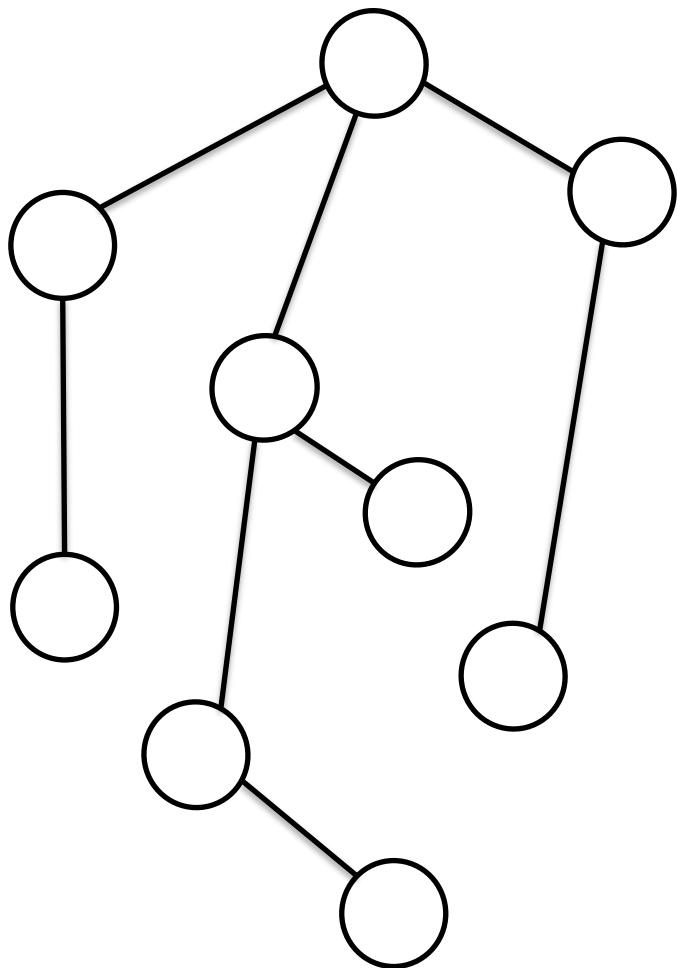
Вершина A  
инцидентна ребру 1



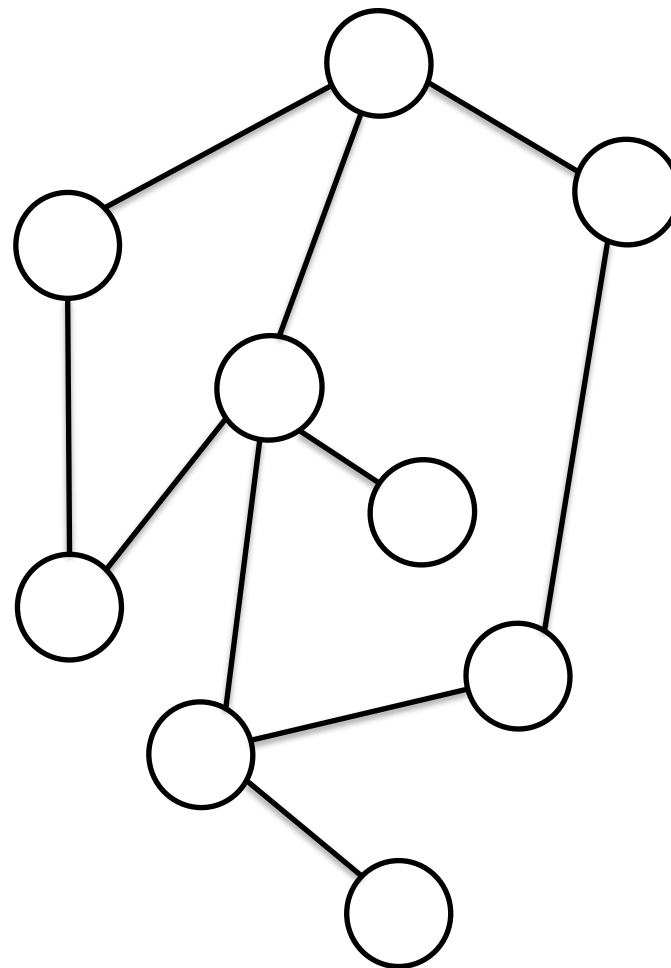
(ребро 1  
инцидентно  
вершине A)

Вершина B  
инцидентна ребру  
1 (ребро 1  
инцидентно  
вершине B)

## Теория графов: основные определения (3)

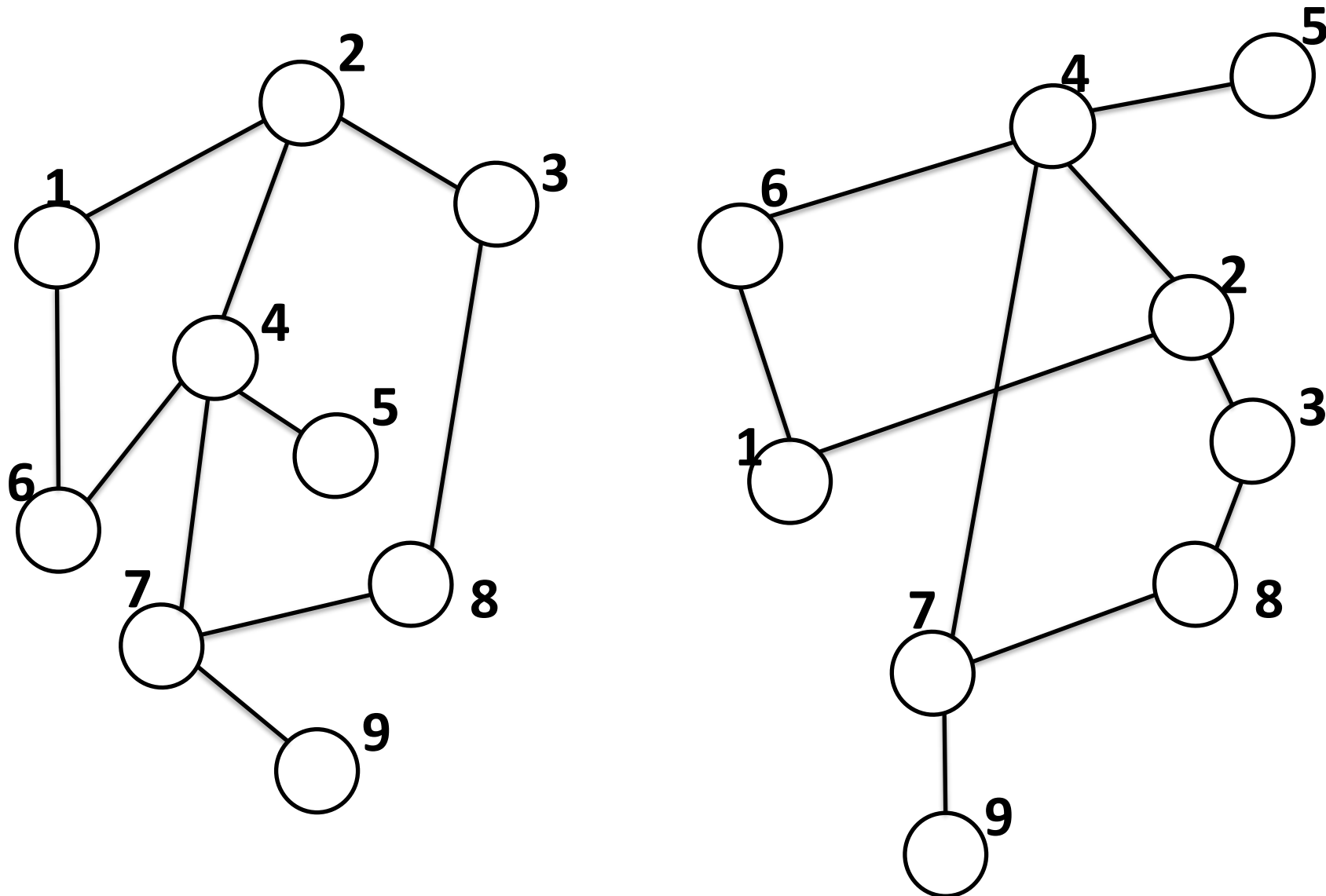


Не содержит циклов  
(дерево)

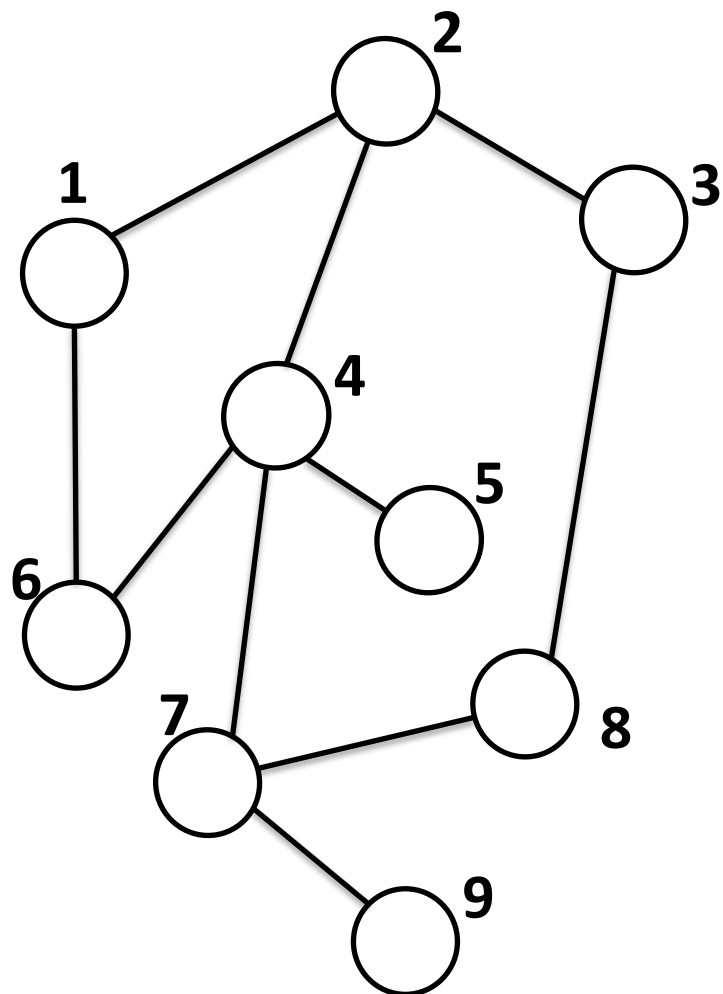


Содержит циклы

## Изоморфизм графов



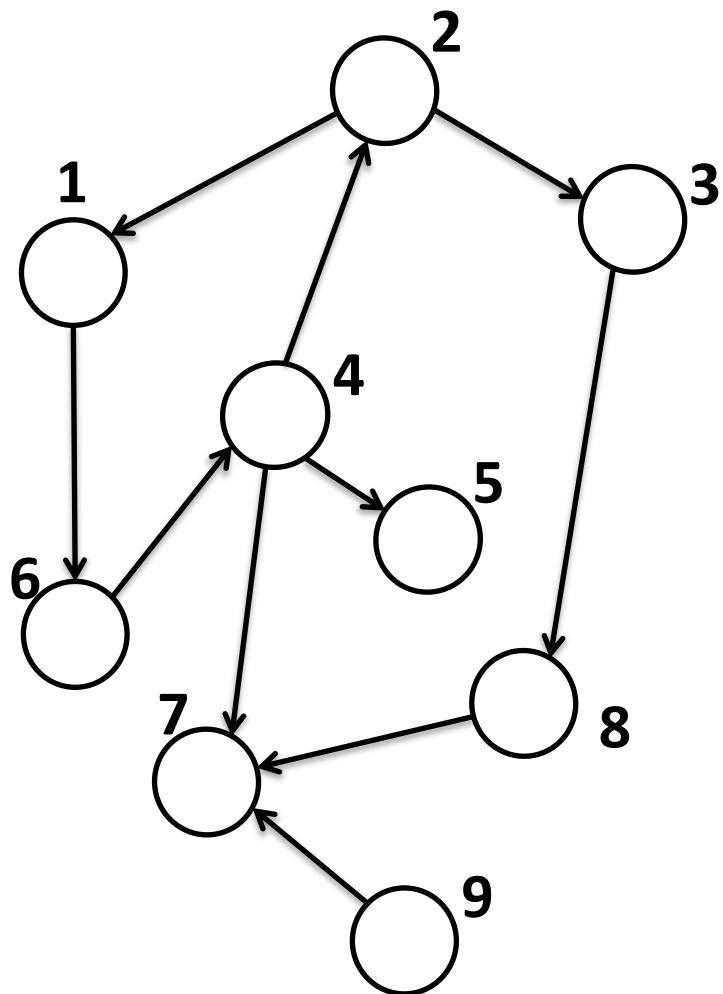
## Способы представления графа: матрица смежности (1)



Матрица смежности

|   | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| 1 |   | 1 |   |   |   | 1 |   |   |   |
| 2 | 1 |   | 1 | 1 |   |   |   |   |   |
| 3 |   | 1 |   |   |   |   |   | 1 |   |
| 4 |   | 1 |   |   | 1 | 1 | 1 |   |   |
| 5 |   |   |   | 1 |   |   |   |   |   |
| 6 | 1 |   |   | 1 |   |   |   |   |   |
| 7 |   |   |   | 1 |   |   |   | 1 | 1 |
| 8 |   |   | 1 |   |   |   | 1 |   |   |
| 9 |   |   |   |   |   |   | 1 |   |   |

## Способы представления графа: матрица смежности (2)



Матрица смежности для орграфа

|   | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| 1 |   |   |   |   |   | 1 |   |   |   |
| 2 | 1 |   | 1 |   |   |   |   |   |   |
| 3 |   |   |   |   |   |   |   | 1 |   |
| 4 |   | 1 |   |   | 1 |   | 1 |   |   |
| 5 |   |   |   |   |   |   |   |   |   |
| 6 |   |   |   | 1 |   |   |   |   |   |
| 7 |   |   |   |   |   |   |   |   |   |
| 8 |   |   |   |   |   |   | 1 |   |   |
| 9 |   |   |   |   |   |   | 1 |   |   |





## Представление графов в виде списков и массивов (1)

Вариант 1 - списки

```
struct GraphNode {  
    std::vector<GraphNode *> nodes;  
  
    std::string          name;  
    size_t              index;  
  
    double              value;  
    double              parameter1,  
                       parameter2;  
  
};
```

## Представление графов в виде списков и массивов (1)

Вариант 2 - массивы

```
struct Branch {  
    std::string name;  
    size_t      index;  
    ...  
};
```

```
struct Node {  
    std::vector<int> my_branches;  
    std::string      name;  
    size_t           index;  
    ...  
};
```

```
std::vector<Node *> nodes;  
std::vector<Branch *> branches;
```

## Алгоритм Дейкстры: обход элементов графа «в ширину» (1)

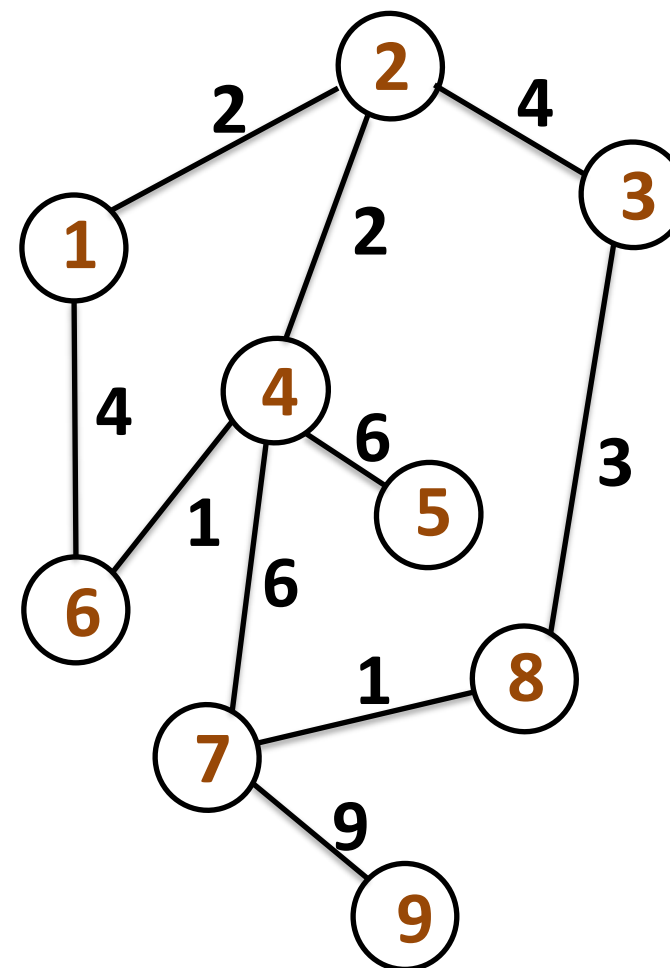
Алгоритм Дейкстры – алгоритм на графах, находящий кратчайшее расстояние от одной из вершин до всех остальных



Эдсгер Вибе Дейкстра

Для демонстрации алгоритма необходимо задать рёбрам весовые коэффициенты.

Такой граф называется взвешенным.

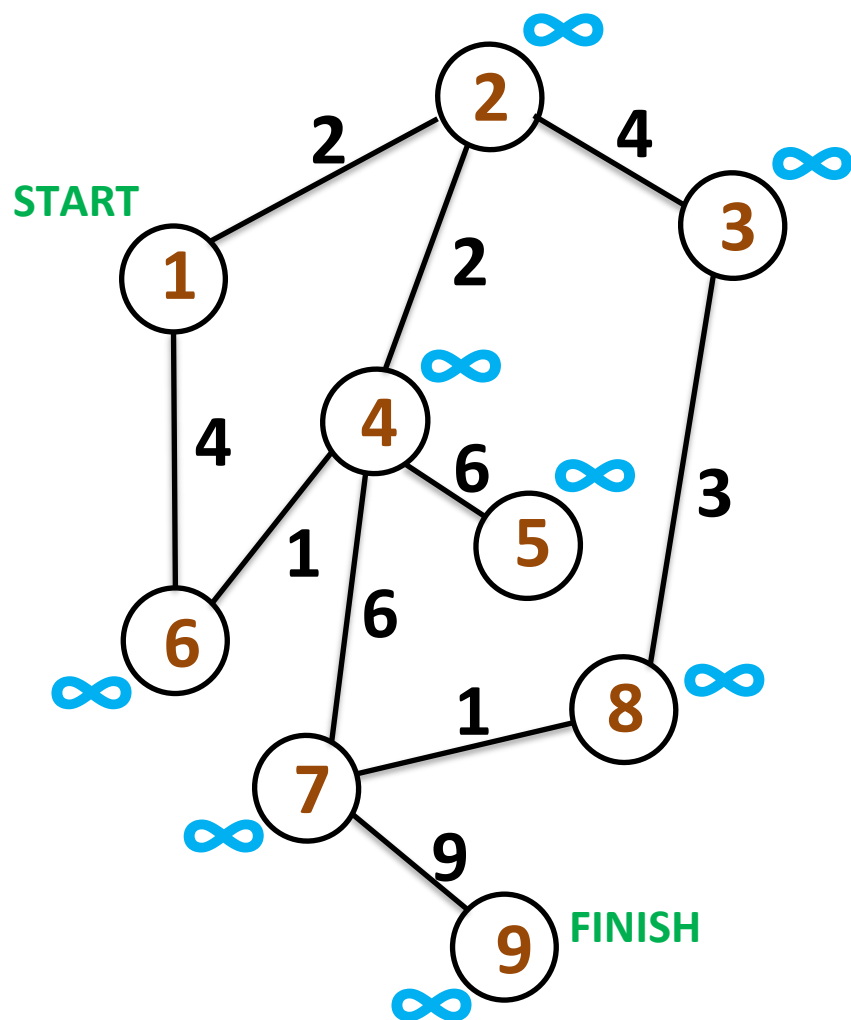


## Алгоритм Дейкстры: обход элементов графа «в ширину» (2)

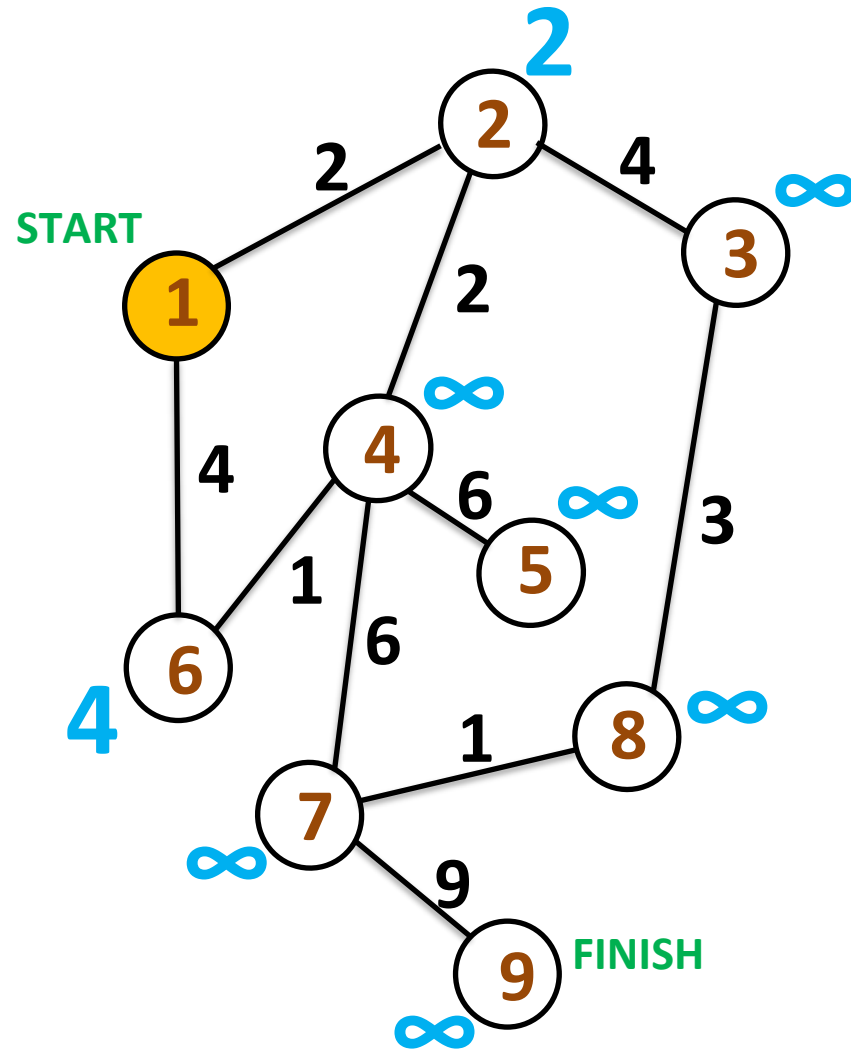
Шаги алгоритма Дейкстры «в ширину»:

1. задать для всех вершин графа значения расстояний, равные бесконечности;
2. выбрать начальную и конечную вершины графа (откуда и куда ищем кратчайшее расстояние);
3. начиная с выбранной вершины вести лексикографический обход смежных вершин;
4. для каждой смежной вершины проставить значение минимального получающегося суммарного расстояния;
5. повторять шаги 3,4, пока есть непосещённые вершины.

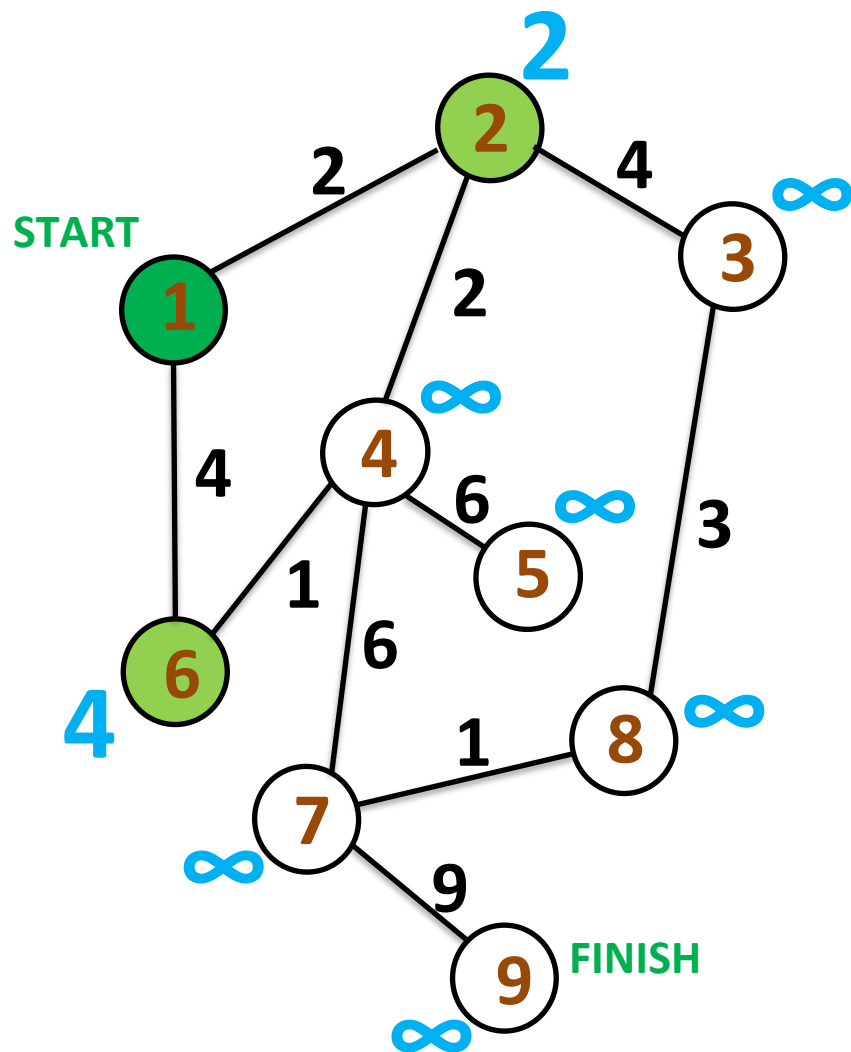
# Алгоритм Дейкстры (1)



## Алгоритм Дейкстры (2)

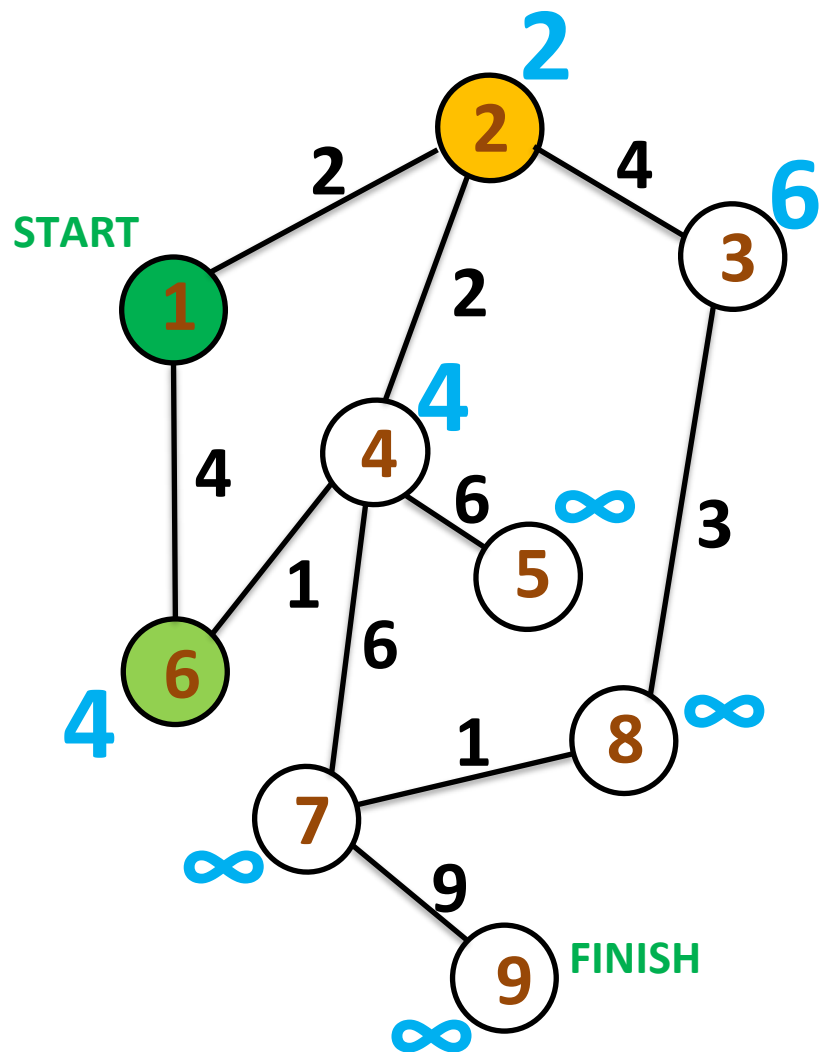


## Алгоритм Дейкстры (3)

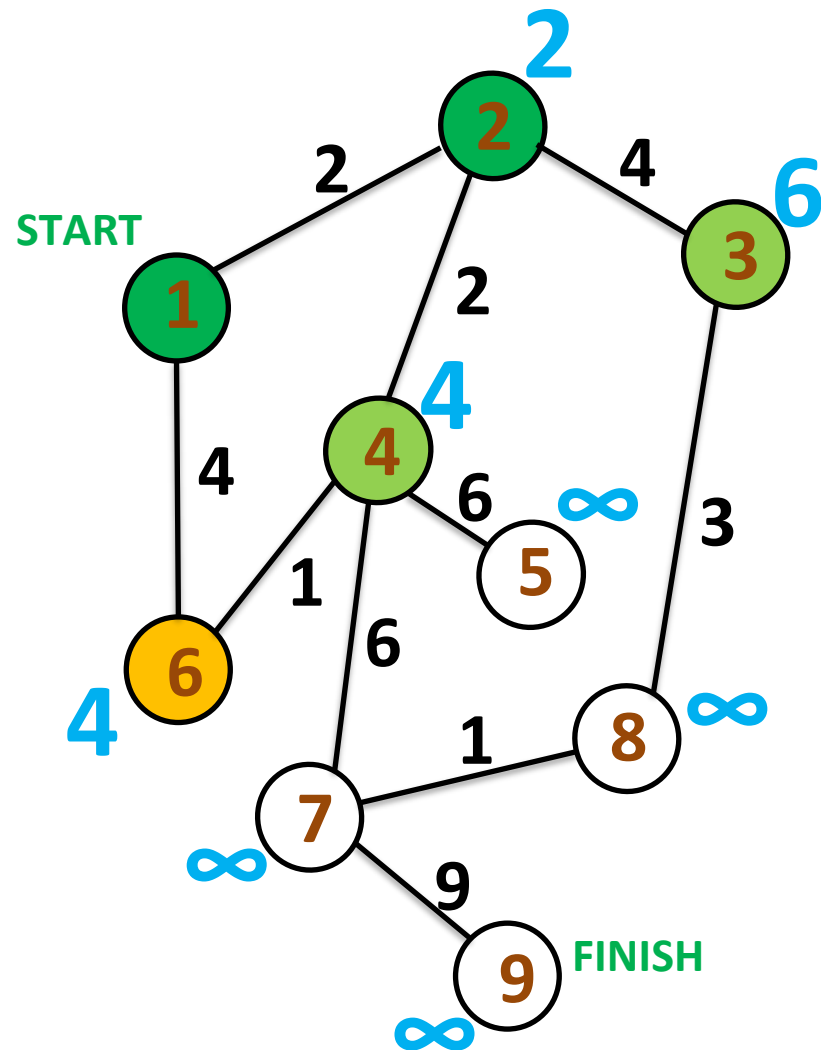




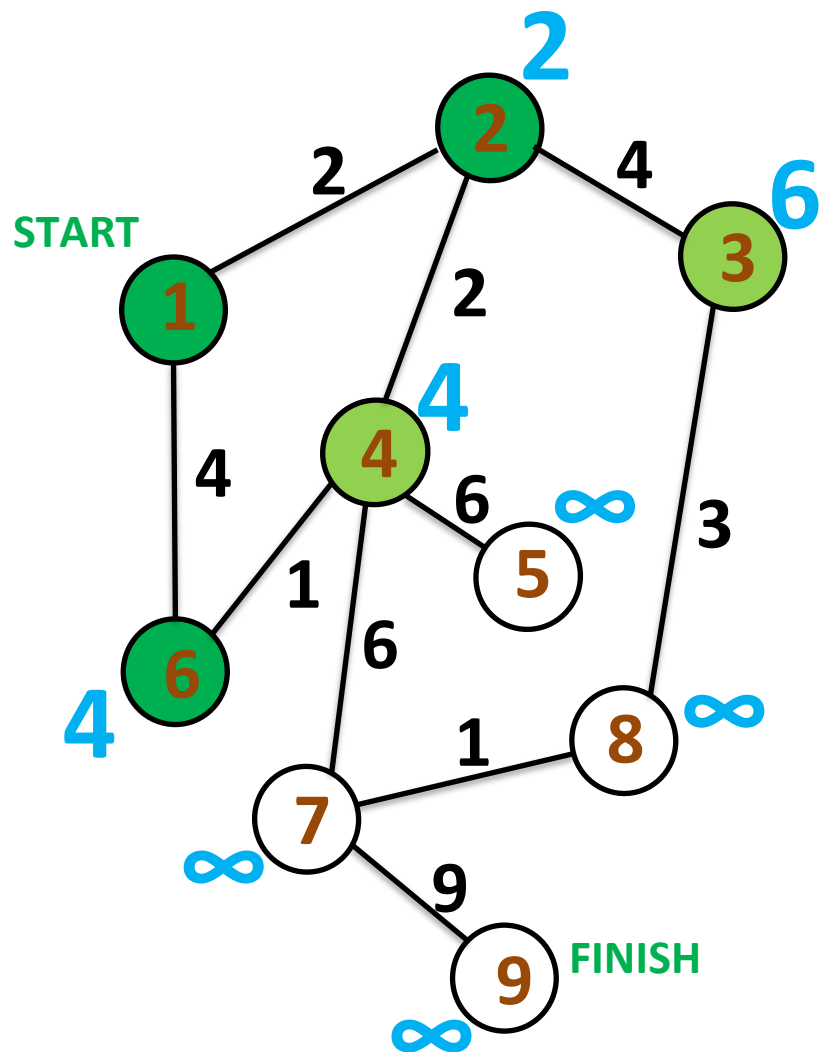
## Алгоритм Дейкстры (4)



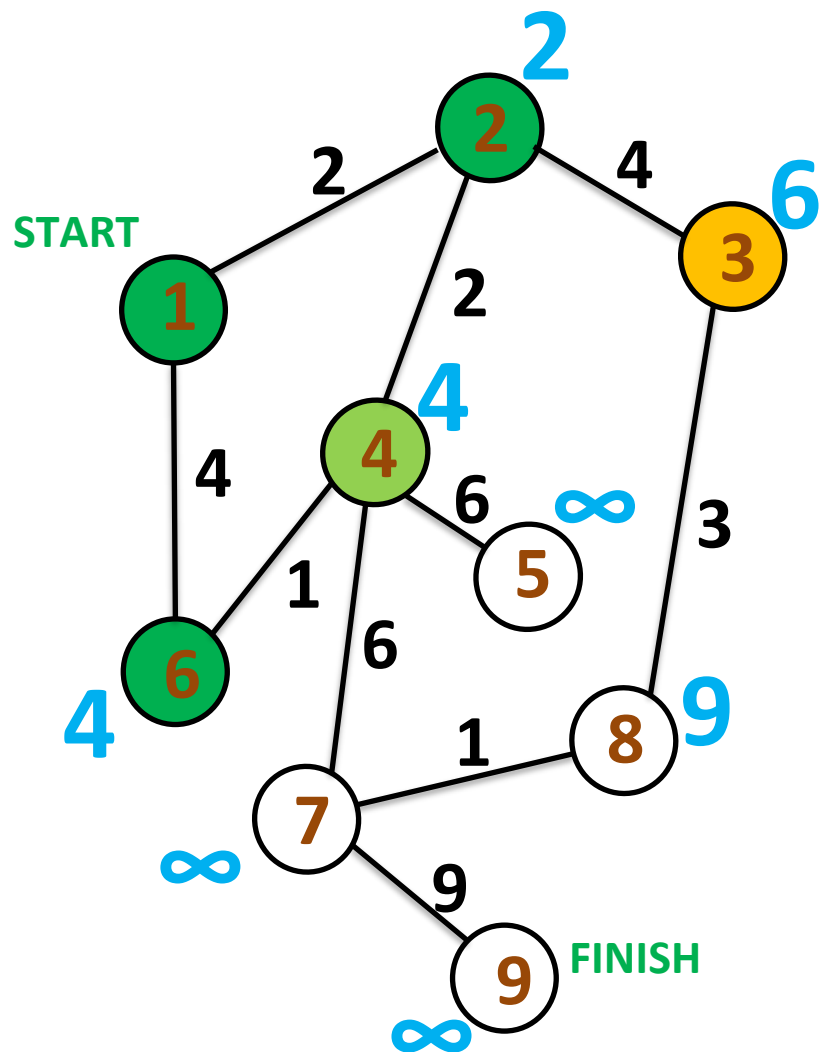
## Алгоритм Дейкстры (5)



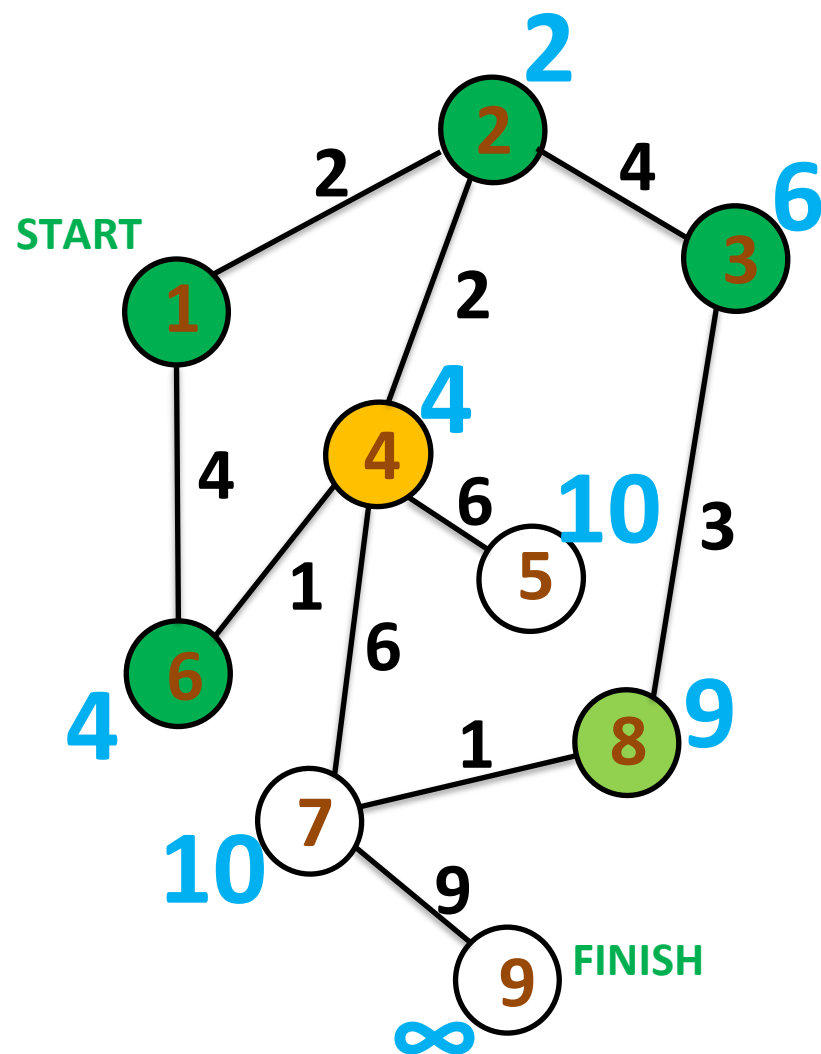
## Алгоритм Дейкстры (6)



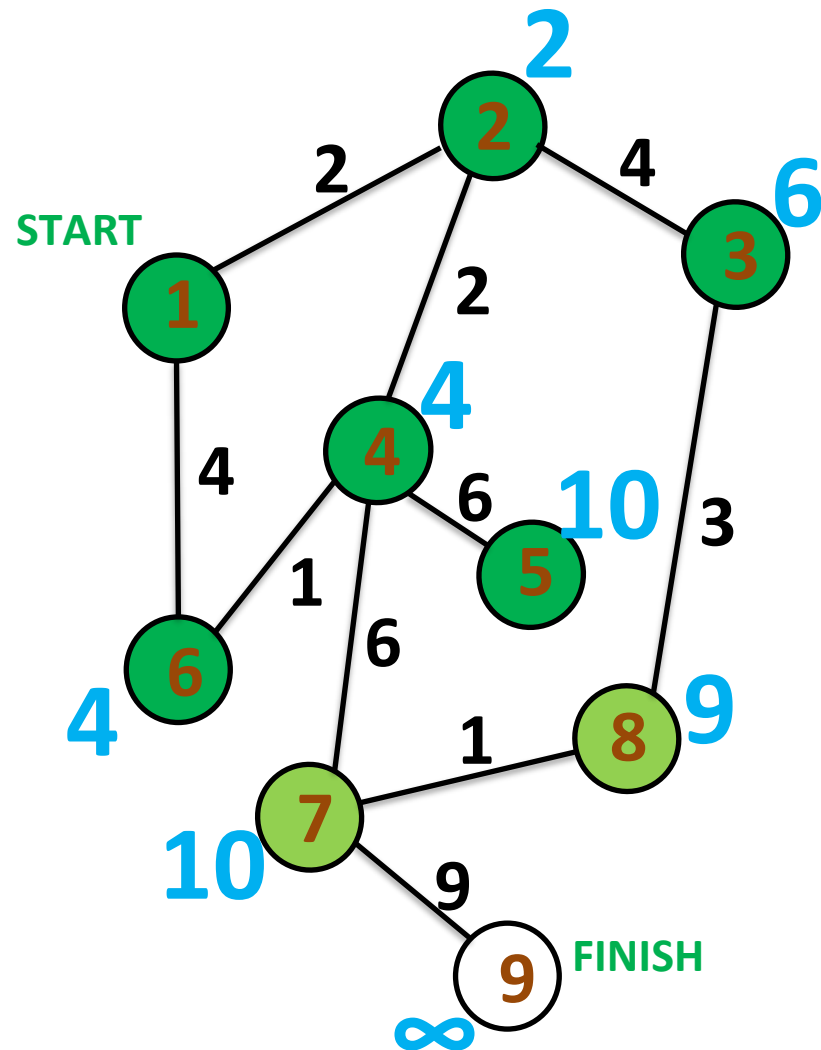
## Алгоритм Дейкстры (7)



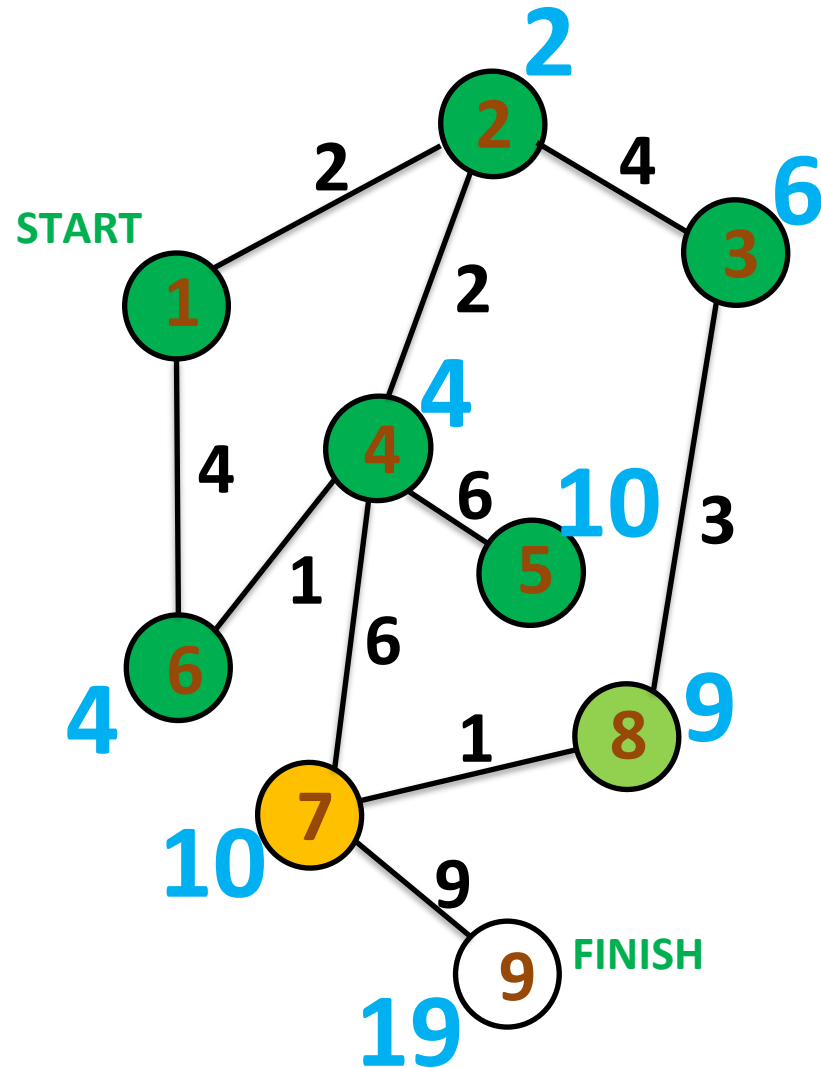
## Алгоритм Дейкстры (8)



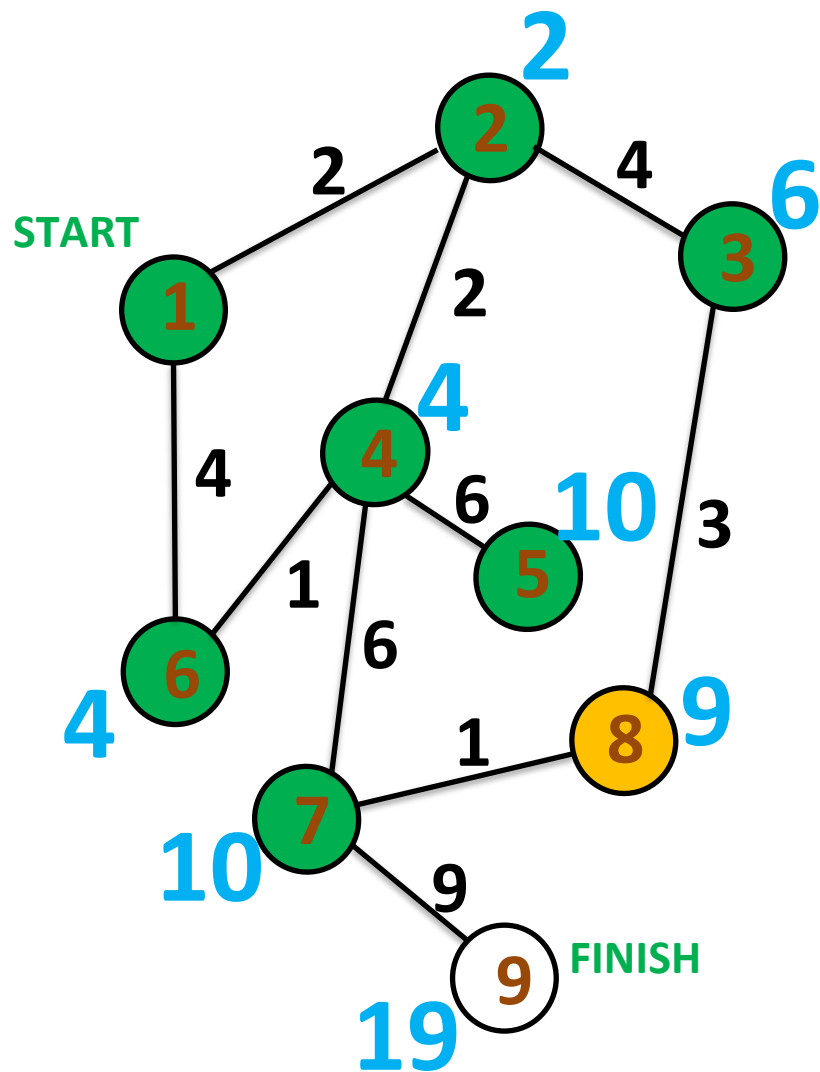
## Алгоритм Дейкстры (9)



## Алгоритм Дейкстры (10)

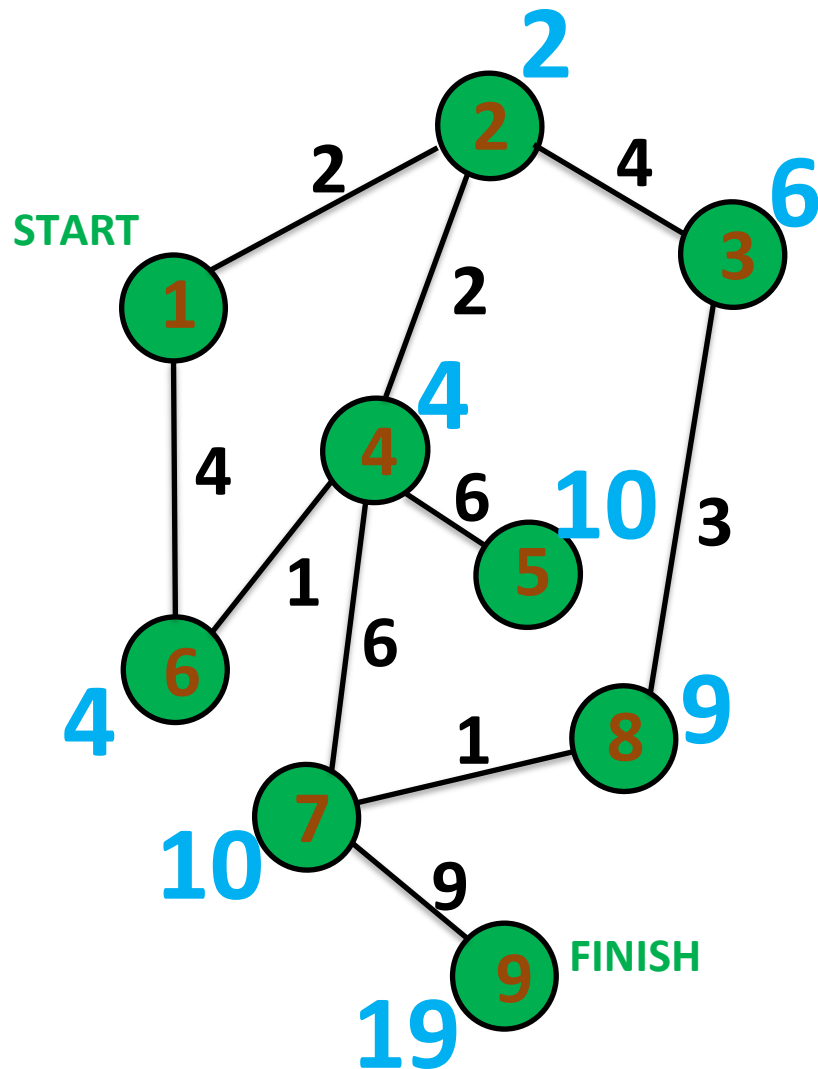


## Алгоритм Дейкстры (11)



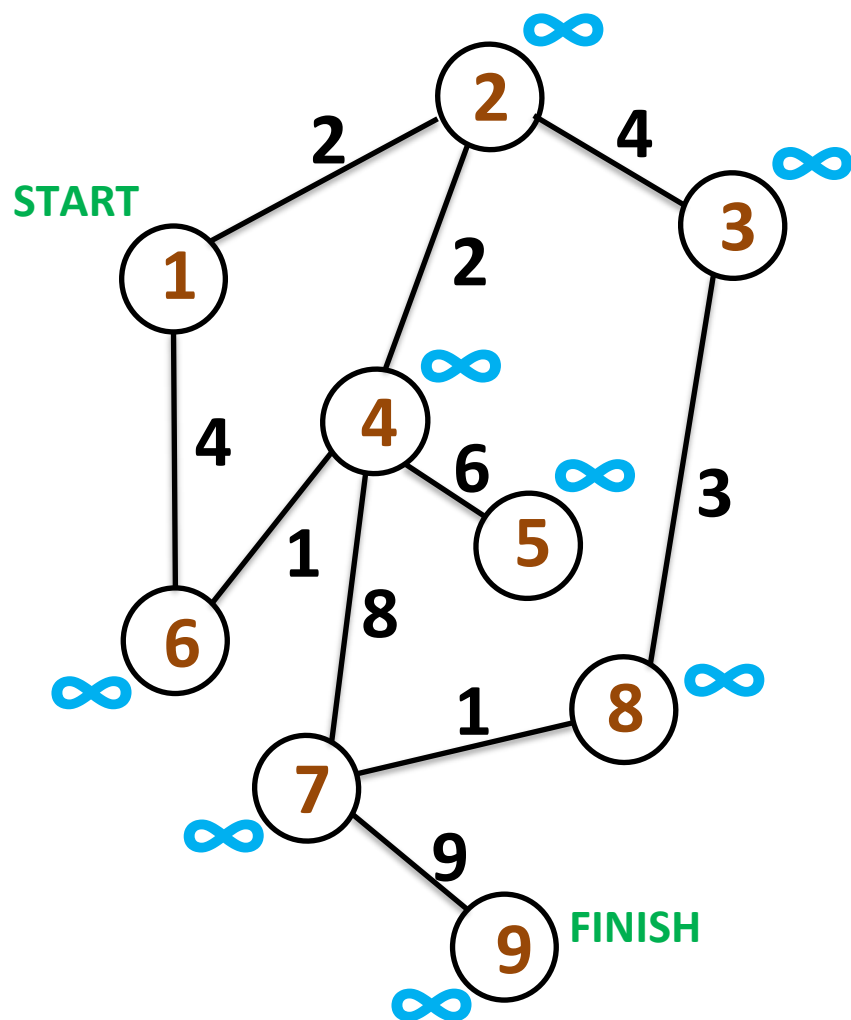


## Алгоритм Дейкстры (12)

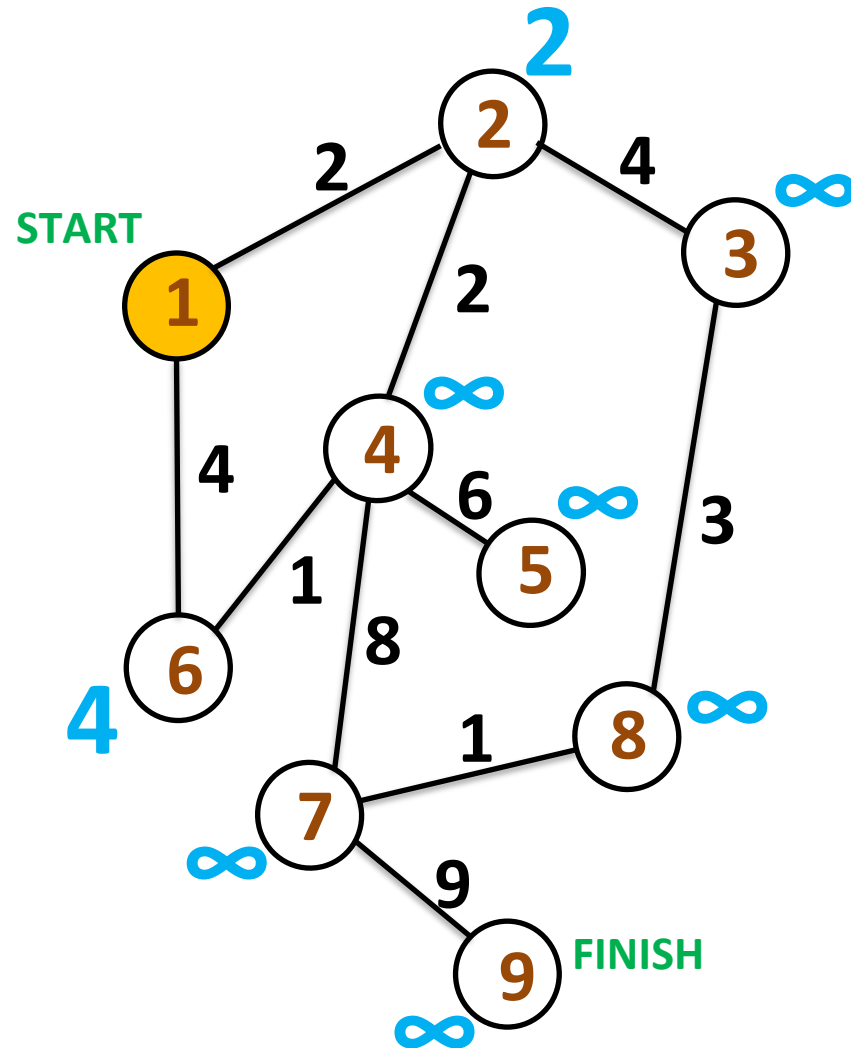


Путь из точки **START**  
в точку **FINISH** существует,  
самый короткий путь  
занимает 19 единиц.

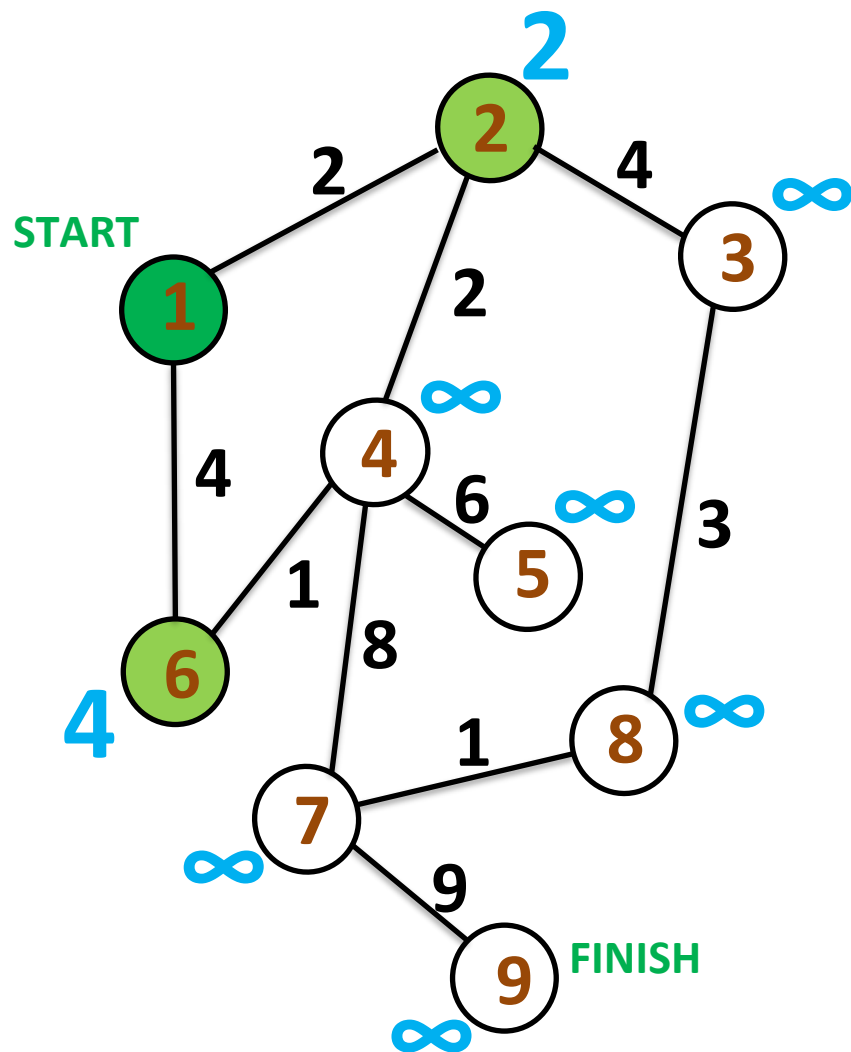
## Алгоритм Дейкстры (1)



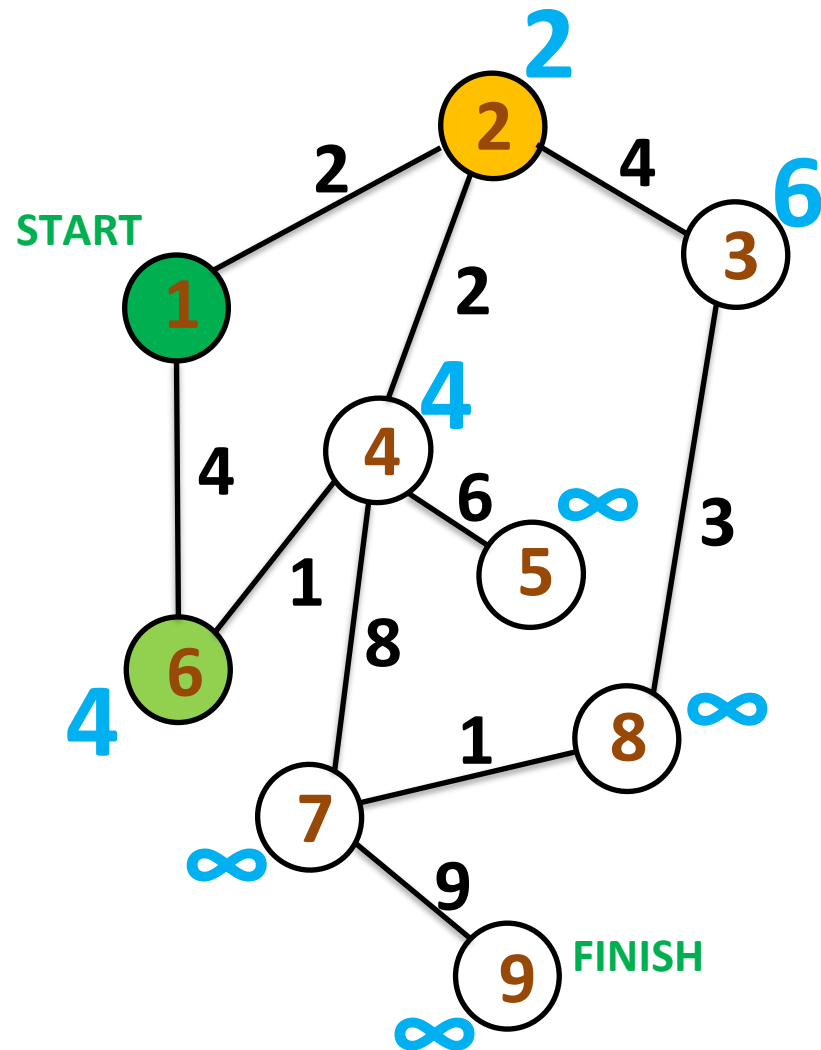
## Алгоритм Дейкстры (2)



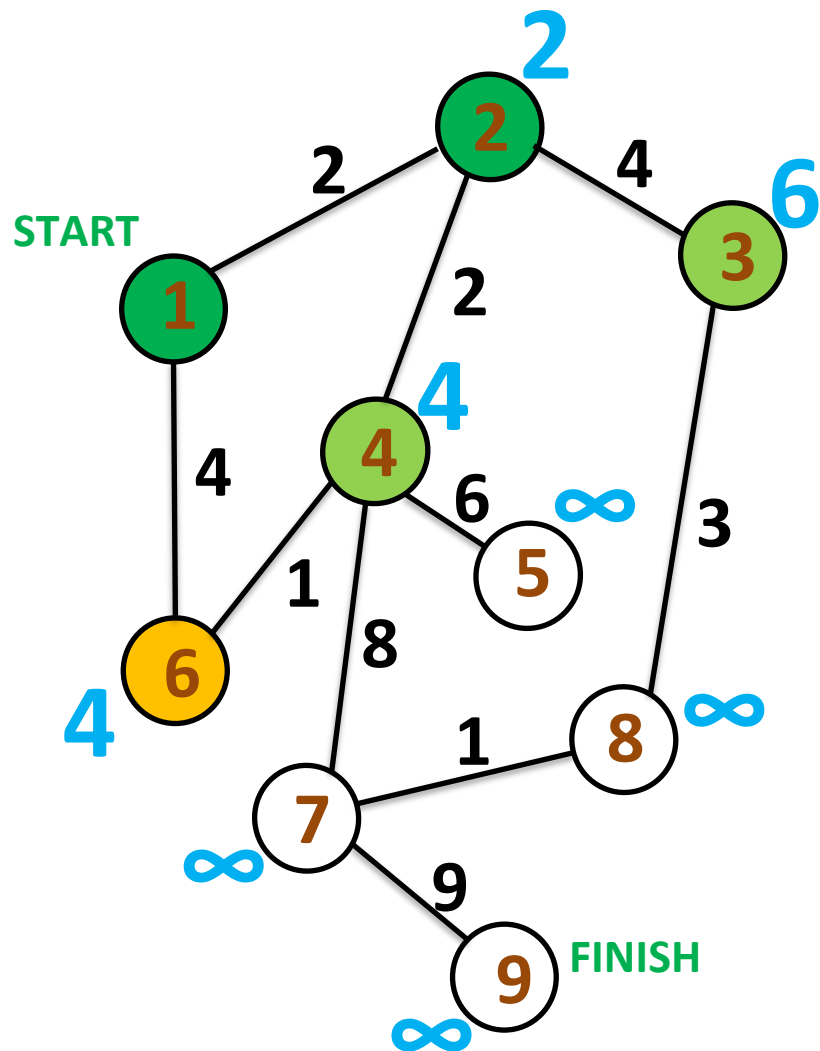
## Алгоритм Дейкстры (3)



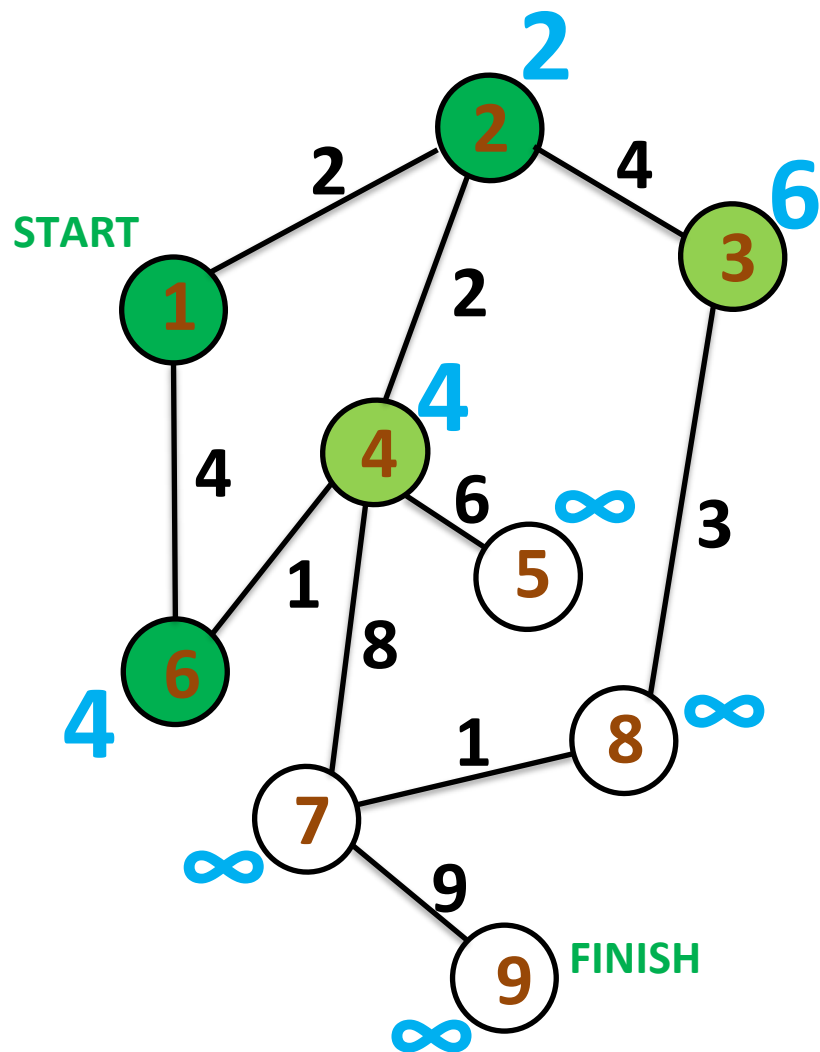
## Алгоритм Дейкстры (4)



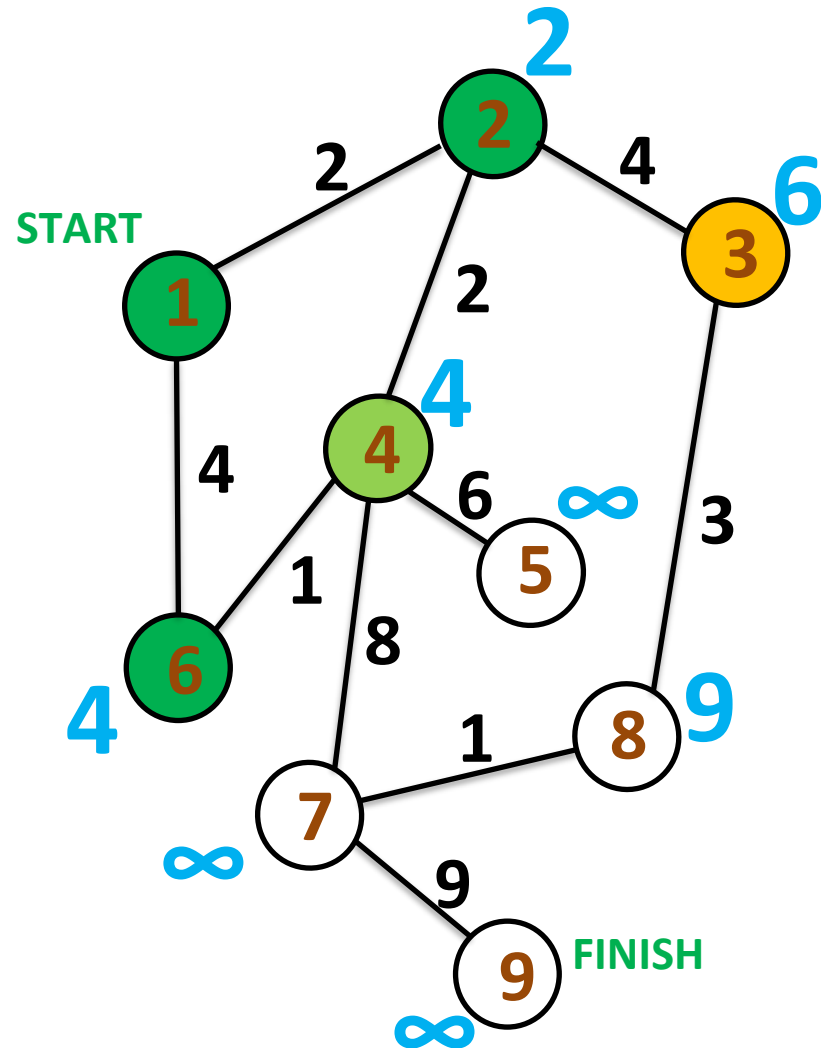
## Алгоритм Дейкстры (5)



## Алгоритм Дейкстры (6)

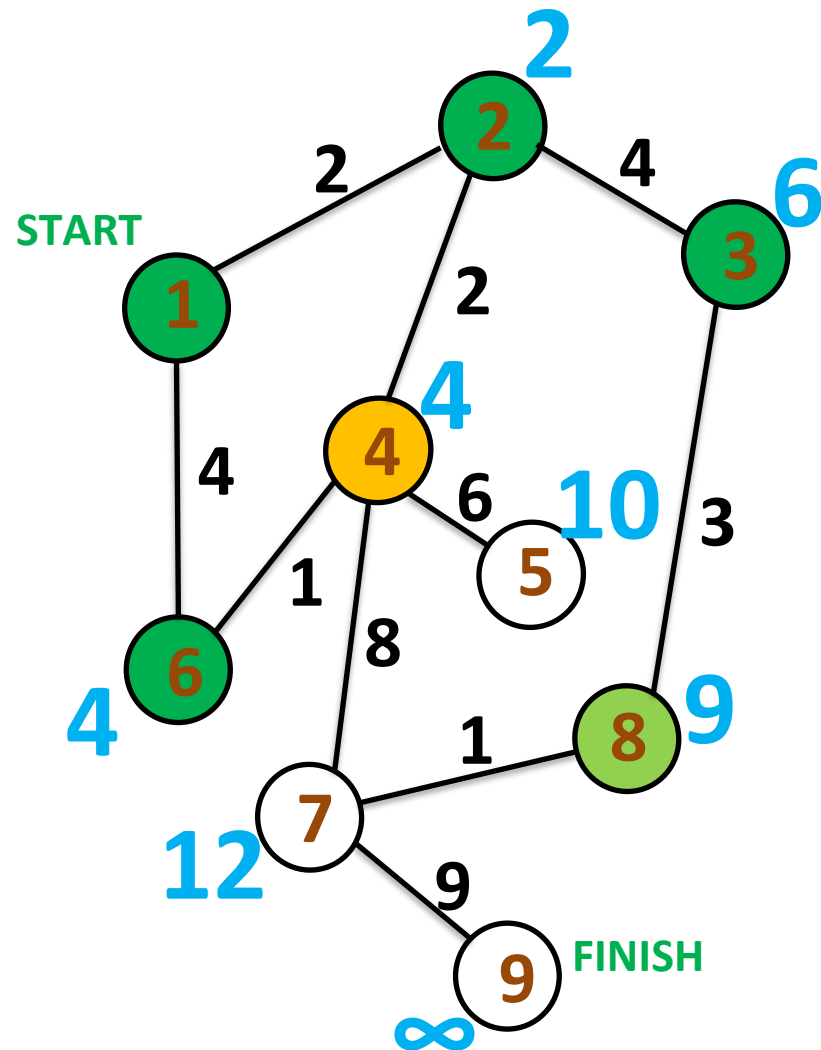


## Алгоритм Дейкстры (7)

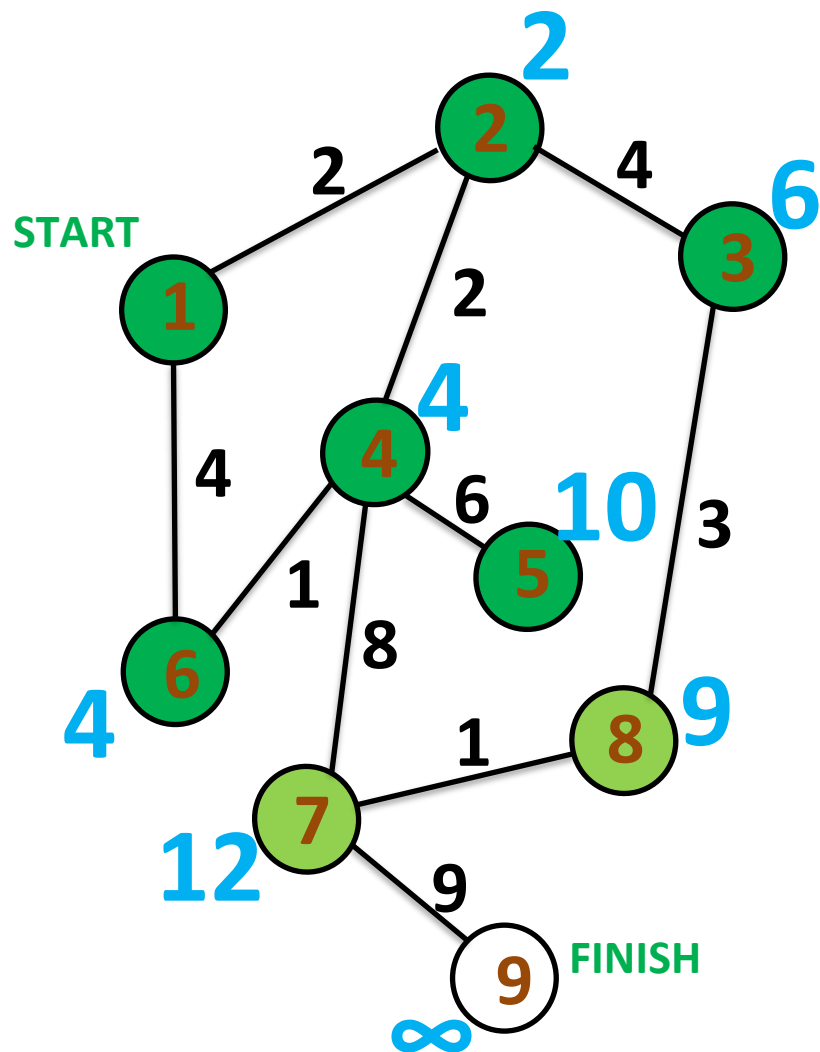




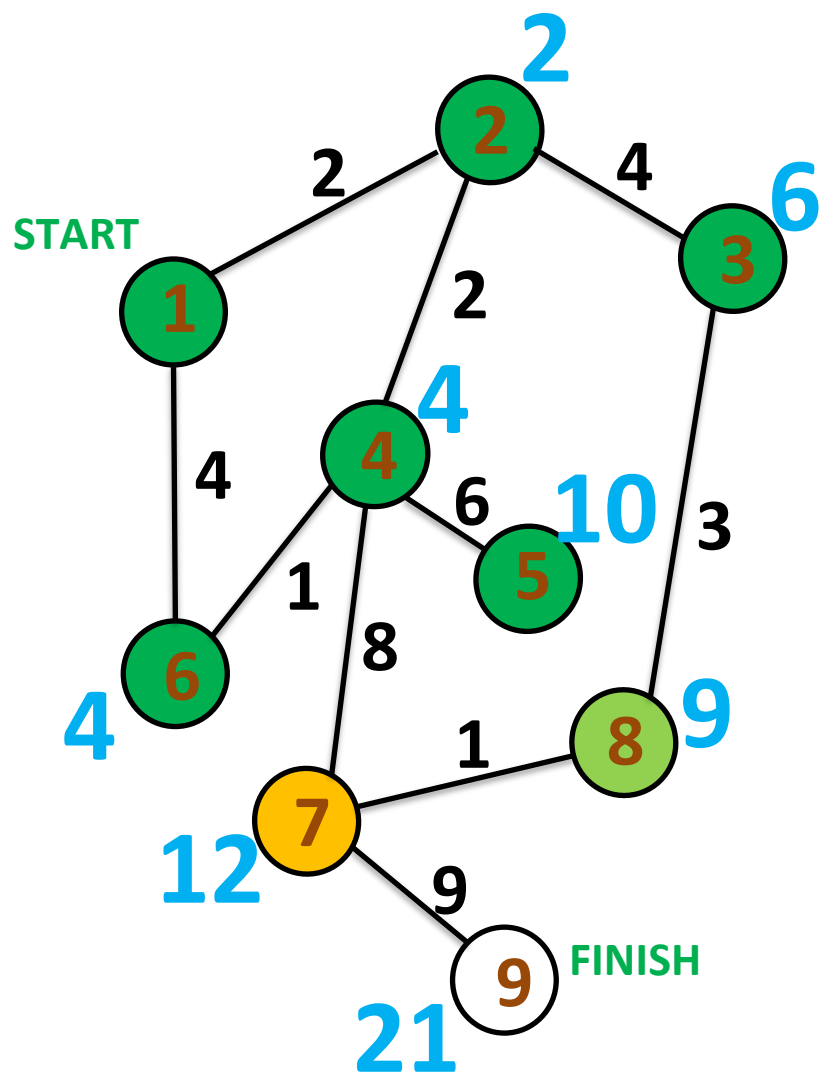
## Алгоритм Дейкстры (8)



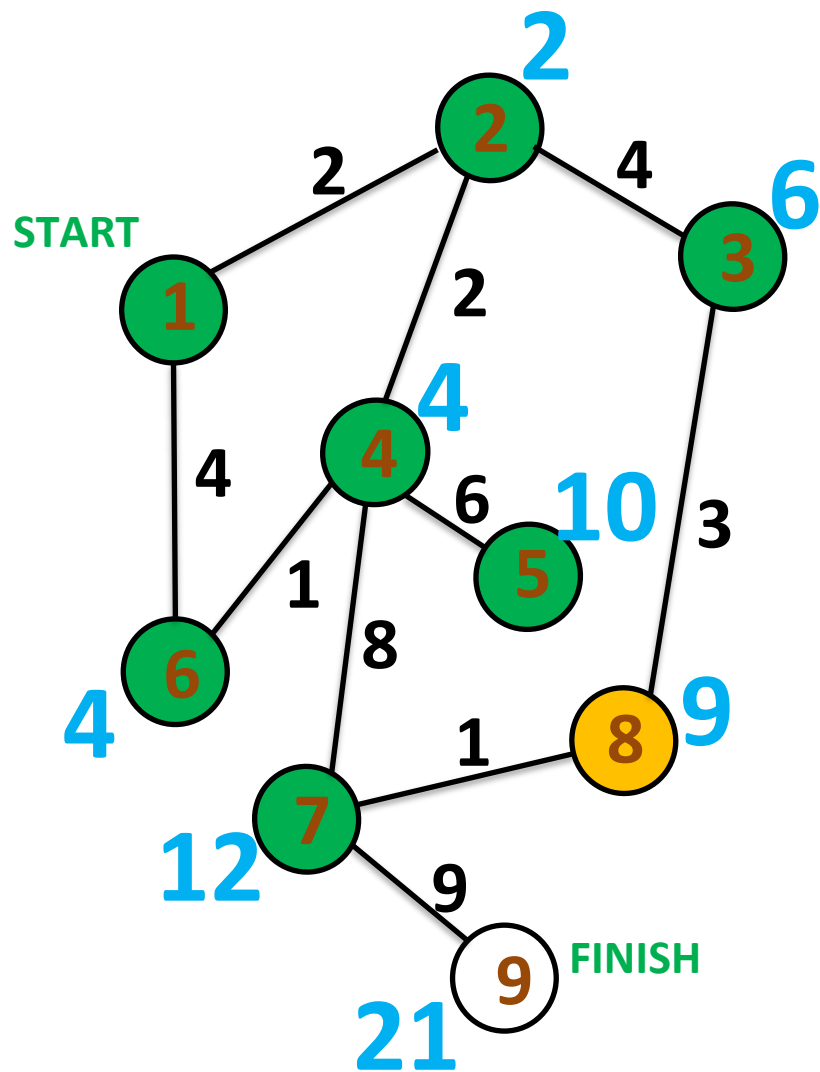
## Алгоритм Дейкстры (9)



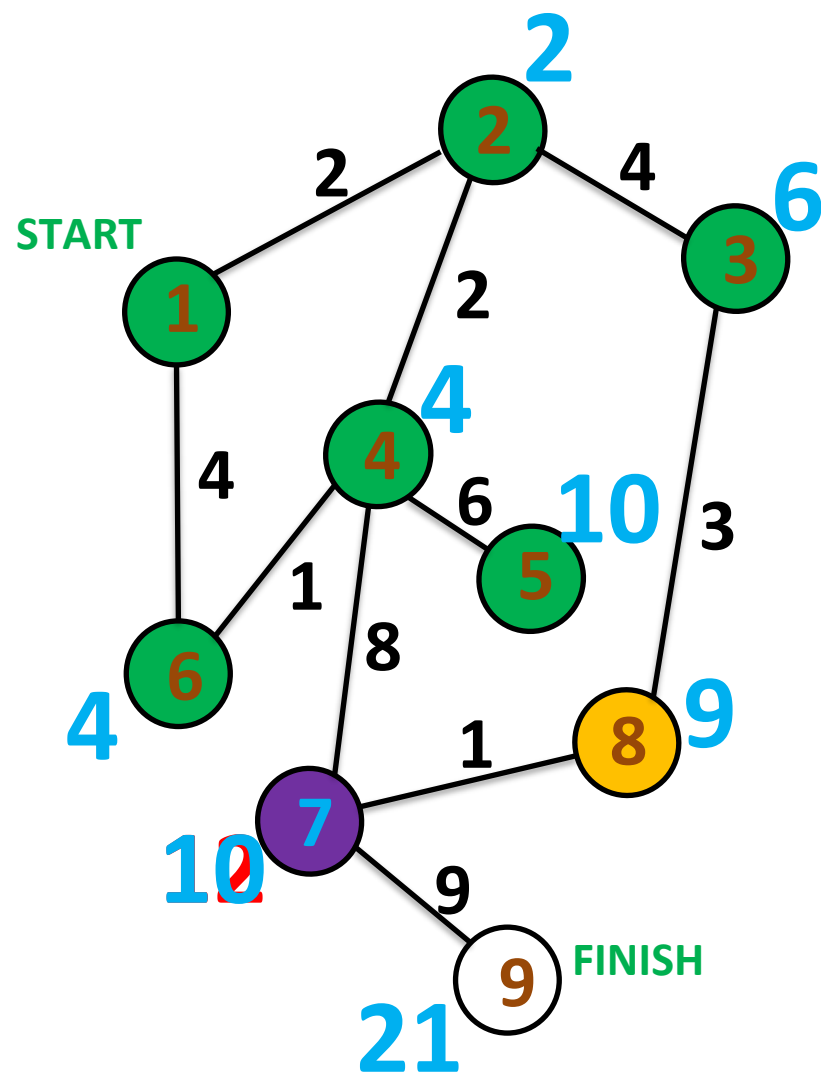
# Алгоритм Дейкстры (10)



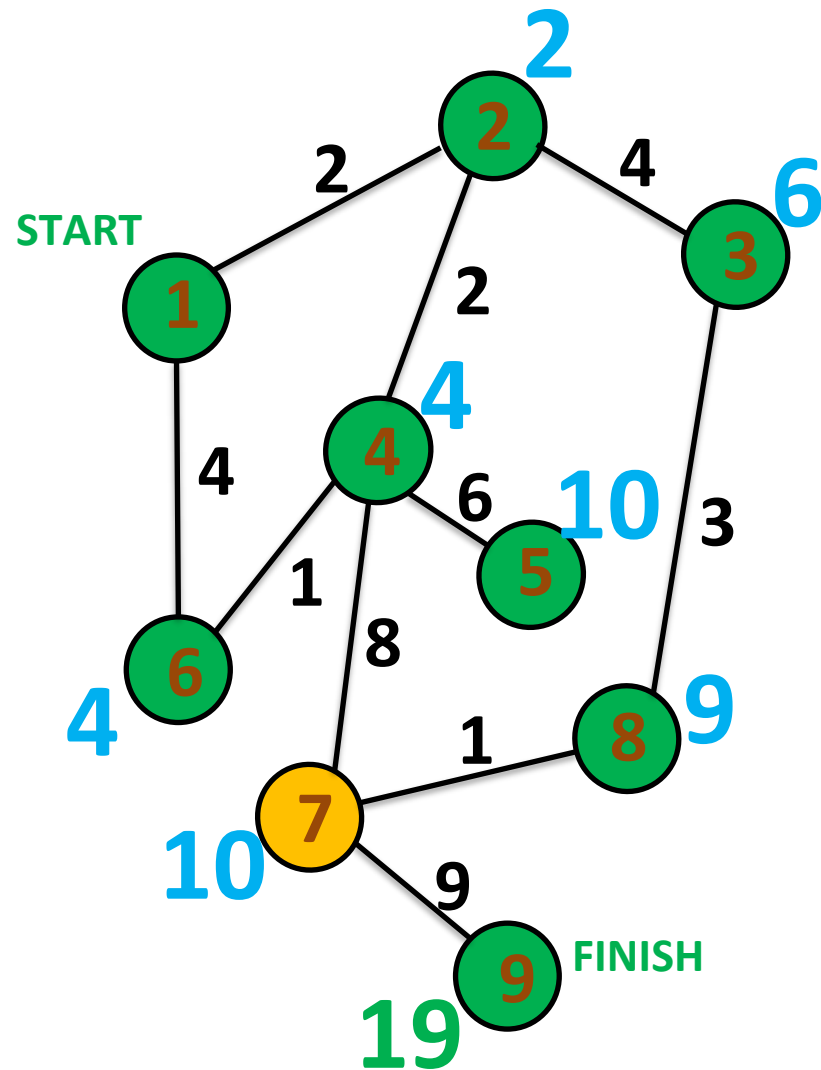
# Алгоритм Дейкстры (11)



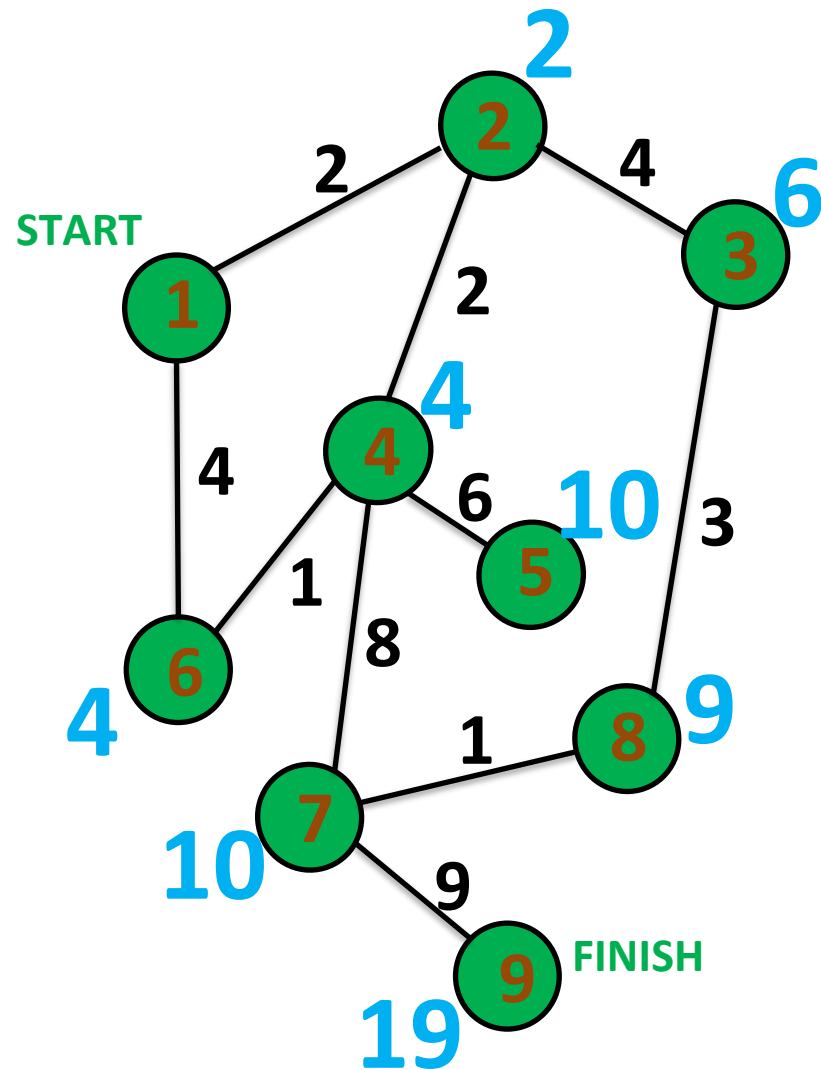
## Алгоритм Дейкстры (12)



## Алгоритм Дейкстры (13)



## Алгоритм Дейкстры (14)



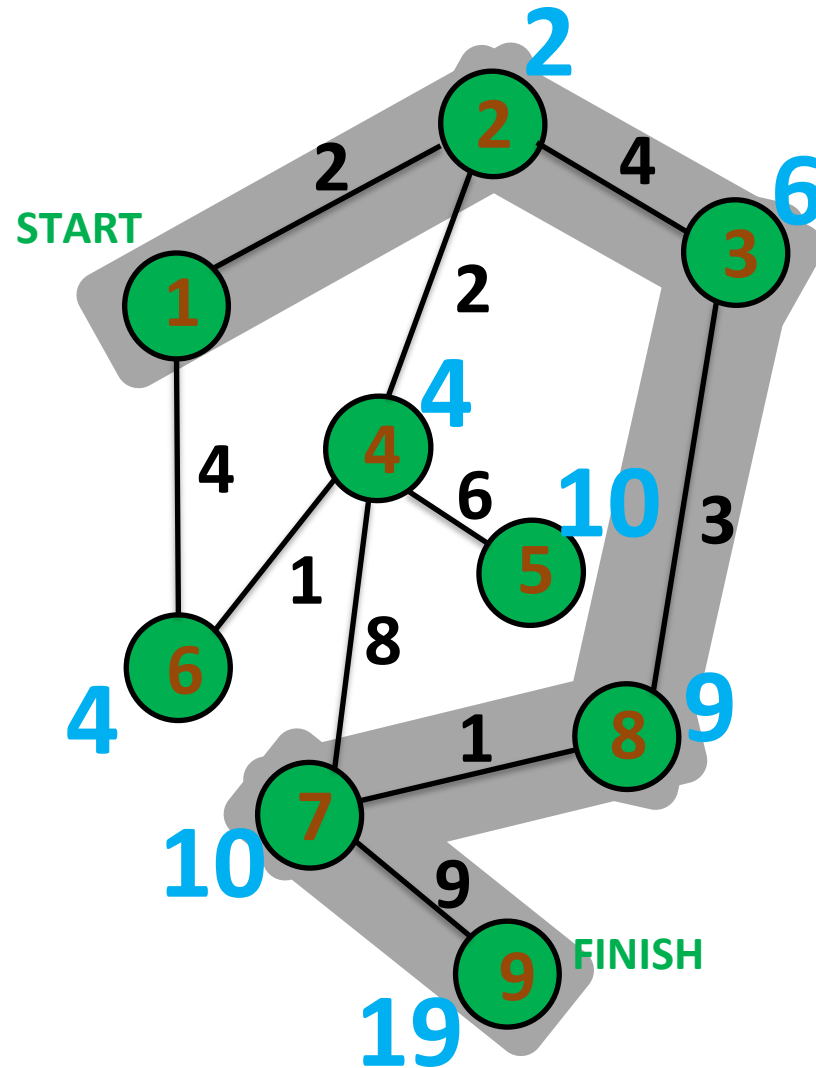
Путь из точки **START**  
в точку **FINISH** существует,  
самый короткий путь  
занимает 19 единиц.

## Алгоритм Дейкстры: поиск обратного пути (1)

1. каждый узел хранит указатель на вершину, от которой мы пришли к текущей вершине;
2. каждый узел хранит номер (индекс) вершины, от которой мы пришли к текущей вершине;
3. каждый узел хранит массив вершин, по которым мы прошли по пути до этой вершины.

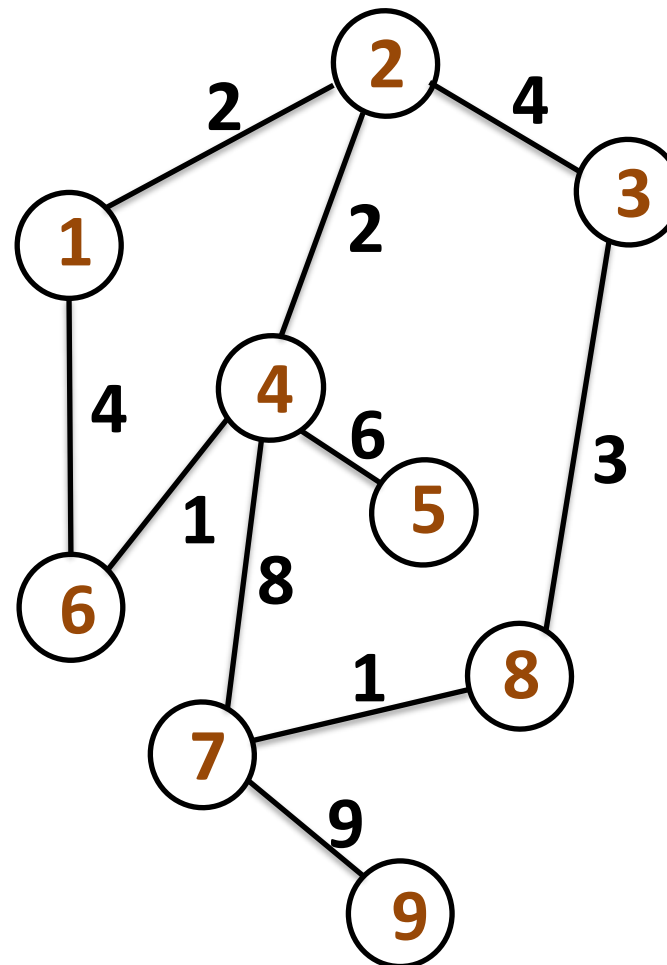


## Алгоритм Дейкстры: поиск обратного пути (2)



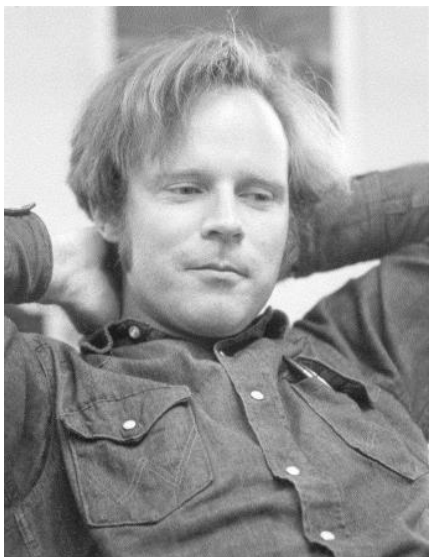
## Алгоритм Дейкстры: обход графа «в глубину»

Эдсгер Вибе Дейкстра

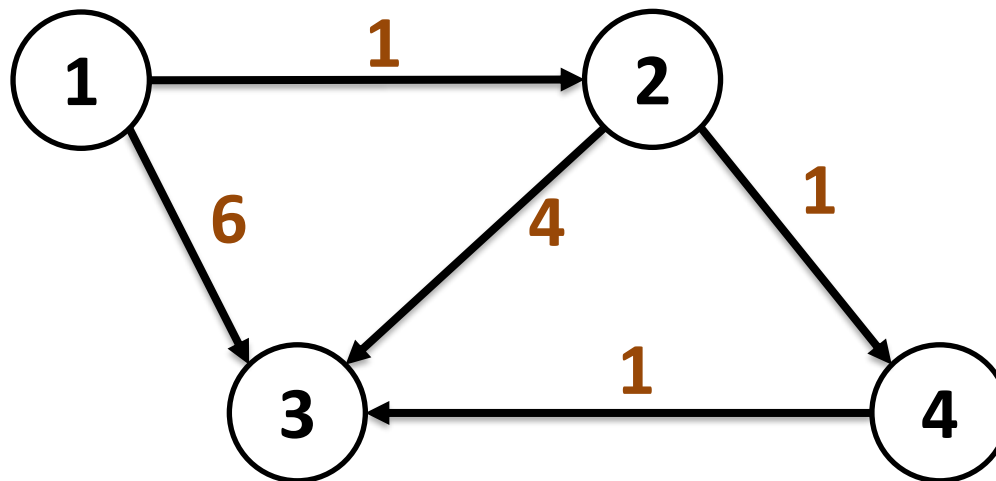


Реализация «в глубину»  
традиционно отличается  
наличием рекурсии

## Алгоритм Флойда: обход орграфа «в ширину» (1)



Роберт Флойд



Шаг 1: заполняем весами  
матрицу смежности

|   | 1        | 2        | 3 | 4        |
|---|----------|----------|---|----------|
| 1 |          | 1        | 6 | $\infty$ |
| 2 | $\infty$ |          | 4 | 1        |
| 3 | $\infty$ | $\infty$ |   | $\infty$ |
| 4 | $\infty$ | $\infty$ | 1 |          |

## Алгоритм Флойда: обход орграфа «в ширину» (2)

|   | 1        | 2        | 3 | 4        |
|---|----------|----------|---|----------|
| 1 |          | 1        | 6 | $\infty$ |
| 2 | $\infty$ |          | 4 | 1        |
| 3 | $\infty$ | $\infty$ |   | $\infty$ |
| 4 | $\infty$ | $\infty$ | 1 |          |

Шаг 2: запускается итерационный процесс по  $k$  от 1 до  $N$ , на каждой итерации которого элементы матрицы обновляются согласно формуле:

$$A_k[i, j] = \min(A_{k-1}[i, j], A_{k-1}[i, k] + A_{k-1}[k, j])$$

## Алгоритм Флойда: пример (1)

| $A_0$ |          | 1        | 2 | 3        | 4 |
|-------|----------|----------|---|----------|---|
| 1     | 0        | 1        | 6 | $\infty$ |   |
| 2     | $\infty$ | 0        | 4 | 1        |   |
| 3     | $\infty$ | $\infty$ | 0 | $\infty$ |   |
| 4     | $\infty$ | $\infty$ | 1 | 0        |   |

| $A_1$ |   | 1 | 2 | 3 | 4 |
|-------|---|---|---|---|---|
| 1     | 0 | 1 |   |   |   |
| 2     |   | 0 |   |   |   |
| 3     |   |   | 0 |   |   |
| 4     |   |   |   | 0 |   |

Diagram illustrating the first iteration of the Floyd-Warshall algorithm. The initial distance matrix  $A_0$  is shown on the left, and the updated matrix  $A_1$  is shown on the right. The iteration is for  $k=1$ , where the intermediate node is 1. The value  $A_1[1,2]$  is updated from  $\infty$  to 1, which is the minimum of the direct edge  $A_0[1,2]$  and the path through node 1 ( $A_0[1,1] + A_0[1,2]$ ). Arrows indicate the update of  $A_1[1,2]$  from  $A_0[1,2]$  and  $A_0[1,1] + A_0[1,2]$ . Brackets below the matrices show the subproblems used in the calculation.

Первая итерация,  $k = 1$ :

$$A_1[i, j] = \min(A_0[i, j], A_0[i, 1] + A_0[1, j])$$

$$\min(1, 0 + 1)$$

## Алгоритм Флойда: пример (2)

| $A_0$          |          | 1        | 2 | 3 <sup>j</sup> | 4 |
|----------------|----------|----------|---|----------------|---|
| 1 <sup>i</sup> | 0        | 1        | 6 | $\infty$       |   |
| 2              | $\infty$ | 0        | 4 | 1              |   |
| 3              | $\infty$ | $\infty$ | 0 | $\infty$       |   |
| 4              | $\infty$ | $\infty$ | 1 | 0              |   |

| $A_1$ |   | 1 | 2 | 3 | 4 |
|-------|---|---|---|---|---|
| 1     | 0 | 1 | 6 |   |   |
| 2     |   | 0 |   |   |   |
| 3     |   |   | 0 |   |   |
| 4     |   |   |   | 0 |   |

Первая итерация,  $k = 1$ :

$$A_1[i, j] = \min(A_0[i, j], A_0[i, 1] + A_0[1, j])$$

$$\min(6, 0 + 6)$$

## Алгоритм Флойда: пример (3)

| $A_0$          |          | 1        | 2 | 3        | 4 <sup>j</sup> |
|----------------|----------|----------|---|----------|----------------|
| 1 <sup>i</sup> | 0        | 1        | 6 | $\infty$ |                |
| 2              | $\infty$ | 0        | 4 | 1        |                |
| 3              | $\infty$ | $\infty$ | 0 | $\infty$ |                |
| 4              | $\infty$ | $\infty$ | 1 | 0        |                |

| $A_1$ |   | 1 | 2 | 3        | 4 |
|-------|---|---|---|----------|---|
| 1     | 0 | 1 | 6 | $\infty$ |   |
| 2     |   | 0 |   |          |   |
| 3     |   |   | 0 |          |   |
| 4     |   |   |   | 0        |   |

Первая итерация,  $k = 1$ :

$$A_1[i, j] = \min(A_0[i, j], A_0[i, 1] + A_0[1, j])$$

$$\min(\infty, 0 + \infty)$$

## Алгоритм Флойда: пример (4)

|   |   | $A_0$    |          |   |          | $A_1$    |   |   |          |
|---|---|----------|----------|---|----------|----------|---|---|----------|
|   |   | 1        | 2        | 3 | 4        | 1        | 2 | 3 | 4        |
| i | 1 | 0        | 1        | 6 | $\infty$ | 0        | 1 | 6 | $\infty$ |
|   | 2 | $\infty$ | 0        | 4 | 1        | $\infty$ | 0 |   |          |
|   | 3 | $\infty$ | $\infty$ | 0 | $\infty$ |          |   | 0 |          |
|   | 4 | $\infty$ | $\infty$ | 1 | 0        |          |   |   | 0        |

Diagram illustrating the Floyd-Warshall algorithm. The matrix  $A_0$  shows the initial distances between nodes 1, 2, 3, and 4. The matrix  $A_1$  shows the result after the first iteration (k=1), where the distance from node 2 to node 1 is updated to  $\infty$  because the path through node 1 is not shorter than the current distance.

Первая итерация,  $k = 1$ :

$$A_1[i, j] = \min(A_0[i, j], A_0[i, 1] + A_0[1, j])$$

$$\min(\infty, \infty + 0)$$



## Алгоритм Флойда: пример (5)

| $A_0$                   |          | 1        | 2 | $3$ <sup><i>j</i></sup> | 4 |
|-------------------------|----------|----------|---|-------------------------|---|
| 1                       | 0        | 1        | 6 | $\infty$                |   |
| $2$ <sup><i>i</i></sup> | $\infty$ | 0        | 4 | 1                       |   |
| 3                       | $\infty$ | $\infty$ | 0 | $\infty$                |   |
| 4                       | $\infty$ | $\infty$ | 1 | 0                       |   |

| $A_1$ |          | 1 | 2 | 3        | 4 |
|-------|----------|---|---|----------|---|
| 1     | 0        | 1 | 6 | $\infty$ |   |
| 2     | $\infty$ | 0 | 4 |          |   |
| 3     |          |   | 0 |          |   |
| 4     |          |   |   | 0        |   |

Первая итерация,  $k = 1$ :

$$A_1[i, j] = \min(A_0[i, j], A_0[i, 1] + A_0[1, j])$$

$$\min(4, \infty + 6)$$

## Алгоритм Флойда: пример (6)

Результат прохода по матрице  $A$  при  $k = 1$

 $A_0$ 

|   | 1        | 2        | 3 | 4        |
|---|----------|----------|---|----------|
| 1 | 0        | 1        | 6 | $\infty$ |
| 2 | $\infty$ | 0        | 4 | 1        |
| 3 | $\infty$ | $\infty$ | 0 | $\infty$ |
| 4 | $\infty$ | $\infty$ | 1 | 0        |

 $A_1$ 

|   | 1        | 2        | 3 | 4        |
|---|----------|----------|---|----------|
| 1 | 0        | 1        | 6 | $\infty$ |
| 2 | $\infty$ | 0        | 4 | 1        |
| 3 | $\infty$ | $\infty$ | 0 | $\infty$ |
| 4 | $\infty$ | $\infty$ | 1 | 0        |

$$A_1[i, j] = \min(A_0[i, j], A_0[i, 1] + A_0[1, j])$$

## Алгоритм Флойда: пример (7)

| $A_1$ |          | 1        | 2 | 3        | 4 |
|-------|----------|----------|---|----------|---|
| 1     | 0        | 1        | 6 | $\infty$ |   |
| 2     | $\infty$ | 0        | 4 | 1        |   |
| 3     | $\infty$ | $\infty$ | 0 | $\infty$ |   |
| 4     | $\infty$ | $\infty$ | 1 | 0        |   |

| $A_2$ |   | 1 | 2 | 3 | 4 |
|-------|---|---|---|---|---|
| 1     | 0 | 1 |   |   |   |
| 2     |   | 0 |   |   |   |
| 3     |   |   | 0 |   |   |
| 4     |   |   |   | 0 |   |

*Note: In the original image, the cell A1[1,2] is circled in green, and the cell A2[1,2] is circled in blue. Arrows indicate the update of A2[1,2] from A1[1,2] (1) to min(A1[1,2], A1[1,2]+A1[2,2]) = 1.*

Вторая итерация,  $k = 2$ :

$$A_2[i, j] = \min(A_1[i, j], A_1[i, 2] + A_1[2, j])$$

$$\min(1, 0 + \infty)$$

## Алгоритм Флойда: пример (8)

| $A_1$          |          | 1        | 2 | 3 <sup>j</sup> | 4 |
|----------------|----------|----------|---|----------------|---|
| 1 <sup>i</sup> | 0        | 1        | 6 | $\infty$       |   |
| 2              | $\infty$ | 0        | 4 | 1              |   |
| 3              | $\infty$ | $\infty$ | 0 | $\infty$       |   |
| 4              | $\infty$ | $\infty$ | 1 | 0              |   |

| $A_2$ |   | 1 | 2 | 3 | 4 |
|-------|---|---|---|---|---|
| 1     | 0 | 1 | 5 |   |   |
| 2     |   | 0 |   |   |   |
| 3     |   |   | 0 |   |   |
| 4     |   |   |   | 0 |   |

Первая итерация,  $k = 2$ :

$$A_2[i, j] = \min(A_1[i, j], A_1[i, 2] + A_1[2, j])$$

$$\min(6, 1 + 4)$$

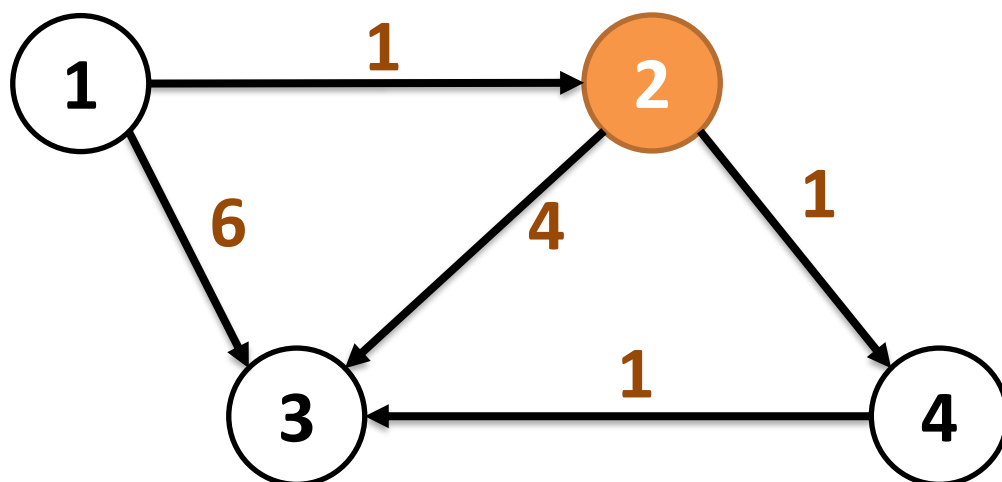
## Алгоритм Флойда: пример (9)

$A_1$

|   | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | 0 |   | 6 |   |
| 2 |   | 0 |   |   |
| 3 |   |   | 0 |   |
| 4 |   |   |   | 0 |

$A_2$

|   | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | 0 |   | 5 |   |
| 2 |   | 0 |   |   |
| 3 |   |   | 0 |   |
| 4 |   |   |   | 0 |



## Алгоритм Флойда: пример (10)

| $A_1$          |          | 1        | 2 | 3        | 4 <sup>j</sup> |
|----------------|----------|----------|---|----------|----------------|
| 1 <sup>i</sup> | 0        | 1        | 6 | $\infty$ |                |
| 2              | $\infty$ | 0        | 4 | 1        |                |
| 3              | $\infty$ | $\infty$ | 0 | $\infty$ |                |
| 4              | $\infty$ | $\infty$ | 1 | 0        |                |

| $A_2$ |   | 1 | 2 | 3 | 4 |
|-------|---|---|---|---|---|
| 1     | 0 | 1 | 5 | 2 |   |
| 2     |   | 0 |   |   |   |
| 3     |   |   | 0 |   |   |
| 4     |   |   |   | 0 |   |

Первая итерация,  $k = 2$ :

$$A_2[i, j] = \min(A_1[i, j], A_1[i, 2] + A_1[2, j])$$

$$\min(\infty, 1 + 1)$$

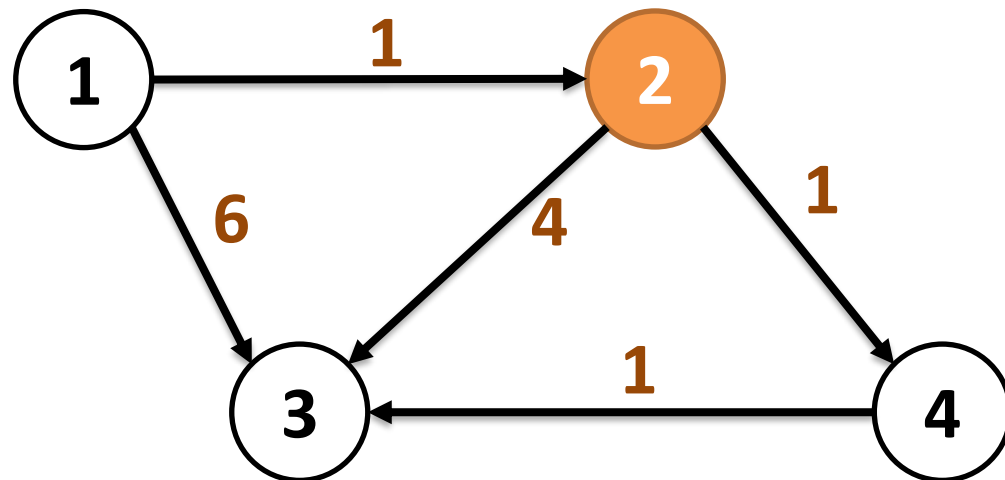
## Алгоритм Флойда: пример (11)

$A_1$

|   | 1 | 2 | 3 | 4        |
|---|---|---|---|----------|
| 1 | 0 |   |   | $\infty$ |
| 2 |   | 0 |   |          |
| 3 |   |   | 0 |          |
| 4 |   |   |   | 0        |

$A_2$

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



## Алгоритм Флойда: пример (12)

Результат прохода по матрице A при  $k = 2$

 $A_1$ 

|   | 1        | 2        | 3 | 4        |
|---|----------|----------|---|----------|
| 1 | 0        | 1        | 6 | $\infty$ |
| 2 | $\infty$ | 0        | 4 | 1        |
| 3 | $\infty$ | $\infty$ | 0 | $\infty$ |
| 4 | $\infty$ | $\infty$ | 1 | 0        |

 $A_2$ 

|   | 1        | 2        | 3 | 4        |
|---|----------|----------|---|----------|
| 1 | 0        | 1        | 5 | 2        |
| 2 | $\infty$ | 0        | 4 | 1        |
| 3 | $\infty$ | $\infty$ | 0 | $\infty$ |
| 4 | $\infty$ | $\infty$ | 1 | 0        |

$$A_2[i, j] = \min(A_1[i, j], A_1[i, 2] + A_1[2, j])$$



## Алгоритм Флойда: пример (13)

Результат прохода по матрице A при  $k = 3$

 $A_2$ 

|   | 1        | 2        | 3 | 4        |
|---|----------|----------|---|----------|
| 1 | 0        | 1        | 5 | 2        |
| 2 | $\infty$ | 0        | 4 | 1        |
| 3 | $\infty$ | $\infty$ | 0 | $\infty$ |
| 4 | $\infty$ | $\infty$ | 1 | 0        |

 $A_3$ 

|   | 1        | 2        | 3 | 4        |
|---|----------|----------|---|----------|
| 1 | 0        | 1        | 5 | 2        |
| 2 | $\infty$ | 0        | 4 | 1        |
| 3 | $\infty$ | $\infty$ | 0 | $\infty$ |
| 4 | $\infty$ | $\infty$ | 1 | 0        |

$$A_3[i, j] = \min(A_2[i, j], A_2[i, 3] + A_2[3, j])$$

## Алгоритм Флойда: пример (14)

| $A_3$ |          | 1        | 2 | 3        | 4 |
|-------|----------|----------|---|----------|---|
| 1     | 0        | 1        | 5 | 2        |   |
| 2     | $\infty$ | 0        | 4 | 1        |   |
| 3     | $\infty$ | $\infty$ | 0 | $\infty$ |   |
| 4     | $\infty$ | $\infty$ | 1 | 0        |   |

| $A_4$ |   | 1 | 2 | 3 | 4 |
|-------|---|---|---|---|---|
| 1     | 0 | 1 |   |   |   |
| 2     |   | 0 |   |   |   |
| 3     |   |   | 0 |   |   |
| 4     |   |   |   | 0 |   |

Diagram illustrating the Floyd-Warshall algorithm step  $k=4$ . The matrix  $A_3$  shows the shortest paths between nodes 1, 2, 3, and 4 using only nodes 1, 2, 3, and 4 as intermediate nodes. The matrix  $A_4$  shows the shortest paths between nodes 1, 2, 3, and 4 using only node 4 as an intermediate node. The value  $A_4[1,2]$  is updated from  $\infty$  to 1, which is the minimum of  $A_3[1,2]$  and  $A_3[1,4] + A_3[4,2]$ .

Первая итерация,  $k = 4$ :

$$A_4[i, j] = \min(A_3[i, j], A_3[i, 4] + A_3[4, j])$$

$$\min(1, \infty + \infty)$$

## Алгоритм Флойда: пример (15)

 $A_3$ 

|                | 1        | 2        | 3 <sup>j</sup> | 4        |
|----------------|----------|----------|----------------|----------|
| 1 <sup>i</sup> | 0        | 1        | 5              | 2        |
| 2              | $\infty$ | 0        | 4              | 1        |
| 3              | $\infty$ | $\infty$ | 0              | $\infty$ |
| 4              | $\infty$ | $\infty$ | 1              | 0        |

 $A_4$ 

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

Первая итерация,  $k = 4$ :

$$A_4[i, j] = \min(A_3[i, j], A_3[i, 4] + A_3[4, j])$$

$$\min(5, 2 + 1)$$

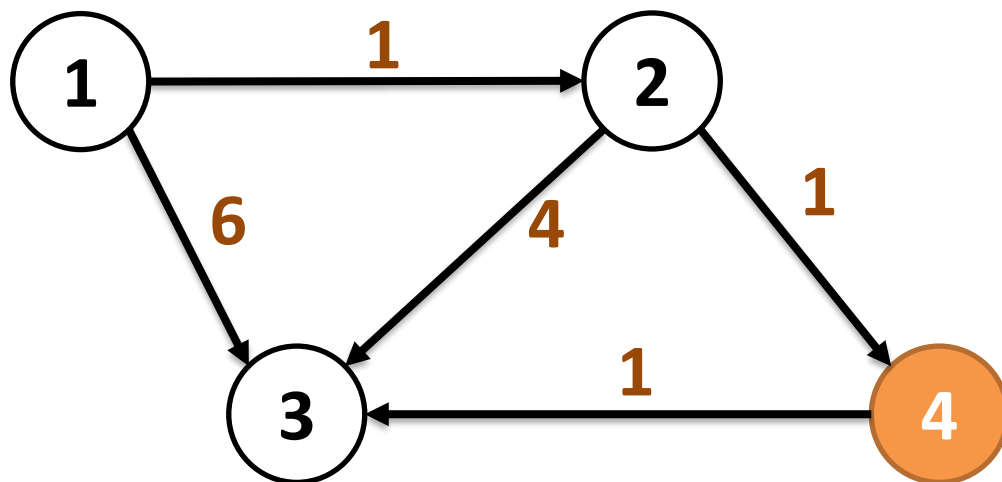
## Алгоритм Флойда: пример (16)

 $A_1$ 

|   | 1        | 2        | 3 | 4        |
|---|----------|----------|---|----------|
| 1 | 0        | 1        | 5 | 2        |
| 2 | $\infty$ | 0        | 4 | 1        |
| 3 | $\infty$ | $\infty$ | 0 | $\infty$ |
| 4 | $\infty$ | $\infty$ | 1 | 0        |

 $A_2$ 

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



## Алгоритм Флойда: пример (17)

|                | 1        | 2        | 3 | 4 <sup>j</sup> |
|----------------|----------|----------|---|----------------|
| 1 <sup>i</sup> | 0        | 1        | 5 | 2              |
| 2              | $\infty$ | 0        | 4 | 1              |
| 3              | $\infty$ | $\infty$ | 0 | $\infty$       |
| 4              | $\infty$ | $\infty$ | 1 | 0              |

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

Первая итерация,  $k = 4$ :

$$A_4[i, j] = \min(A_3[i, j], A_3[i, 4] + A_3[4, j])$$

$$\min(2, 2 + 0)$$

## Алгоритм Флойда: пример (18)

 $A_3$ 

|          | $j$<br>1 | 2        | 3 | 4        |
|----------|----------|----------|---|----------|
| 1        | 0        | 1        | 5 | 2        |
| $i$<br>2 | $\infty$ | 0        | 4 | 1        |
| 3        | $\infty$ | $\infty$ | 0 | $\infty$ |
| 4        | $\infty$ | $\infty$ | 1 | 0        |

 $A_4$ 

|   | 1        | 2        | 3 | 4        |
|---|----------|----------|---|----------|
| 1 | 0        | 1        | 3 | 2        |
| 2 | $\infty$ | 0        | 4 | 1        |
| 3 | $\infty$ | $\infty$ | 0 | $\infty$ |
| 4 | $\infty$ | $\infty$ | 1 | 0        |

Первая итерация,  $k = 4$ :

$$A_4[i, j] = \min(A_3[i, j], A_3[i, 4] + A_3[4, j])$$

$$\min(\infty, 1 + \infty)$$

## Алгоритм Флойда: пример (19)

 $A_3$ 

|                | 1        | 2        | 3 <sup>j</sup> | 4        |
|----------------|----------|----------|----------------|----------|
| 1              | 0        | 1        | 5              | 2        |
| 2 <sup>i</sup> | $\infty$ | 0        | 4              | 1        |
| 3              | $\infty$ | $\infty$ | 0              | $\infty$ |
| 4              | $\infty$ | $\infty$ | 1              | 0        |

 $A_4$ 

|   | 1        | 2        | 3 | 4        |
|---|----------|----------|---|----------|
| 1 | 0        | 1        | 3 | 2        |
| 2 | $\infty$ | 0        | 2 | 1        |
| 3 | $\infty$ | $\infty$ | 0 | $\infty$ |
| 4 | $\infty$ | $\infty$ | 1 | 0        |

Первая итерация,  $k = 4$ :

$$A_4[i, j] = \min(A_3[i, j], A_3[i, 4] + A_3[4, j])$$

$$\min(4, 1 + 1)$$

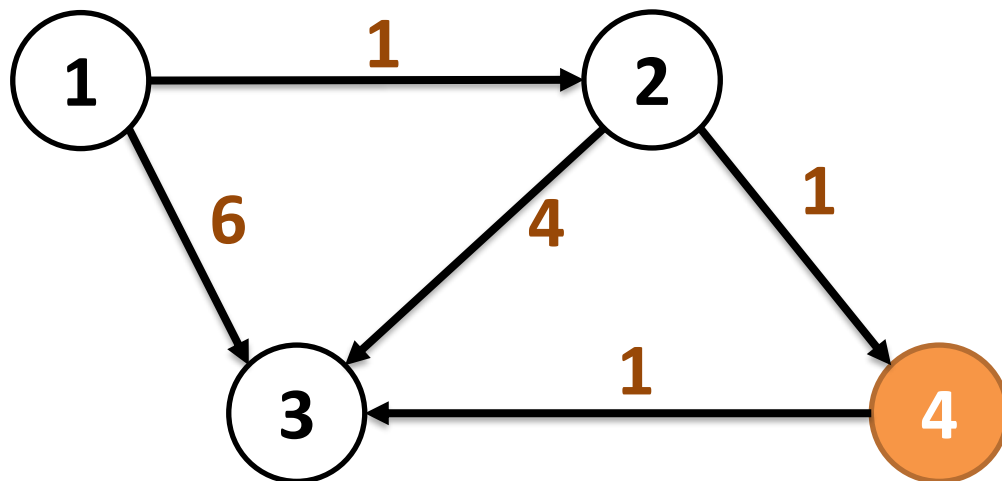
## Алгоритм Флойда: пример (20)

 $A_1$ 

|   | 1        | 2        | 3 | 4        |
|---|----------|----------|---|----------|
| 1 | 0        | 1        | 5 | 2        |
| 2 | $\infty$ | 0        | 4 | 1        |
| 3 | $\infty$ | $\infty$ | 0 | $\infty$ |
| 4 | $\infty$ | $\infty$ | 1 | 0        |

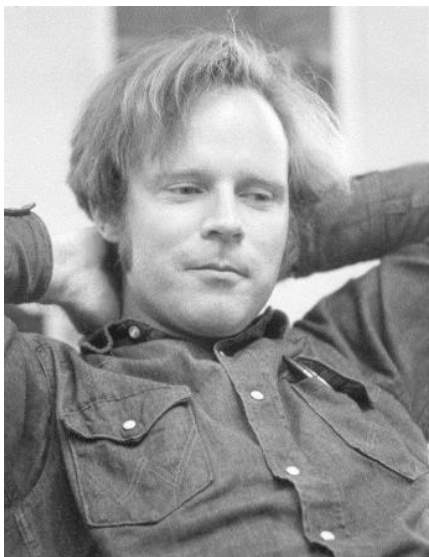
 $A_2$ 

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

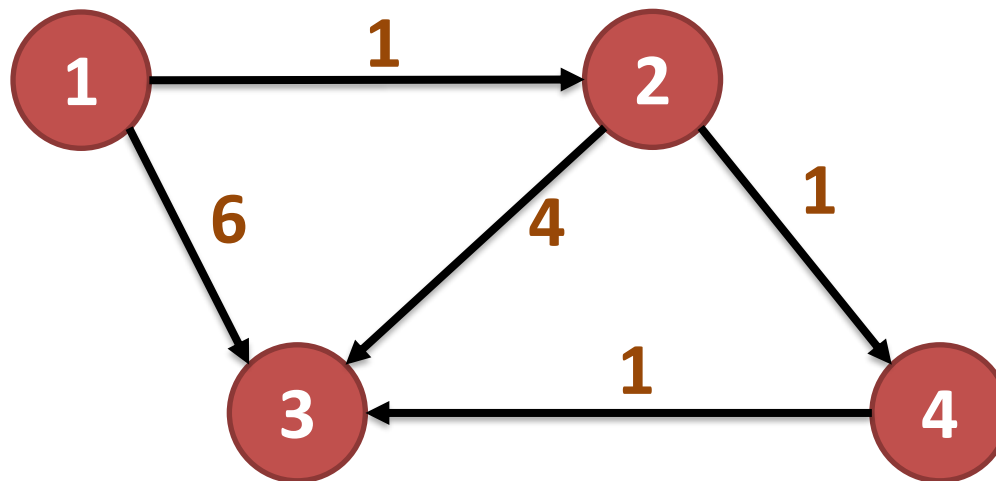




## Алгоритм Флойда: обход орграфа «в ширину»



Роберт Флойд



Особенность алгоритма:

после работы алгоритма есть информация о том, за сколько можно дойти **в любую** вершину из **любой** вершины.

|   | 1        | 2        | 3 | 4        |
|---|----------|----------|---|----------|
| 1 |          | 1        | 3 | 2        |
| 2 | $\infty$ |          | 2 | 1        |
| 3 | $\infty$ | $\infty$ |   | $\infty$ |
| 4 | $\infty$ | $\infty$ | 1 |          |

## Построение минимального остовного дерева (1)

Алгоритм Краскала позволяет построить минимальное остовное дерево для неориентированного взвешенного графа:

1. выписать таблицу рёбер с их весами;
2. отсортировать таблицу рёбер с их весами по возрастанию;
3. выбрать ребро минимального веса, добавить его к дереву;
4. выбрать минимальное ребро, не дающее цикла в дереве, добавлять его к дереву;
5. повторять пункт 4, пока не дошли до конца списка рёбер.

## Построение минимального остовного дерева (3)

Рёбра графа с весами:

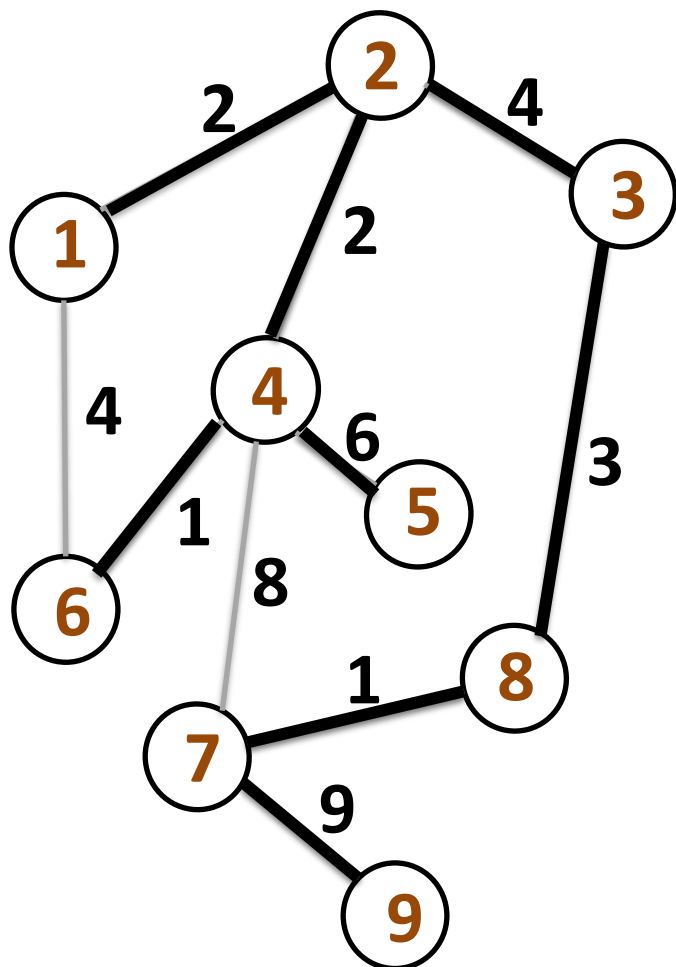
|     |     |     |     |     |     |     |     |     |     |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 1-2 | 1-6 | 2-4 | 4-6 | 2-3 | 4-5 | 3-8 | 4-7 | 7-8 | 7-9 |
| 2   | 4   | 2   | 1   | 4   | 6   | 3   | 8   | 1   | 9   |

Отсортированные рёбра графа по весам:

|     |     |     |     |     |     |     |     |     |     |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 4-6 | 7-8 | 1-2 | 2-4 | 3-8 | 1-6 | 2-3 | 4-5 | 4-7 | 7-9 |
| 1   | 1   | 2   | 2   | 3   | 4   | 4   | 6   | 8   | 9   |

Начинаем формировать дерево:

|     |     |     |     |     |     |     |     |     |     |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 4-6 | 7-8 | 1-2 | 2-4 | 3-8 | 1-6 | 2-3 | 4-5 | 4-7 | 7-9 |
| 1   | 1   | 2   | 2   | 3   | 4   | 4   | 6   | 8   | 9   |



## Декомпозиция: базовый алгоритм кластеризации

Критерии декомпозиции:

1. внутренняя связанность блоков должна быть максимальной
2. есть ограничение на размер блоков

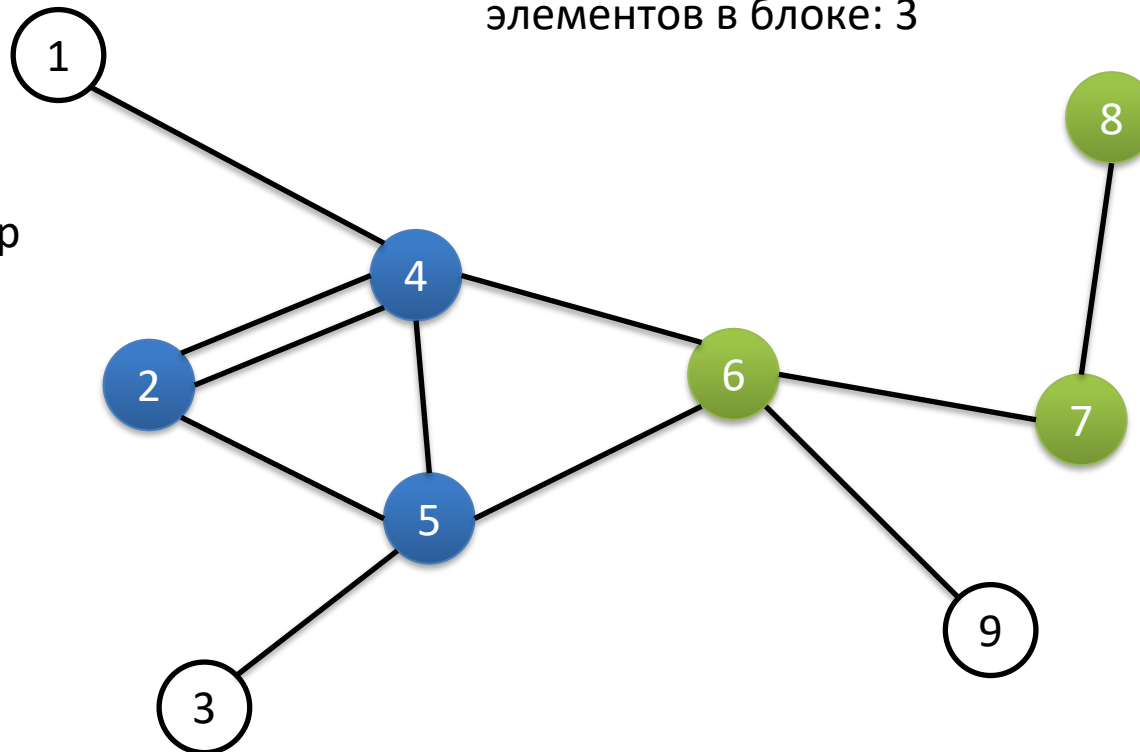
Шаг 1.

Находим лексикографически элемент с максимальным количеством связей

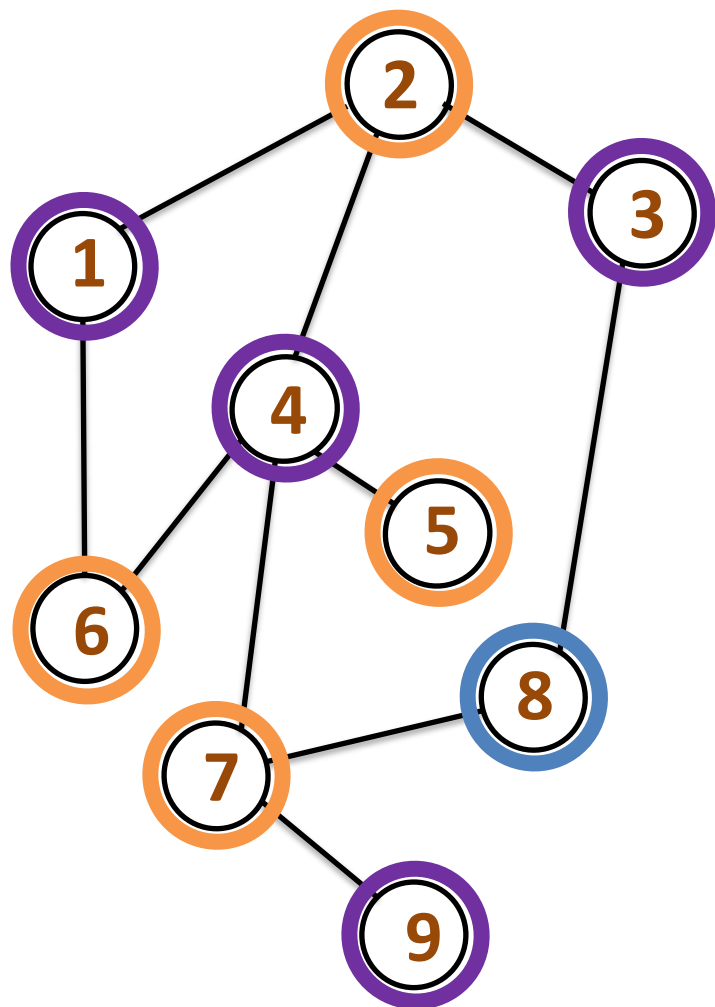
**Шаг 2.**

Находим лексикографически элемент, максимально связанный с формируемым блоком

Ограничение на размер элементов в блоке: 3



## Алгоритм раскраски графа



В один цвет раскрашиваются все несмежные вершины слева направо:

|   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|
| 4 | 2 | 7 | 1 | 3 | 6 | 8 | 5 | 9 |
| 4 | 3 | 3 | 2 | 2 | 2 | 2 | 1 | 1 |
| ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ |

Когда нельзя применить цвет к вершинам (мы рассмотрели все вершины), мы выбираем новый цвет и начинаем заново