ДИ) Имаме неориентиран свързан граф $G=\langle V, E\rangle$, $w: E \rightarrow \{a, b\}$

$0 < a < b$.

Нека r е теглото на някое MST на G.
Как да намерим това r с линеен алгоритъм, който не е нужно да намира MST.

Дадено: $G=\langle V, E\rangle$ свързан, неор.

I стъпка: Взимаме $G' \subseteq_{\text{подграф}} G$, i.e.

$G=\langle V, E'\rangle : E' = E \setminus \{e \mid w(e)=b\}$

$E' = \{e \mid w(e)=a\}$

Сл. 1 Свързан е
Сл. 2 Не е свързан
$\Big\}$ G'

Сл. 1 G' е свързан и всички ребра са с еднакво тегло

прим. с dfs намираме покриващо дърво $\to O(|V|+|E'|)$

$r \geq (n-1).a$ ребра

Сл. 2 G' не е свързан

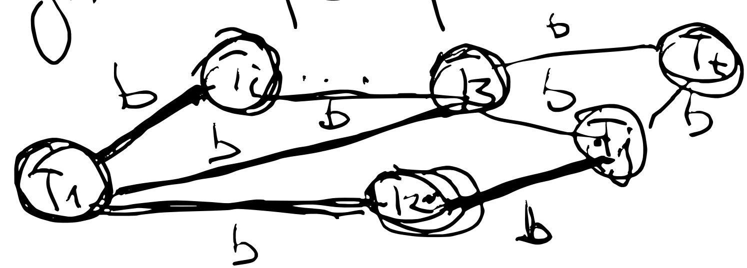
Нека с t бележим броя на компонентите на свързаност в G'.
Нека с $V_1, \dots, V_t$ - м-ва от върхове
V_i - върхове участващи в i-та комп.

- Нека за всяко $G_i = \langle V_i, \mathcal{E}_i^{\circ} \rangle$ бележим i-та комп. на свързаност с ребра $\mathcal{E}_i \subseteq \mathcal{E}'$ и върхове $\underline{V_i \subseteq V}$.

От всяко G_i пускаме dfs и получаваме покриващо дърво на $G_i \Rightarrow \underline{T_i} = \langle V_i, \mathcal{E}_i'' \rangle$ $\underline{|\mathcal{E}_i''| = |V_i| - 1}$.

В крайна сметка получаваме горе $\underline{T_1}, \ldots, \underline{T_t}$.

Нека T_i да го третираме като нов връх



Последно строим граф

$G'' = \langle \{T_1, \dots, T_t\}$, тежки ребра $|_{E \setminus E'} \rangle$

пускаме dfs/bfs и получаваме покриващо дърво за него, което ползва точно $(t-1)$ тежки ребра тип b

$\Theta(|V|+|E|)$

$$r = \sum_{i=1}^{t} (|V_i|-1) \cdot a + (t-1) \cdot b =$$

$$= a \cdot \Big(\underbrace{\sum_{i=1}^{t} |V_i|}_{|V|=n} - t\Big) + (t-1) \cdot b = a \cdot (n-t) + (t-1) \cdot b$$

Зад Модифицирайте алгоритъма на Floyd-Warshall да намира транзитивно затваряне на матрицата на съседство.

```
Floyd-Warshall (G, w):
  r[1...n][1...n]: reachability
  init(G, w, r)
  for k ← 1 to n do
     for i ← 1 to n do
        for j ← 1 to n do
           r[i][j] ← r[i][j] v
             (r[i][k] & r[k][j])
```

```
init(G, w, r):
    for i ← 1 to n do
        r[i][i] ← 1
        for j ← 1 to n do
            if <i, j> ∈ E
                r[i][j] ← 1
            else
                r[i][j] ← 0
```

Зад Даден е ориентиран граф $G=\langle V,E\rangle$ без цикли. Предложете алгоритъм с време $O(|V|+|E|)$, който намира броя пътища между два върха s и t.

Имаме DAG и търсим # пътища от s до t

$1 \ldots n$ $\quad 1 \le s \le n \quad 1 \le t \le n$

Стъпка 1: $e \leftarrow$ topoSort(G)

paths[i] = # пътища от връх i до t

paths[i] ←

i е преди t в топо нареждането

paths[t] ← 1

for u in reverse(e)

  if u is before t in e

    for $v \in adj(u)$ $\langle u, v \rangle$

      paths[u] ← paths[u] + paths[v]

paths[t] ← 1

paths[i] ← 0

$i \neq t$

return paths[s]

$$\text{paths}[u] \leftarrow \sum_{\langle u,v\rangle \in \mathcal{E}} \text{paths}[v]$$

u before t

3ад Най-къс път и най-дълъг път в DAG: $G = \langle V, E \rangle$ от връх v до всички останали, с произволни тегла.

→ топологическа наредба на върховете

Bellman-Ford - $O(|V|.|E|)$

```
alg(G w s) < dist[|V|]
  L ← topSort(G)
  for v in V do
      dist[v] ← +∞
  dist[s] ← 0
```

for v in el2sy do $O(V)$

for $\langle u,v\rangle \in E$ do

relax(G, u, v)

$O(\text{indegree}(v))$

for $i \leftarrow 1$ to $|V|-1$

for $\langle u,v\rangle \in E$

if $d[s][v] > d[s][u] + w(\langle u,v\rangle)$ $\leq i$

$d[s][v] \leftarrow d[s][u] + w(\langle u,v\rangle)$

relax(G, u, v)

$O(|V| + |E|)$

$$dist[v]_{k} \leftarrow \min_{\langle u,v\rangle \in E}\{dist[u]_{k-1} + w(\langle u,v\rangle)\}$$

$$\begin{tikzcd} a \arrow[r] & b \arrow[r] & c \arrow[ll, bend right] \end{tikzcd}$$

$dist[c]$

$\uparrow$

$dist[b]$

$\uparrow$

$dist[a]$

$\uparrow$

$dist[c]$