EE360C ALGORITHMS - FALL 2018 - October 9 - 11-12:30

our greedy solution: $i_1, \dots, i_k$ (earliest finish time first)

any other solution: $j_1, \dots, j_m$ } any set of mutually compatible jobs.

want to prove that $m \le k$.

First, we prove that greedy "stays ahead".

$\boxed{\text{Claim}}$ For all indices $r \le k$, we have $f(i_r) \le f(j_r)$

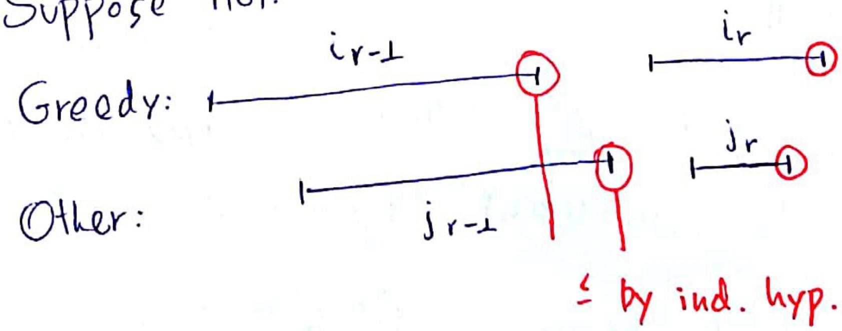
$\boxed{\text{Pf}}$: By induction on r.

Base: $r=1$, clearly $i_1 \le j_1$ by greedy criterion. ✓

Ind. Hyp.: assume true for $r-1$

Want to prove true for r.

Suppose not.



Not possible because $f(i_{r-1}) \le f(j_{r-1}) \le s(j_r)$

So j_r is compatible with i_{r-1} and greedy would have chosen it.

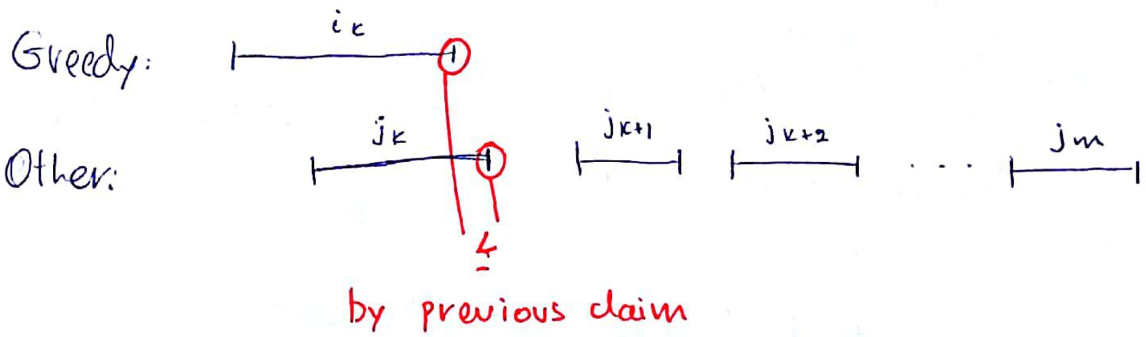
□

$f(k)$: finish time of job k

$s(k)$: start time of job k

Claim: $m \le k$

Pf: Suppose not: $m > k$.



Not possible since $j_{k+1} \dots j_m$ are compatible with each other and with i_k and greedy would have added it.

Greedy Alg.: Dijkstra's Algorithm

$\boxed{\text{Shortest Paths}}$

- **Weighted graph**: each edge (u,v) has a weight $w(u,v)$ (aka length)

  Here $w(u,v) \in \mathbb{R}^+$

- **The weight of a path p** written as $w(p)$ is the sum of the weight of its constituent edges.

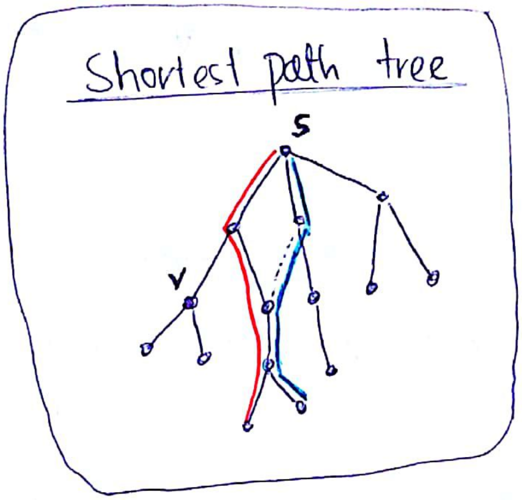
- **Shortest path weight** from s to v:

$$\delta(s,v) = \begin{cases} \min\limits_{\substack{\text{paths } p \\ \text{from } s \text{ to } v}} \{w(p)\} & , \text{ if } v \text{ is reachable from } s \\ \infty & , \text{ otherwise} \end{cases}$$

Problem: Given start node s, compute shortest paths to all other nodes in the graph.

Output: for every node v: $v.d$ = length of shortest path weight from s to v.

$v.\pi$ = parent of node v in shortest path tree.



Dijkstra's Algorithm

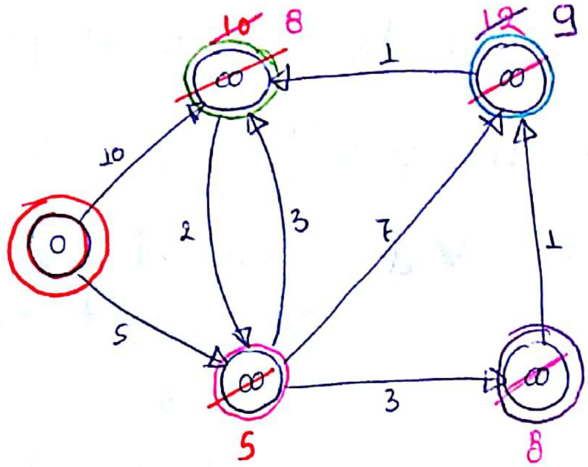
1. Initialize (G, s)
2. Q := all vertices
3. While $Q \neq \emptyset$
4. v := Extract_min (Q) // find smallest v.d and remove the node v from Q
5. for each edge (v,u)
6. RELAX (v,u)

Initialize (G, s)

for each vertex v:
$v.d := \infty$
$v.\pi$ = nil
$s.d = 0$.

RELAX (v,u)

if $u.d > v.d + w(v,u)$:
$u.d := v.d + w(v,u)$
$u.\pi := v$



First Loop

Second Loop

Third Loop

Fourth Loop

Fifth Loop

Q is empty!