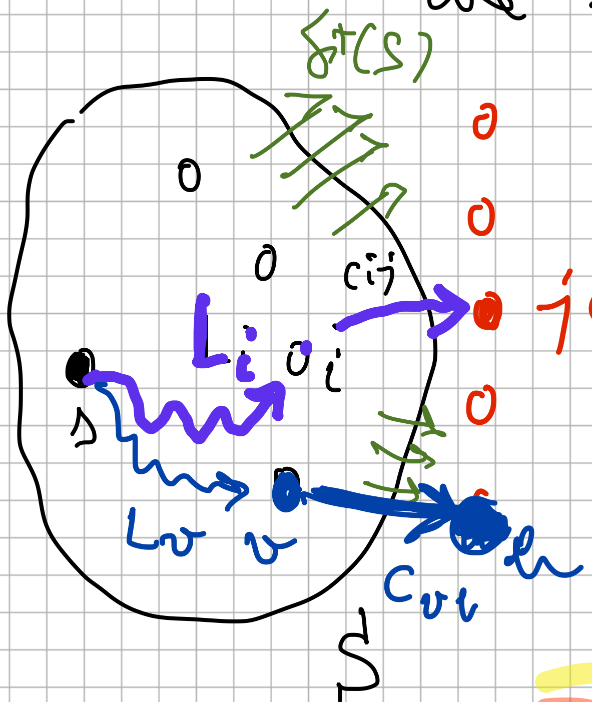
OR 1 1-DEC-2021

DIJKSTRA'S ALG.

Hp: $\boxed{c_{ij} \geq 0}$, $\forall (i,j) \in A$

$s \in V$ SOURCE ($t \in V$ TARGET)

- L_h := cost of the minimum path $s \to h$, $\forall h \in V$
- ($pred_h$ predecessor of h in the shortest path $s \to h$)



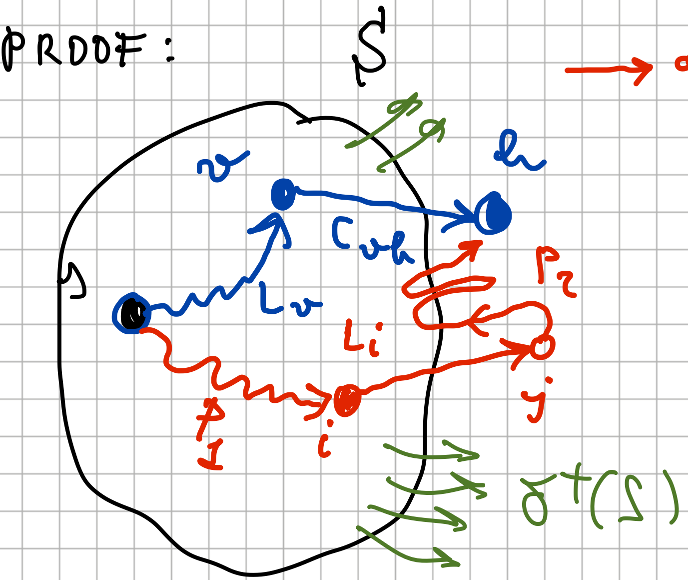
$s \in S$

Values L_i, $\forall i \in S$ are known ($L_s = 0$)

$$(v,h) := \operatorname{argmin} \left\{ L_i + c_{ij} : \forall (i,j) \in \delta^+(S) \right\}$$

$\Rightarrow$ $\underline{L_v + c_{vh}}$ IS THE COST OF THE SHORTEST PATH $s \to h$

PROOF:



→ alternative path P' $s \to h$

$$P' = P_1 \cup \{(i,j)\} \cup P_2$$

$$\mathrm{cost}(P') = \boxed{\mathrm{cost}(P_1)} + c_{ij} + \boxed{\mathrm{cost}(P_2)} \geq \boxed{L_i} + c_{ij} + \boxed{0}$$

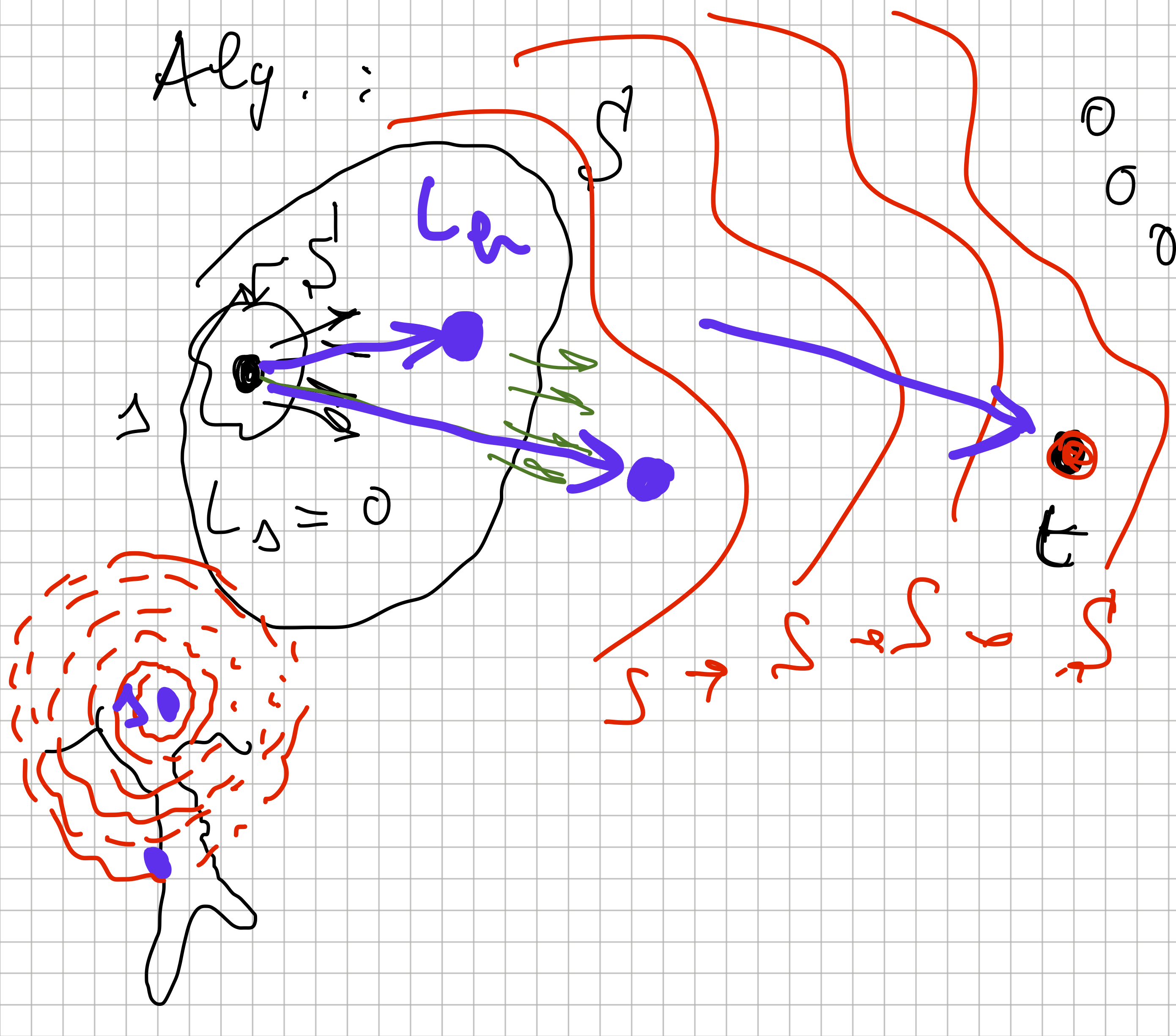
because $\mathrm{cost}(P_2) \geq 0$ ($\Leftarrow$ all costs are nonnegative)

due to the argmin ...

$$\geq L_v + c_{vh}$$

∎

Alg.:



TIME COMPLEXITY:

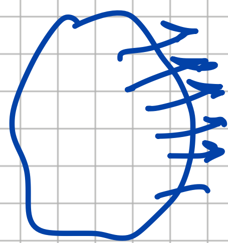
$$C = \begin{array}{cc} & j \\ i & \begin{bmatrix} & \vert & \\ - & \boxed{c_{ij}} & - \\ & \vert & \end{bmatrix} \end{array}$$

COMPLETE GRAPH

$n \times n$ input matrix

$\Rightarrow$ $\boxed{O(n^2) \text{ time}}$ for reading the input

BASIC IMPLEM.



$\delta^+(S)$

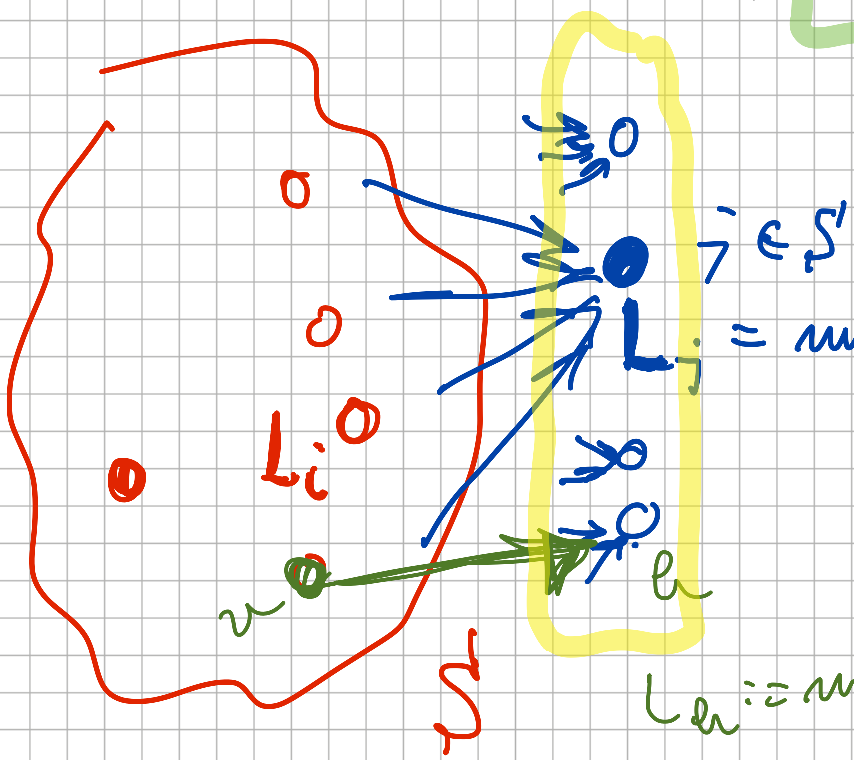
$L_i + c_{ij}$

$\sim n/2 \cdot n/2$

$\sim O(n^2)$

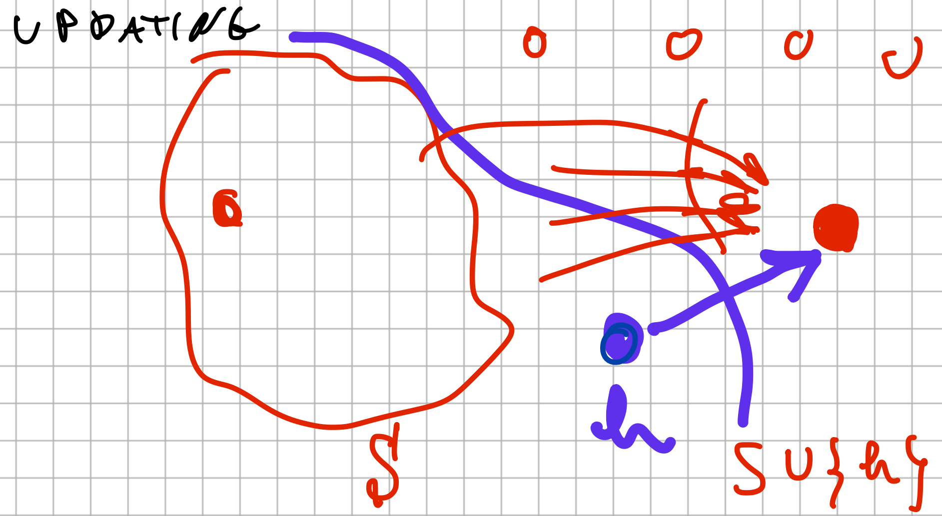
$\sim n$ iterations before reading node t $\rightarrow$ $\boxed{O(n^3) \text{ time}}$

DIJKSTRA'S TRICK $\Rightarrow$ $\boxed{O(n^2) \text{ time}}$



$L_j = \min\{L_i + c_{ij} : \forall i \in S\}$

$L_\ell := \underbrace{\min\{L_j : j \notin S\}}_{O(n)}$



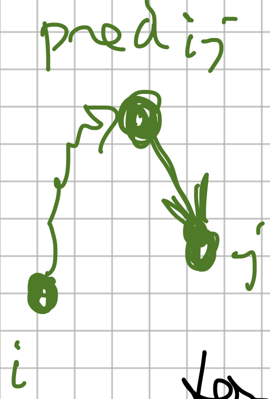
$$\underbrace{L_j\,,\ L_h + c_{hj}}_{\min}$$

→ UPDATE ALL L_j's in $O(n)$ time

⇒ $O(n^2)$ in total !

$$\boxed{\text{CASE WHERE } c_{ij} < 0 \text{ for some arcs } (i,j) \in A}$$

FLOYD-WARSHALL $O(n^3)$

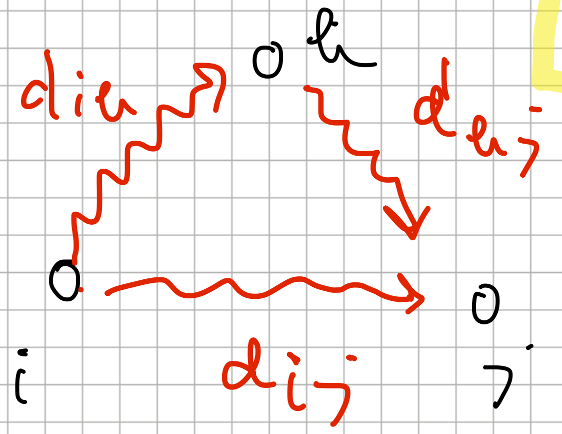


d_{ij} = cost shortest path $i \to j$

for all $(i,j) \in A$

$$d = \begin{array}{c|ccc} & 1 & j & n \\ \hline 1 & & & \\ 2 & & & \\ i & & d_{ij} & \\ n & & & \end{array}$$

pred$_{ij}$ = predecessor of j in the shortest path from i

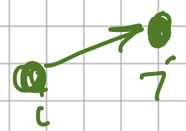


TRIANGULARITY PROP.

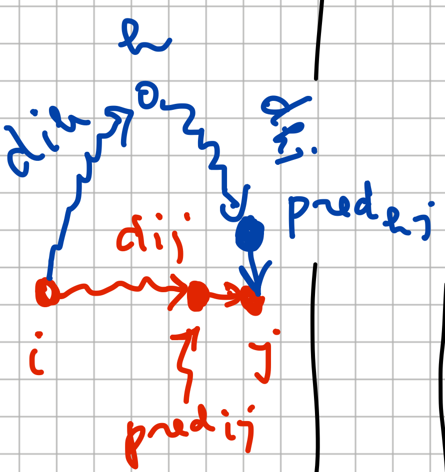
$$\underbrace{d_{ij}}_{\substack{\text{best}\\ \text{path}\\ i\to j}} \le \underbrace{d_{ik}+d_{kj}}_{\substack{\text{alternative}\\ \text{path } i\to j}}$$

$\boxed{\text{FW - alg.}}$ $O(n^3)$ time

1. $d_{ij} := c_{ij}$
   $pred_{ij} := i$ $\Big\}\ \forall\, i,j = 1,\dots,n$

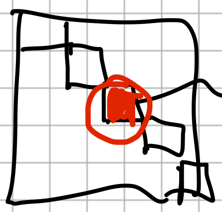


2. **for** $k = 1$ **to** n **do**
   - **for** $i = 1$ **to** n **do**
     - **for** $j = 1$ **to** n **do**
       - **if** $d_{ij} > d_{ik} + d_{kj}$ **then**
         - $d_{ij} := d_{ik} + d_{kj}$;
         - $pred_{ij} := pred_{kj}$;
       - **fi**
     - **rof**
   - **rof**



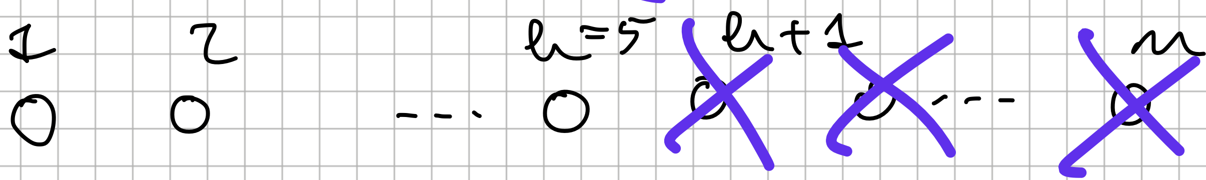
```
for i := 1 to n do
    if d_ii < 0 then STOP
    rof
rof
```

d

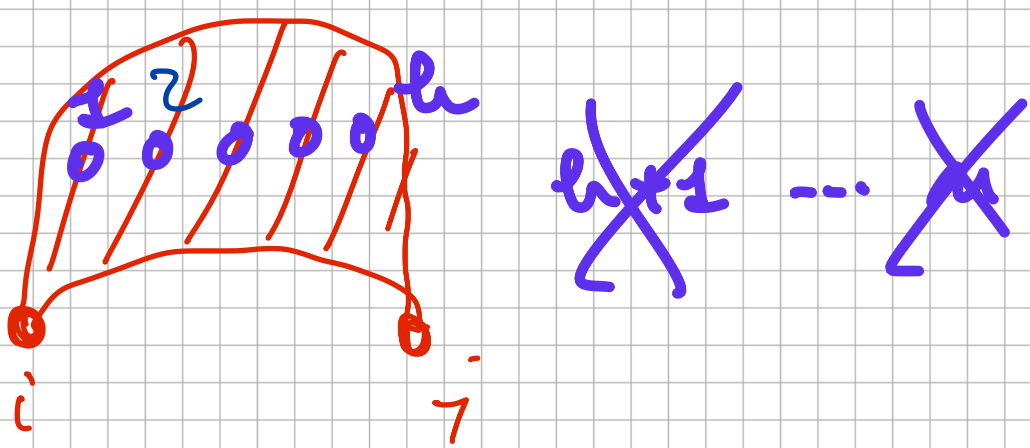


d_{ii} = shortest distance $i \to i$

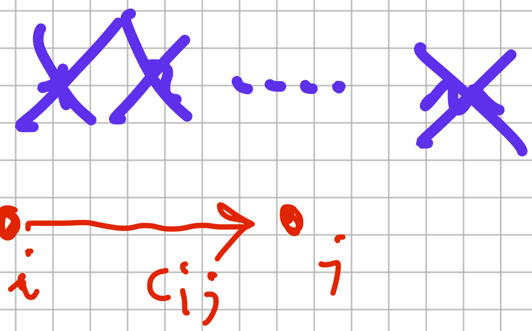
PROOF OF CORRECTNESS



d_{ij} = cost of the shortest path $i \to j$ when nodes $k+1, \dots, n$ are removed



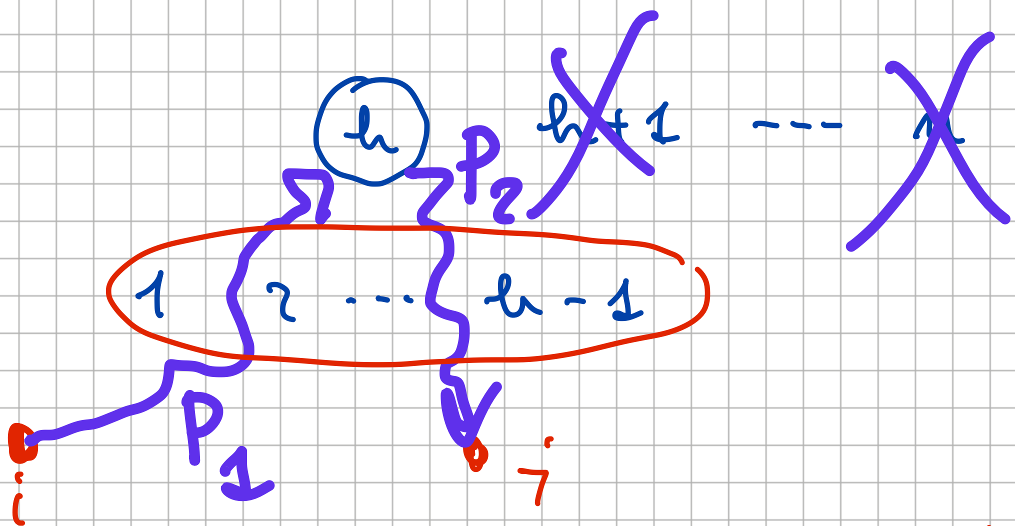
- TRUE at step 1 "initialization"



$d_{ij} = c_{ij}$

- TRUE at iterat. $k-1$

$\Rightarrow$ TRUE at iter. k



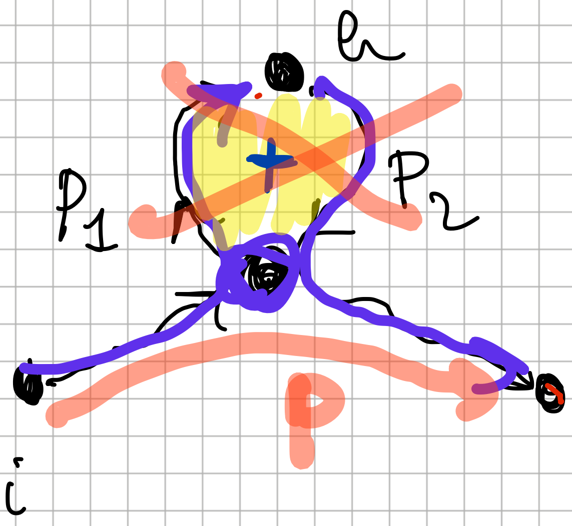
* P_{ij} does not visit node k → best such path has a cost d_{ij}

* P_{ij} goes through node k

$$\underbrace{d_{ik}}_{\text{best } P_1} + \underbrace{d_{kj}}_{\text{best } P_2}$$

BUT... ARE THE PATHS <u>SIMPLE</u>? □

YES!



~~h+1~~ --- ~~h~~

$P_1 \cup P_2 =$ P ∪ cycle

$$\boxed{d_{ih} + d_{hj} < d_{ij}}$$

cannot happen! indeed:

→ path P:

$$d_{ij} \le \text{cost}(P) \le \underbrace{\text{cost}(P_1)}_{d_{ih}} + \underbrace{\text{cost}(P_2)}_{d_{hj}}$$

↑ does not go through node h

⇒ IN THIS SITUATION, d_{ij} IS <u>NOT</u> updated!