Кратчайшие пути между всеми парами вершин (APSP)

Алгоритм Флойда-Уоршалла $\underline{O(n^3)}$

Динамическое программирование

Выход: $\{(d_{ij}, p_{ij})\}_{i,j=1}^{n}$

d_{ij} - длина пути от i до j

p_{ij} - предпоследняя вершина на пути $i \rightsquigarrow j$

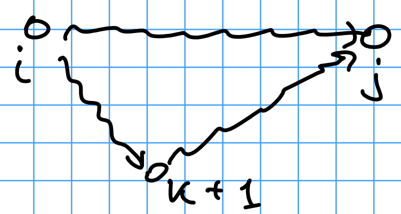
$\Rightarrow$ Путь можно восстановить за O(длина)

1. $D[i,j,k]$ = длина кратчайшего пути из i в j через вершины $\{1,\dots k\}$

$D[i,j,0]$ = матрица смежности

2. $D[i,j,k+1] = \min\{D[i,j,k],\ D[i,k+1,k] + D[k+1,j,k]\}$

3. Порядок - "по слоям"




```
Floyd-Worshal(G):
    D ← Матрица смежности
    P[i,j] = i если (i,j) ∈ E
    for k = 1 to n
        for i = 1 to n
            for j = 1 to n
                if D[i,j] > D[i,k] + D[k,j]
                    D[i,j] = D[i,k] + D[k,j]
                    P[i,j] = P[k,j]
```

$i \rightsquigarrow \underline{k} \rightarrow j$

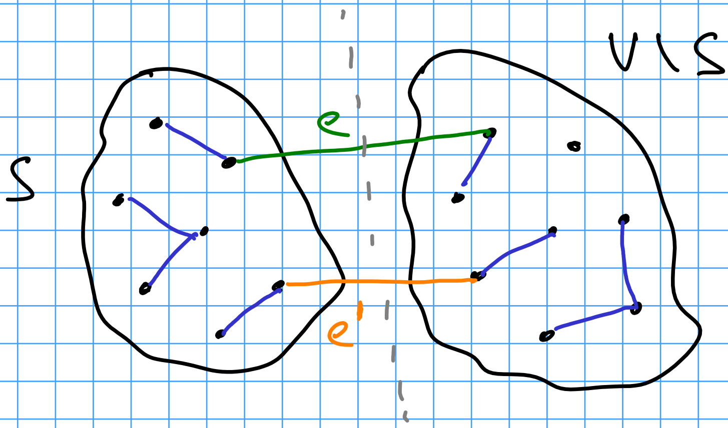
Утв: $D[i,k,k-1] = D[i,k,k]$, $D[k,j,k-1] = D[k,j,k]$.

Минимальное остовное дерево

Утв: Свойство разреза

$\exists$ S — разрез графа

S и $V \setminus S$ — две части разреза



$\exists$ M — подмножество некоторого MST:
в M нет рёбер, соединяющих S и $V \setminus S$

$\exists$ e — минимальное ребро соед. S и $V \setminus S$.

$\exists MST' : M \cup \{e\} \subseteq MST'$

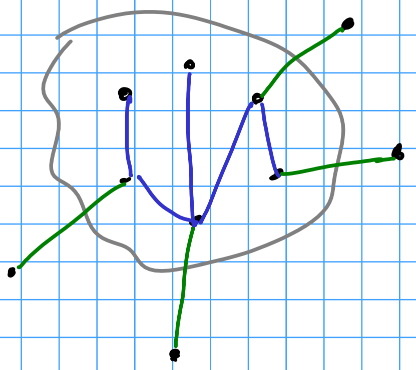
$\triangleright$ $\exists$ это неверно. $\measuredangle$ MST. Добавим в него e $\Rightarrow$ $\Rightarrow$ получился цикл. $\Rightarrow$ $\exists$ ребро e' этого цикла, что пересекает разрез S.

$w(e) \leq w(e') \Rightarrow MST \setminus e' \cup \{e\}$ — мин. ост. дер. $\triangleleft$

Алгоритм Прима

```
Q ← MakePriorityQueue((1,0)) // одна вершина
dist[1] = 0,   dist[i] = ∞, i > 1
prev[i] = 0 , ∀i

while Q.size() > 0:
    (v, d) ← Q.extract_min()
    for (v, u) ∈ E:
```



```
if dist[u] = ∞
    dist[u] = w(v, u)
    Q.insert((u, dist[u]))
    prev[u] = v
else if dist[u] > w(v, u)
    Q.decrease_key((u, w(v, u)))
    dist[u] = w(v, u)
    prev[u] = v
dist[v] = -∞
```

$$M = \{(prev[i], i)\}_{i=2}^{n}$$

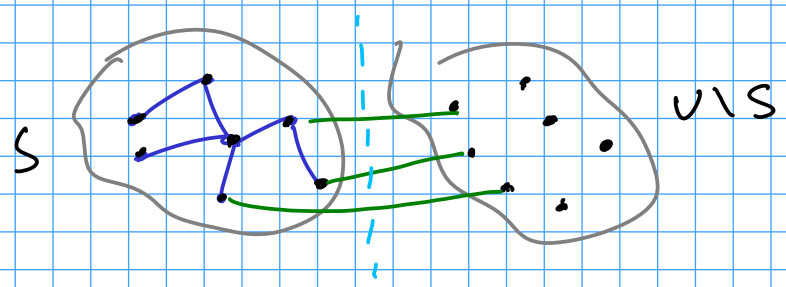
Утв: То, что мы получили — дерево

1. Получили связный граф.
2. В этом графе n вершин и $n-1$ ребро

Утв: M — MST

Пусть S — вершины добавл. в дерево ($dist[v] = -\infty$)

Рассмотрим разрез S и $V \setminus S$

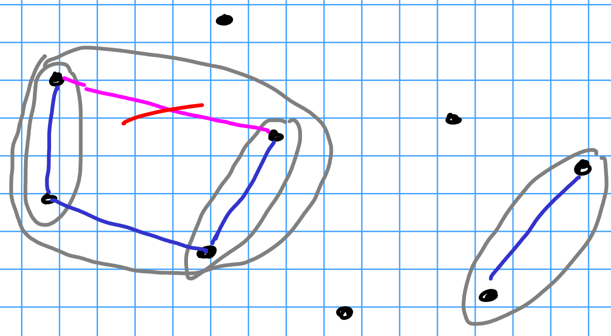


Заметим, что алгоритм добавляет на каждой итерации минимальное ребро из этого разреза. Применяем св-во разреза. ◁

Сложность = $O(E \cdot \log V)$

$O(V^2)$ для плотн. графа

Алгоритм Краскала (Kruskal)



```
Kruskal((V, E))
O(E log V)          sort(E, w)
                    for v in V:
O(V)·|makeset|          make_set(v)
                    for (v, u) ∈ E:
O(E)·|set|              if set(v) ≠ set(u)
                            M = M ∪ {(v, u)}
O(V)|union|                 union(set(v), set(u))
                    return M
```

Если $|set|$ работает медленнее $O(\log V)$
$|set| \neq O(\log V)$

Следующее решение.
$O(E \cdot \log^* V)$

Утв.: На каждой итерации добавляется ребро, кот. является минимальным в разрезе $set(u)$ и $V \setminus set(u)$.
$\Rightarrow$ применяем св-во разреза ◁