

# Графы

Графом  $G(V, E)$  называется совокупность двух множеств – множества вершин  $V$  и множества ребер  $E$  – неупорядоченных пар множества  $V$ :  $G(V, E) = \langle V, E \rangle$ ,  $V \neq \emptyset$ ,  $E \in V \times V$ .

Инцидентность. Смежность. Степень (валентность).

Полустепень захода, исхода.

Ориентированный граф – упорядоченные пары (узлы?, дуги).

Маршрут - цепь (все ребра различны) – простая цепь (все вершины различны) – цикл (замкнутая цепь) – простой цикл (замкнутая простая цепь). Цепь – путь, цикл – контур.

Связность.

Длина маршрута – количество ребер в нем. Расстояние – длина кратчайшей цепи. Диаметр – наибольшее расстояние.

Эксцентриситет – максимальное расстояние от вершины. Радиус – наименьший эксцентриситет. Центр – множество центральных вершин (эксцентриситет = радиус графа).

# Графы. Операции над графами

**Создание.**

**Добавление вершины.**

**Добавление ребра.**

**Удаление вершины.**

**Удаление ребра.**

**Объединение графов.**

**Соединение графов.**

++++++

**Определение связных компонент**

**Поиск путей, кратчайших путей**

**Транзитивное замыкание**

**Остовные [минимальные] деревья**

**Раскраска графов**

**Циклы (эйлеровы, гамильтоновы)**

**Покрытия**

# Представление графов

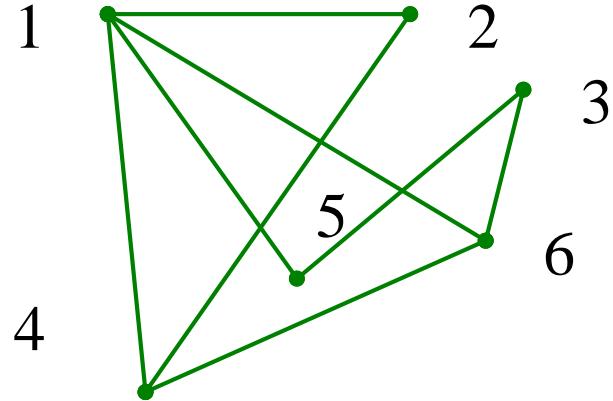
**Матрица смежности.**  $M[i, j] = \{1, v_i \text{ смежна с } v_j; 0 \text{ в противном случае}\}$ .  $O(p^2)$ . Для обычного графа достаточно хранить одну «половину» матрицы. Для взвешенного графа матрица смежности может хранить веса, связанные с ребрами.

**Матрица инцидентности.**  $H[i, j] = \{1, v_i \text{ смежна с } e_j; 0 \text{ в противном случае}\}$ . Для ориентированного графа  $H[i, j] = \{1, v_i \text{ инцидентна } e_j \text{ и является его концом}; 0 - \text{не инцидентны}; -1, v_i \text{ инцидентна } e_j \text{ и является его началом}\}$ .  $O(pq)$ .

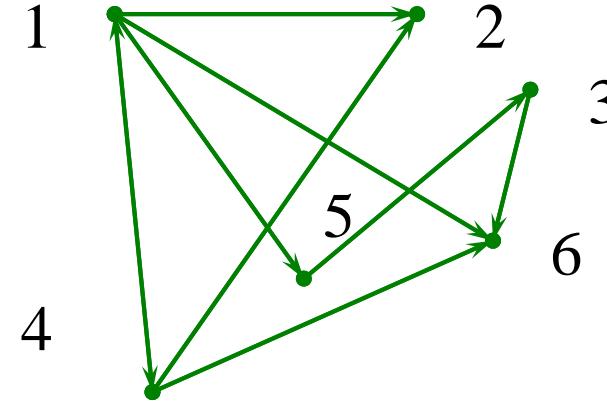
**Списки смежности.** Массив (список) списков смежных вершин.  $O(p+2q)$  – для неориентированного графа,  $O(p+q)$  – для ориентированного.

**Список (массив) ребер.** Массив (список) пар смежных вершин.  $O(q)$ .

# Представление графов. Матрица смежности

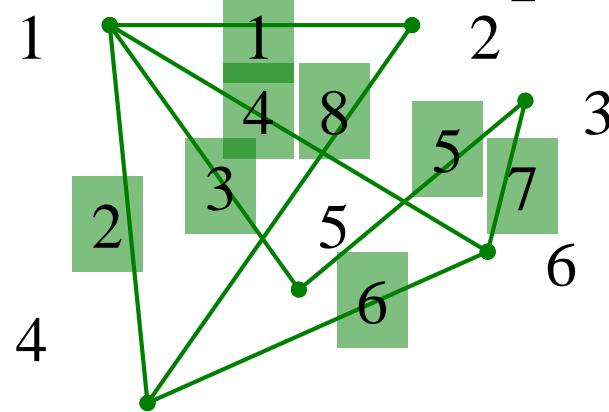


0	1	0	1	1	1
1	0	0	1	0	0
0	0	0	0	1	1
1	1	0	0	0	1
1	0	1	0	0	0
1	0	1	1	0	0

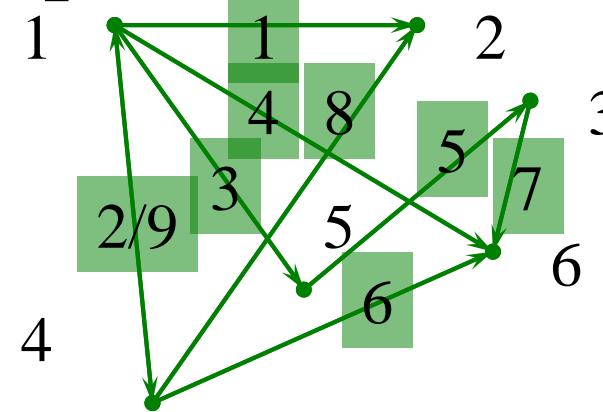


0	1	0	1	1	1
0	0	0	1	0	0
0	0	0	0	0	1
1	1	0	0	0	1
0	0	1	0	0	0
0	0	0	0	0	0

# Представление графов. Матрица инцидентности

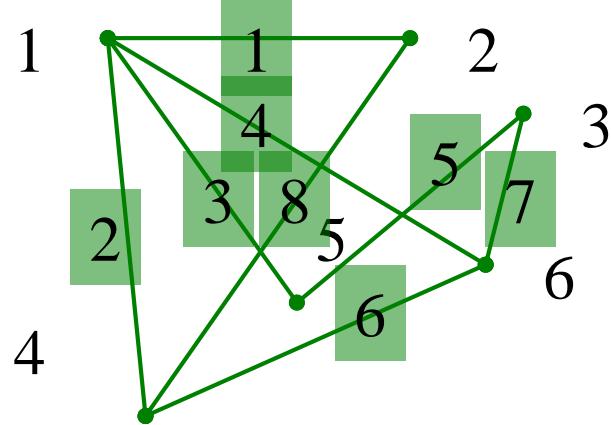


1	1	0	0	0	0
1	0	0	1	0	0
1	0	0	0	1	0
1	0	0	0	0	1
0	0	1	0	1	0
0	0	0	1	0	1
0	0	1	0	0	1
0	1	0	1	0	0

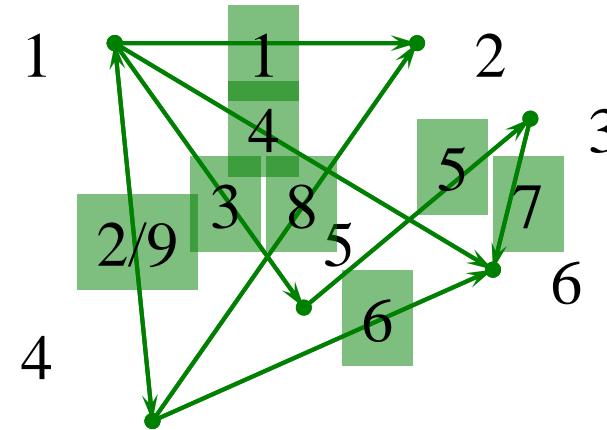


-1	1	0	0	0	0
-1	0	0	1	0	0
-1	0	0	0	1	0
-1	0	0	0	0	1
0	0	1	0	-1	0
0	0	0	-1	0	1
0	0	-1	0	0	1
0	1	0	-1	0	0
1	0	0	-1	0	0

# Представление графов. Списки смежности

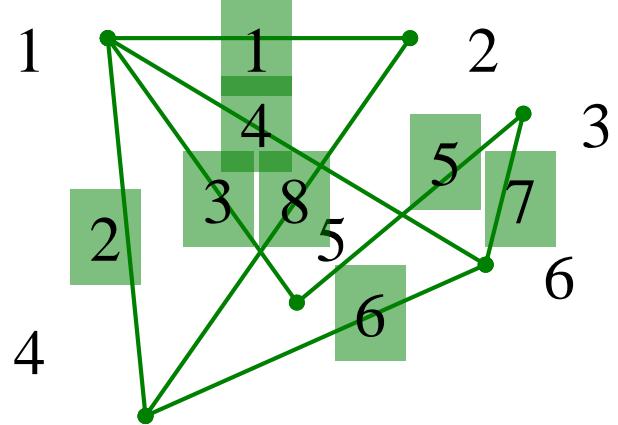


2, 4, 5, 6
1, 4
5, 6
1, 2, 6
1, 3
1, 3, 4

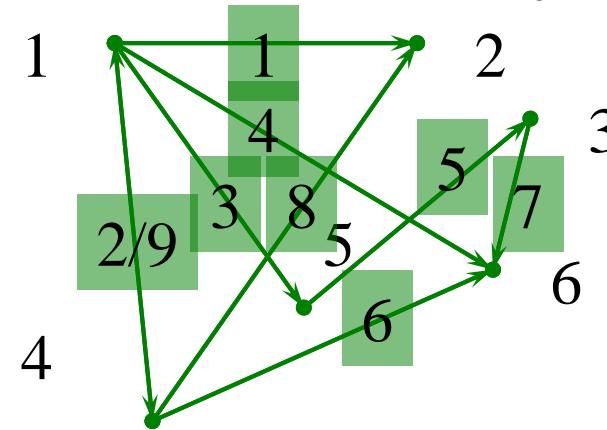


2, 4, 5, 6
6
1, 2, 6
3

# Представление графов. Списки дуг

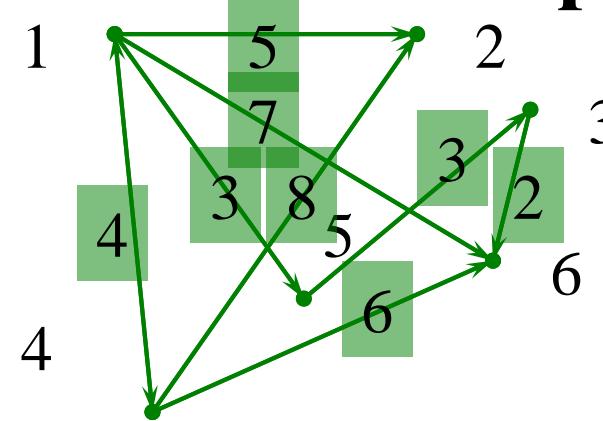
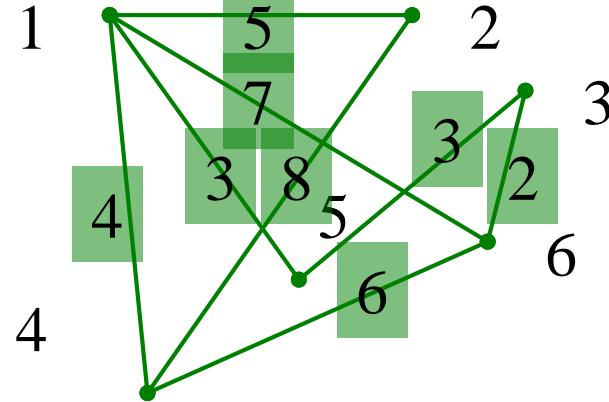


1, 2
1, 4
1, 5
1, 6
2, 4
3, 5
3, 6
4, 6



1, 2
1, 4
1, 5
1, 6
4, 2
5, 3
3, 6
4, 6
4, 1

# Представление графов. Взвешенные графы



0	5	$\infty$	4	3	7
5	0	$\infty$	8	$\infty$	$\infty$
$\infty$	$\infty$	0	$\infty$	3	2
4	8	$\infty$	0	$\infty$	6
3	$\infty$	3	$\infty$	0	$\infty$
7	$\infty$	2	6	$\infty$	0

0	5	$\infty$	4	3	7
$\infty$	0	$\infty$	$\infty$	$\infty$	$\infty$
$\infty$	$\infty$	0	$\infty$	$\infty$	2
4	8	$\infty$	$\infty$	$\infty$	6
$\infty$	$\infty$	3	$\infty$	0	$\infty$
$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	0

# Обходы графов

граф  $G(V, E)$ , вспомогательные массив Visited[p] = {0} и контейнер C

1.  $v \leftarrow G(V, E)$
2.  $v \rightarrow C$
3. Visited[v] = 1
4.  $u \leftarrow C$
5. process  $u$
6. for  $w \in \Gamma(u)$ 
  7. if (! Visited[w])
    - 7.1.  $w \rightarrow C$
    - 7.2. Visited[w] = 1
  8. if ( $C \neq \emptyset$ ) п.4
9. end

$\Gamma(u)$  – множество вершин, смежных с  $u$ .

Если контейнер – *стек*, то обход в глубину, если – *очередь*, то в ширину. Сложность зависит от способа представления, для матрицы смежности –  $O(p^2)$ , для списков смежности –  $O(p + q)$ .  
9

# Транзитивное замыкание

Транзитивным замыканием графа  $G$  называется граф  $G'$ , в котором существует дуга из вершины  $i$  в вершину  $j$ , если в графе  $G$  существует путь из вершины  $i$  в вершину  $j$ .

1.  $M^1 = M$ ,

2.  $M^k[i, j] = \textcolor{blue}{V} M^{k-1}[i, l] \textcolor{red}{\&} M[l, j], l \in [1, p]$ ,

где  $M^k[i, j] = 1$ , если существует путь из  $i$  в  $j$  длиной не более  $k$ .

3.  $T = \textcolor{blue}{V} M^k, k \in [1, p]$

Сложность представленного решения  $O(p^4)$ .

Транзитивное замыкание  $\rightarrow$  задача достижимости

# Транзитивное замыкание. Алгоритм Уоршалла

Алгоритм Уоршалла обеспечивает сложность  $O(p^3)$ :

1. **for**  $l \in [1, p]$ 
  2. **for**  $i \in [1, p]$ 
    3. **for**  $j \in [1, p]$ 
      4.  $T[i, j] = T[i, j] \text{ } \textcolor{blue}{\vee} \text{ } T[i, l] \& T[l, j]$

## Модификация алгоритма

1. **for**  $l \in [1, p]$ 
  2. **for**  $i \in [1, p]$ 
    3. **if** ( $T[i, l] = 1$ )
      4. **for**  $j \in [1, p]$ 
        5. **if** ( $T[l, j] = 1$ )  $T[i, j] = 1$

# Кратчайшие пути. Алгоритм Дейкстры

граф  $G(V, E)$ , вершины From, To, вспомогательные массивы  
 $\text{Length}[p] = \{\infty\}$ ,  $\text{Mark}[p] = \{0\}$ ,  $\text{History}[p] = \{0\}$

1.  $v = \text{From}$ ,  $\text{Length}[v] = 0$

2. while ( $v \neq \text{To} \ \& \ v \neq 0$ )

    2.1.  $\text{Mark}[v] = 1$

    2.2. for  $u \in \Gamma(v)$

        2.2.1. if ( $\text{Mark}[u]=0 \ \& \ \text{Length}[u] > \text{Length}[v] + W[v, u]$ )

            2.2.1.1.  $\text{Length}[u] = \text{Length}[v] + W[v, u]$

            2.2.1.2.  $\text{History}[u] = v$

    2.3.  $\text{Min} = \infty$ ,  $v = 0$

    2.4. for  $u \in [1, p]$

        2.4.1. if ( $\text{Mark}[u]=0 \ \& \ \text{Length}[u] < \text{Min}$ )

            2.4.1.1.  $v = u$

            2.4.1.2.  $\text{Min} = \text{Length}[u]$

3. end

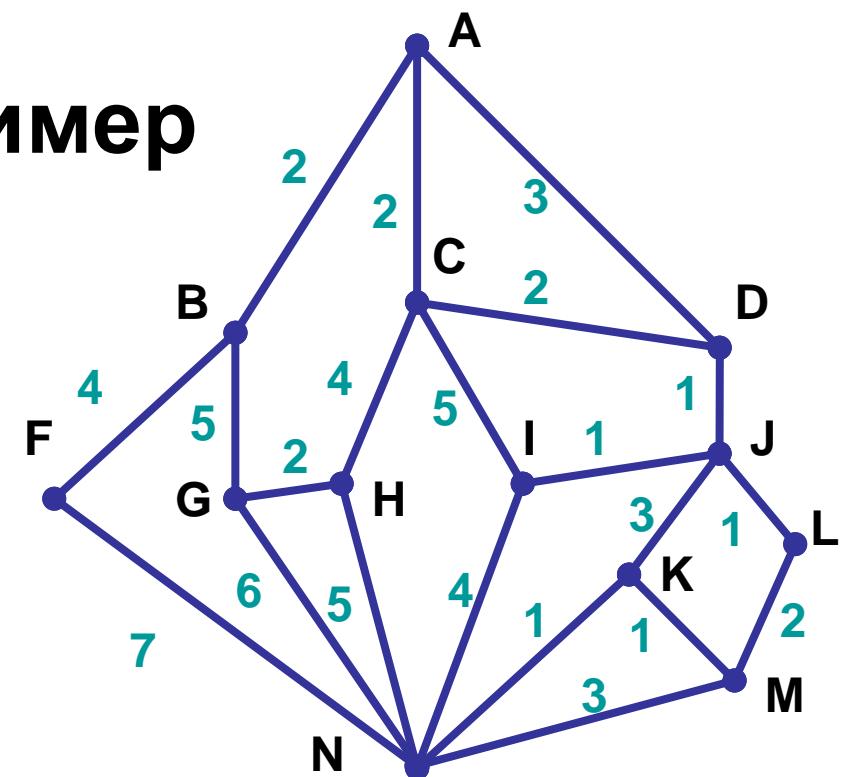
# Алгоритм Дейкстры. Пример

1	A	B	C	D	F	G	H	I	J	K	L	M	N
L	0												
M	1												
H													

2	A	B	C	D	F	G	H	I	J	K	L	M	N
L	0	2	2	3									
M	1	1											
H	A	A	A	A									

3	A	B	C	D	F	G	H	I	J	K	L	M	N
L	0	2	2	3	6	7							
M	1	1	1										
H	A	A	A	A	B	B	C	C	D				

4	A	B	C	D	F	G	H	I	J	K	L	M	N
L	0	2	2	3	6	7	6	7					
M	1	1	1	1									
H	A	A	A	A	B	B	C	C	D	J	J	13	



5	A	B	C	D	F	G	H	I	J	K	L	M	N
L	0	2	2	3	6	7	6	7		4			
M	1	1	1	1						1			
H	A	A	A	B	B	C	C	D					

6	A	B	C	D	F	G	H	I	J	K	L	M	N
L	0	2	2	3	6	7	6	7	4	7	5		
M	1	1	1	1					1	1	1		
H	A	A	A	B	B	C	C	D	J	J	13		

# Алгоритм Дейкстры. Пример

7	A	B	C	D	F	G	H	I	J	K	L	M	N
L	0	2	2	3	6	7	6	7	4	7	5	7	
M	1	1	1	1	1			1	1				
H	A	A	A	B	B	C	C	D	J	J	L		

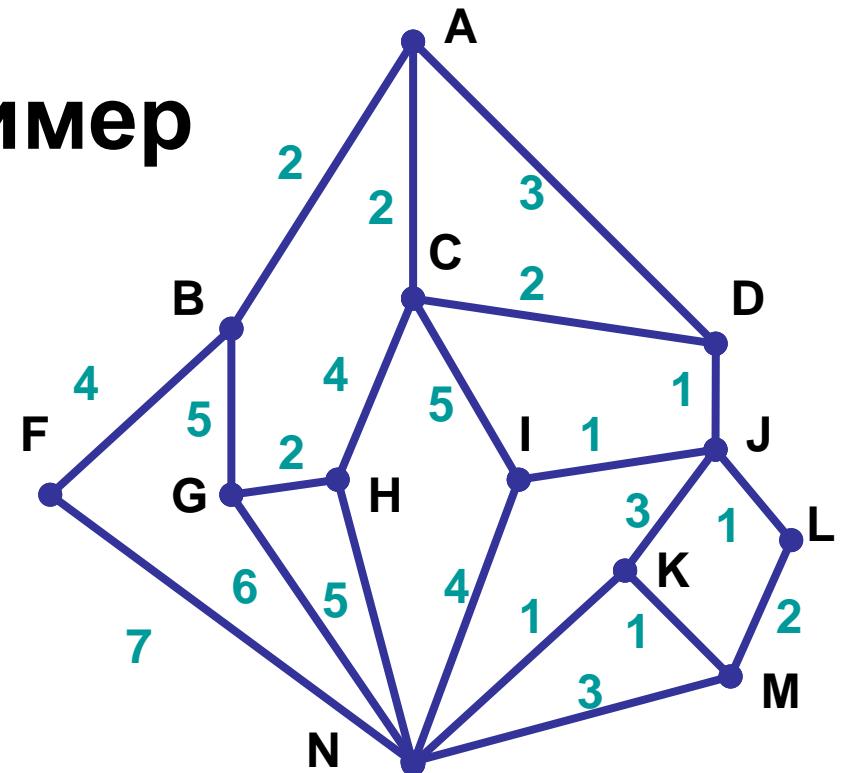
8	A	B	C	D	F	G	H	I	J	K	L	M	N
L	0	2	2	3	6	7	6	7	4	7	5	7	13
M	1	1	1	1	1		1	1	1				
H	A	A	A	B	B	C	C	D	J	J	L	F	

9	A	B	C	D	F	G	H	I	J	K	L	M	N
L	0	2	2	3	6	7	6	7	4	7	5	7	11
M	1	1	1	1	1	1	1	1	1	1			
H	A	A	A	B	B	C	C	D	J	J	L	H	

10	A	B	C	D	F	G	H	I	J	K	L	M	N
L	0	2	2	3	6	7	6	7	4	7	5	7	11
M	1	1	1	1	1	1	1	1	1	1	1	1	
H	A	A	A	B	B	C	C	D	J	J	L	H	

11	A	B	C	D	F	G	H	I	J	K	L	M	N
L	0	2	2	3	6	7	6	7	4	7	5	7	11
M	1	1	1	1	1	1	1	1	1	1	1	1	
H	A	A	A	B	B	C	C	D	J	J	L	H	

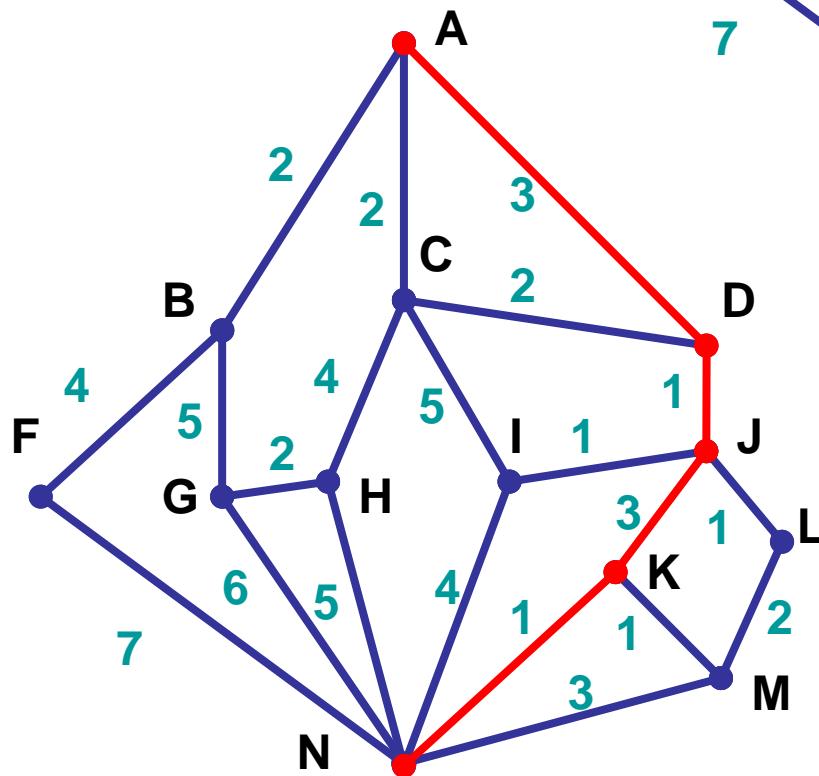
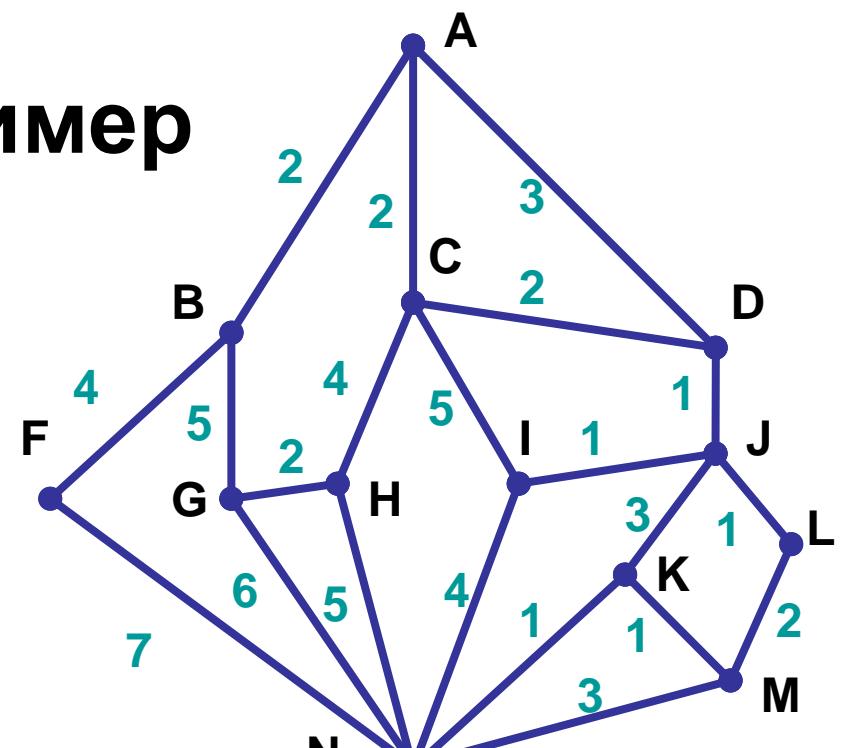
12	A	B	C	D	F	G	H	I	J	K	L	M	N
L	0	2	2	3	6	7	6	7	4	7	5	7	8
M	1	1	1	1	1	1	1	1	1	1	1	1	
H	A	A	A	B	B	C	C	D	J	J	L	K	



# Алгоритм Дейкстры. Пример

13

	A	B	C	D	F	G	H	I	J	K	L	M	N
L	0	2	2	3	6	7	6	7	4	7	5	7	8
M	1	1	1	1	1	1	1	1	1	1	1	1	
H		A	A	A	B	B	C	C	D	J	J	L	K



# Кратчайшие пути. Алгоритм Флойда

## Основан (или аналогичен) на алгоритме Уоршалла

граф  $G(V, E)$ ,  $\text{History}[p, p] = \{ 0 \}$ .

$\text{History}[i, j] = 0$ , если пути из  $i$  в  $j$  нет; равно  $j$ , если есть.

1. **for**  $l \in [1, p]$ 
  2. **for**  $i \in [1, p]$ 
    3. **if** ( $W[i, l] \neq \infty$ )
      4. **for**  $j \in [1, p]$ 
        5. **if** ( $W[i, j] > W[i, l] + W[l, j]$ )
          - 5.1.  $W[i, j] = W[i, l] + W[l, j]$
          - 5.2.  $\text{History}[i, j] = \text{History}[i, l]$

Путь из вершины  $i$  в  $j$ :

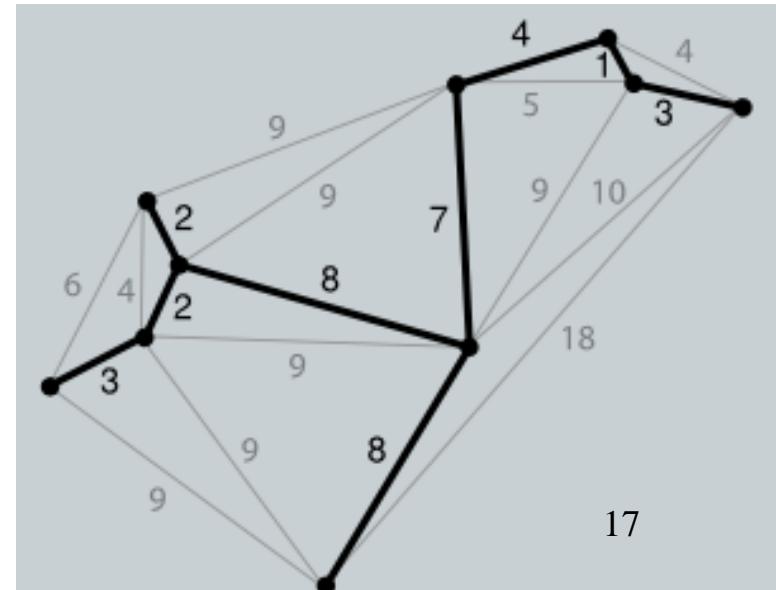
1.  $l = i$ , **yield**  $l$
2. **while** ( $l \neq j$ )
  - 2.1.  $l = \text{History}[l, j]$
  - 2.2. **yield**  $l$

# Минимальные остовные деревья

Пусть  $G(V, E)$  – граф. Остовной подграф графа  $G(V, E)$  – это подграф, содержащий все вершины. Остовной подграф, являющийся деревом, называется остовом или остовным деревом.

Если заданы длины ребер или веса, то можно поставить задачу нахождения минимального (кратчайшего) остова, т.е. остова, сумма длин ребер которого минимальна.

Например, пусть задано множество аэропортов и необходимо определить минимальный (по сумме расстояний) набор авиарейсов, который позволил бы перелететь с любого аэропорта на любой другой. Решением этой задачи будет кратчайший остов полного графа расстояний между аэропортами.



# Алгоритм Прима

Пусть задан граф  $G(V, E)$  и матрица весов  $W$ , в результате должны получить остов  $T(V', E')$

1.  $V' = \{v_i\}$ ,  $v_i \in V_i$  /\*выбираем произвольную вершину\*/
2. while  $|V'| \neq |V|$ 
  - 2.1  $e = \min E$ ,  $e = (v_i, v_j)$ ,  $v_i, v_j \in V$ ,  $v_i \in V'$ ,  $v_j \notin V'$
  - 2.2  $E' \leftarrow e$

Асимптотическая сложность:

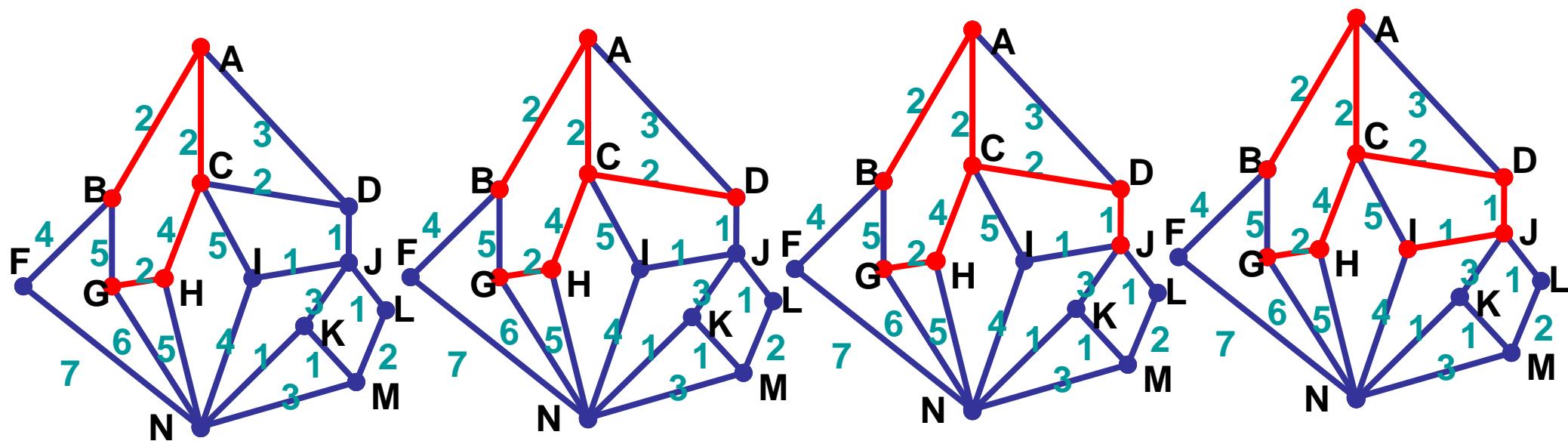
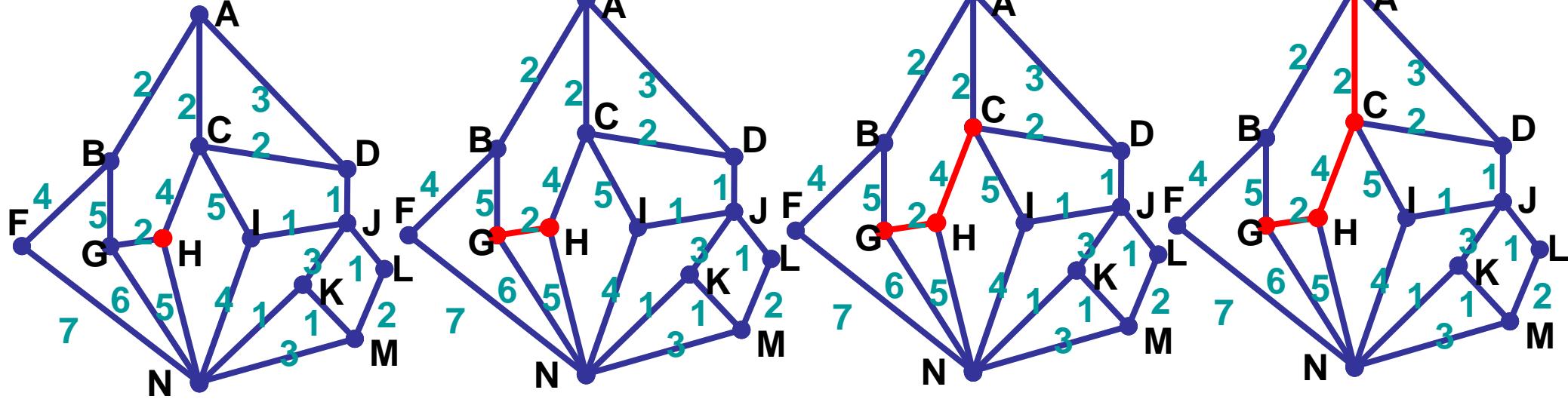
$O(p^2)$  – матрица смежности;

$O(q \log_2 p)$  – бинарная куча и списки смежности ( $O((V+E)\log_2 V)$ );

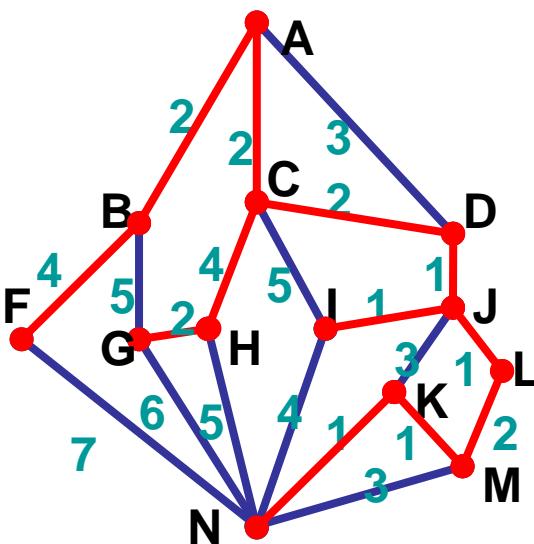
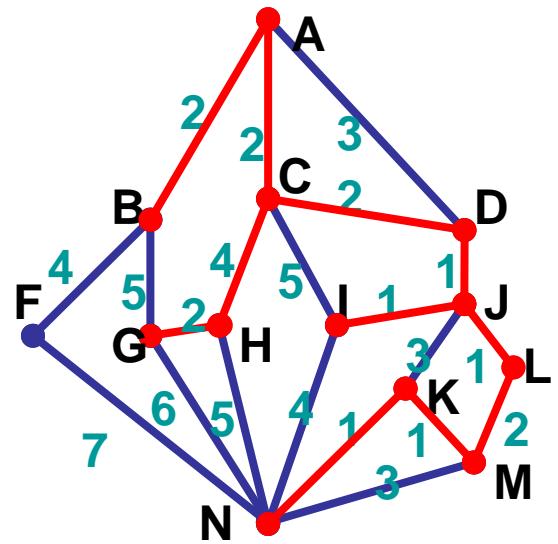
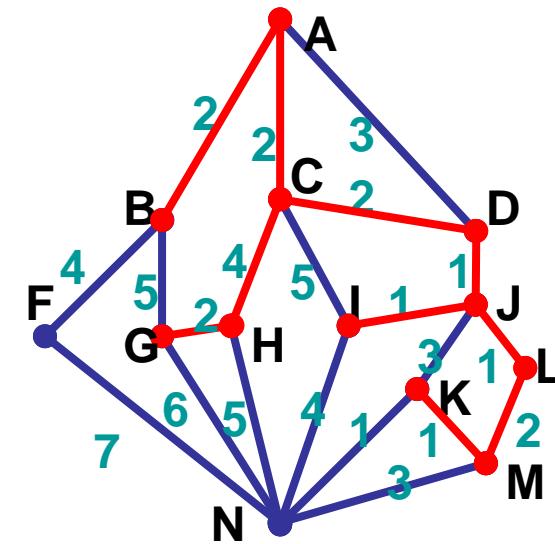
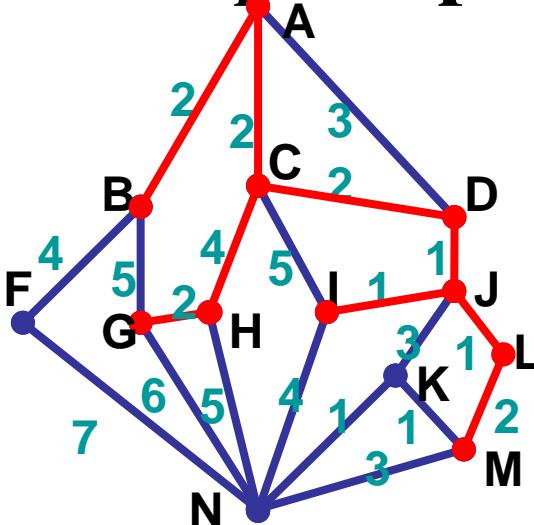
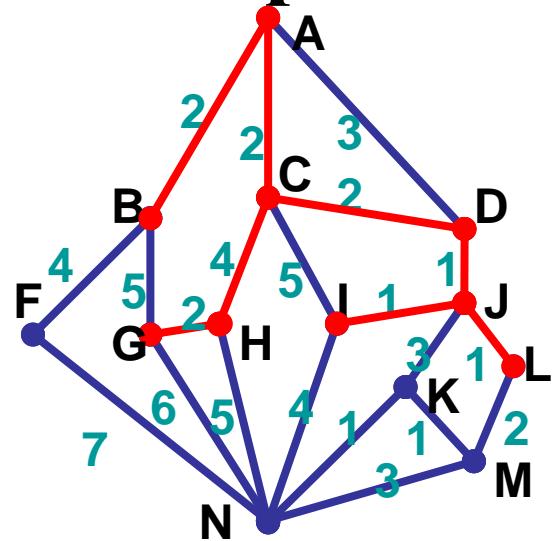
$O(q + p \log_2 p)$  – куча Фибоначчи и списки смежности.

Эффективен для насыщенных графов.

# Алгоритм Прима. Пример



# Алгоритм Прима. Пример



# Алгоритм Кра[у]скала (1956)

Граф  $G(V, E)$ ,  $E^s$  – отсортированное множество ребер.

1.  $E' := \emptyset$ ;  $E^s = \text{sort } E$ ;  $k := 1$
2. **for**  $i \in [1; p - 1]$ 
  - 2.1. **while**  $E' \cup \{e^s_k\}$  образует цикл
    - 2.2.1.  $k = k + 1$
    - 2.2.  $E' = E' \cup \{e^s_k\}$

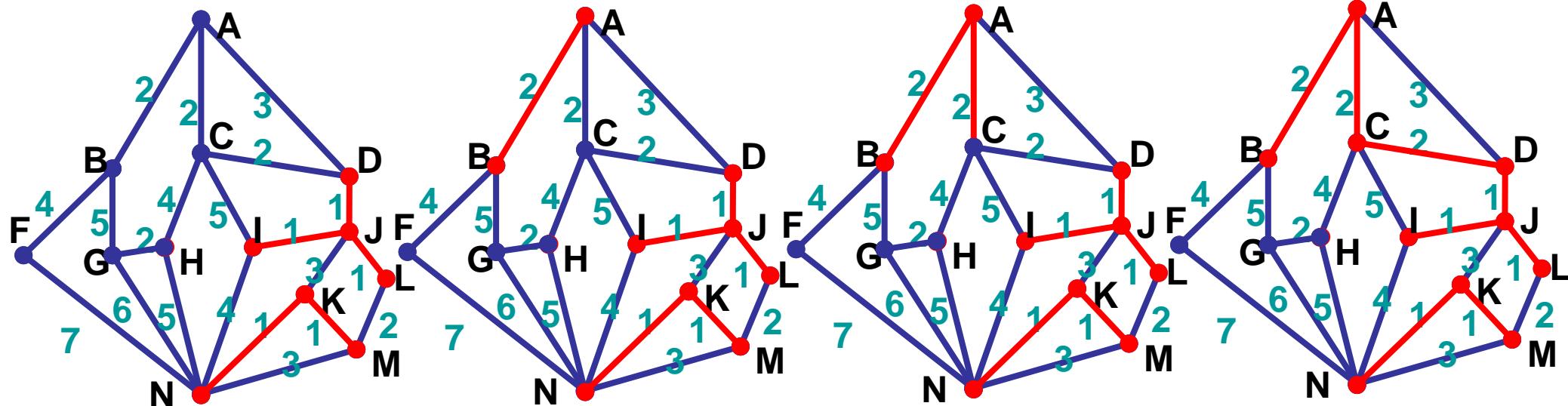
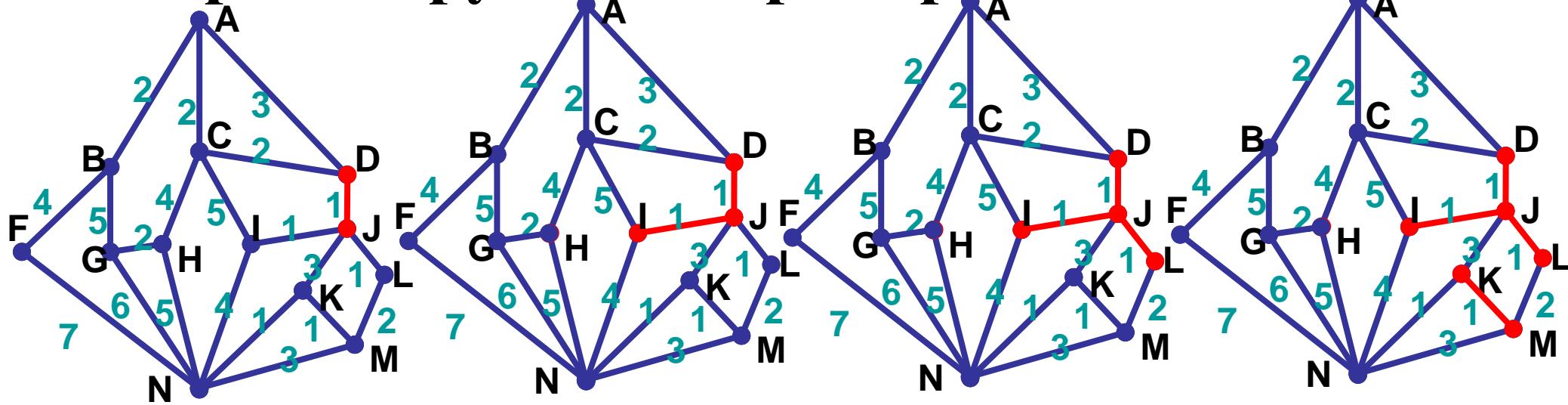
**Асимптотическая сложность:**

$O(q \log_2 p)$

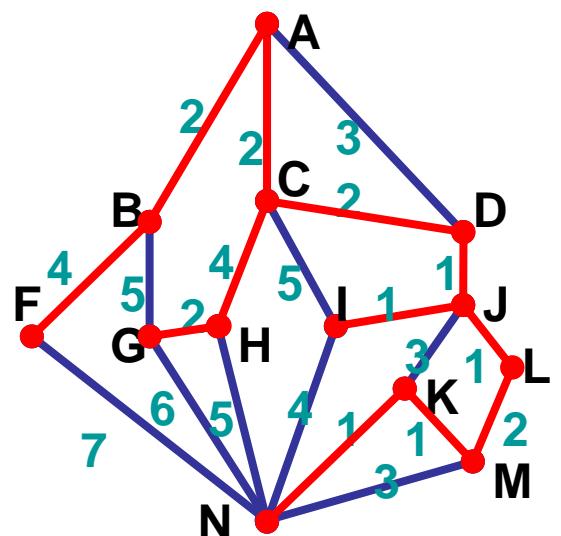
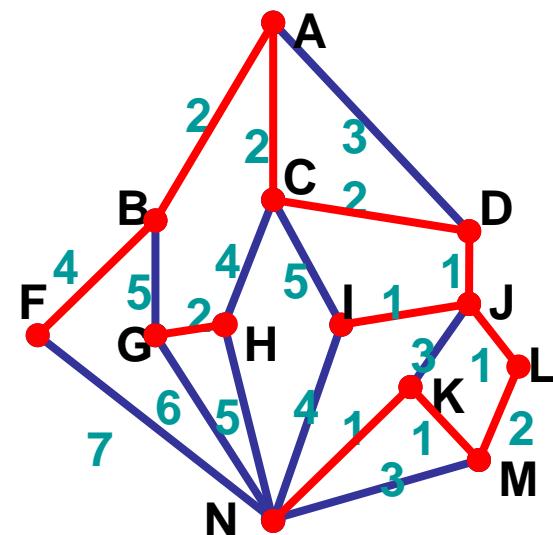
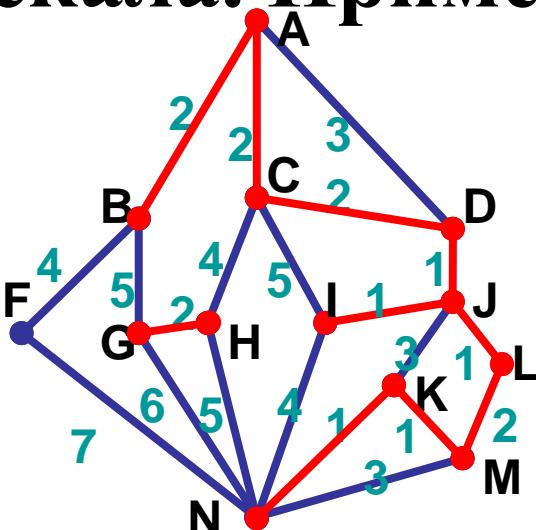
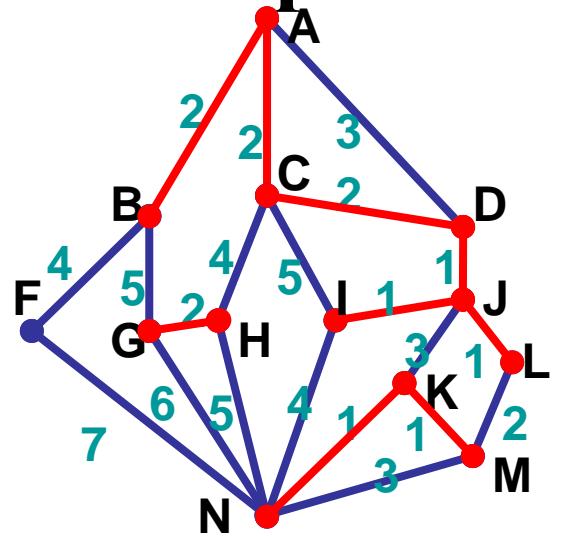
$O(q \log_2 q)$

**Эффективен для ненасыщенных графов.**

# Алгоритм Крускала. Пример



# Алгоритм Крускала. Пример



# Алгоритм Борувки (1926, первый)

Граф  $G(V, E)$ ,  $L$  – множество деревьев.

1.  $L = \{v_1, v_2, \dots\}; V' = V; E' = \emptyset$

2. while  $|L| > 1$

    2.1. for  $t \in L$

        2.2.1.  $e = \min E, e = (v_i, v_j), v_i, v_j \in V, v_i \in t, v_j \notin t$

        2.2.2.  $t \leftarrow e$

Асимптотическая сложность:

$O(q \log_2 p)$

