10/9/18

Interval Scheduling

- Greedy algorithms - builds up solution in small steps, look it myopically.
- Uses a technique called "stays ahead" (Interval Scheduli

→ Our greedy solution: $i_1, i_2, i_3, \ldots, i_k$

Another solution: $j_1, j_2, j_3, \ldots, j_m$ → mutually compatible courses.

Want to prove that $m \le k$

↳ Prove it with out "stays ahead" algorithm.

Claim: For all indices $r \le k$:

We have $f(i_r) \le f(j_r)$

finish time

Proof: By induction on r:

Base Case: $r = 1$; $f(i_1) \le f(j_1)$

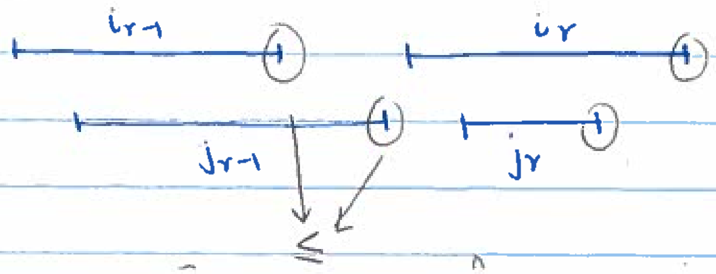
Induction Hypothesis: Assume true for $r-1$,

$f(i_{r-1}) \le f(j_{r-1})$

Want to show $f(i_r) \le f(j_r)$

Suppose not,

greedy



j_r is compatible with i_{r-1} because

$$f(i_{r-1}) \leq f(j_{r-1}) \leq s(j_r)$$

So greedy would have added it $\Rightarrow\Leftarrow$

$\Rightarrow$ SHORTEST PATH: DIJKSTRA'S ALGORITHM

- what are we trying to solve?
  - Think Google Maps
  - Want to find the shortest path

- Weighted graph - each edge (u,v) is assigned $\underbrace{\text{weight}}_{\text{length}}$ $w(u,v)$

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

- The weight $w(p)$ of path p is the sum of its constituent edges' weights.

- Shortest path weight from s to v

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

$\rightarrow$ Input: weighted graph G (Directed)*

Start node s

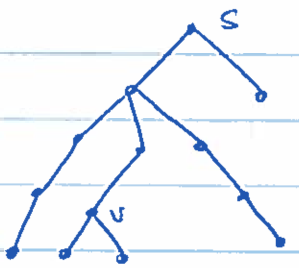
→ Algorithm computes shortest path from one node to all other nodes!

Output: for each node v

$v.d$ = shortest path length/weight from s to v

$v.\pi$ = parent of v in the shortest path tre

- A way to encode this,



Shortest Path Tree

Need to return parent of each node to build this tre

Algorithm:

INITIALIZE (G, s) — for each vertex v: $v.d := \infty$, $v.\pi := nil$; $s.d := 0$

A datastructure that we maintain

$Q :=$ all vertices

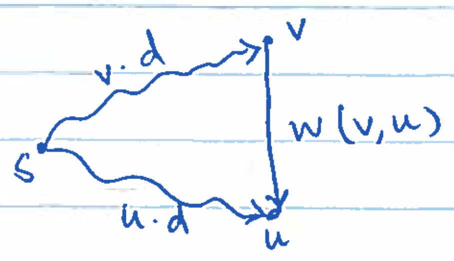
vertices for which we haven't yet determined the shortest path.

while $Q \neq \phi$

$v :=$ Extract-Min (Q) — Find the minimum v.c Q and remove v from

for each edge (v, u):

RELAX (v, u)



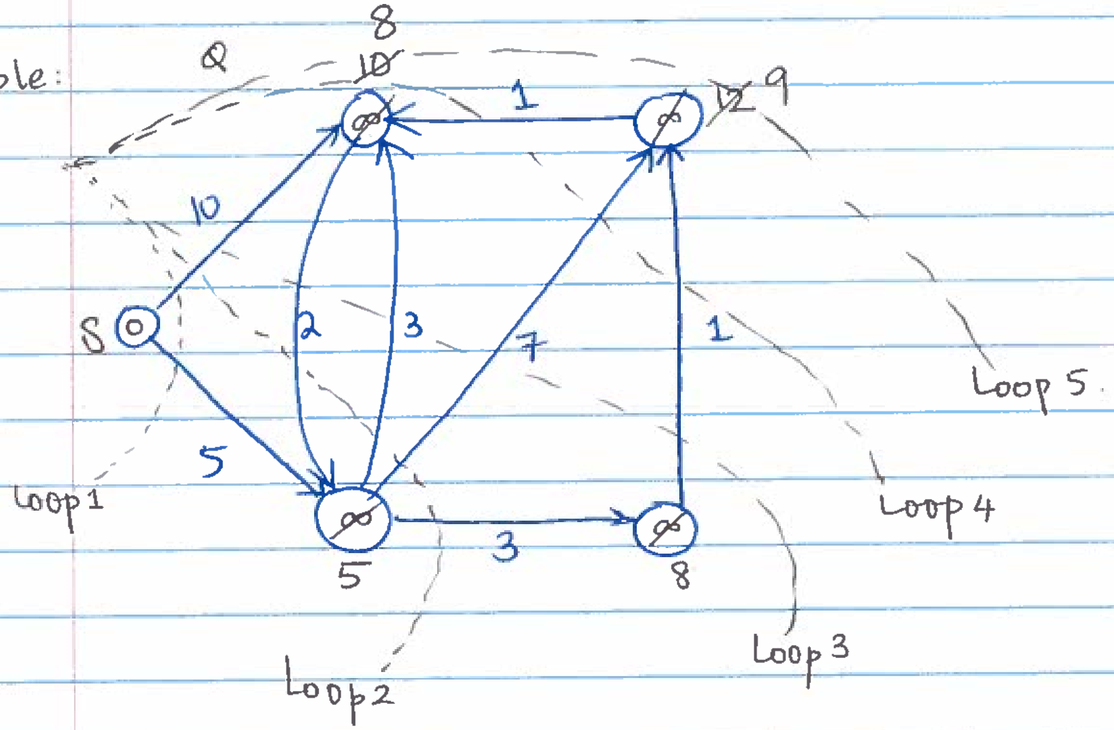
if $u.d > v.d + w(v,u)$

(New shorter path discovere

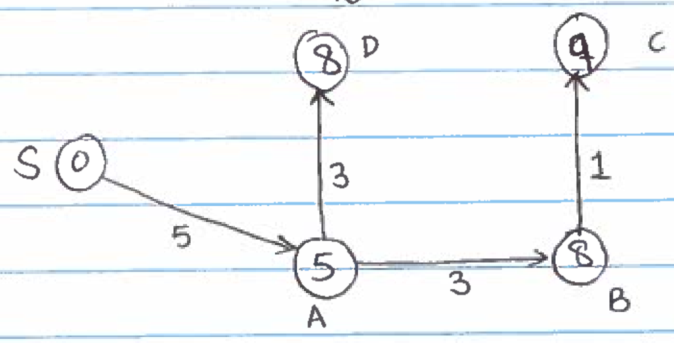
then: $u.d := v.d + w(v,$

$u.\pi := v$

ample:



Track the parents also!



$A \cdot \pi = S$	$A \cdot d = 5$	
$D \cdot \pi = A$	$D \cdot d = 8$	
$B \cdot \pi = A$	$B \cdot d = 8$	
$C \cdot \pi = B$	$C \cdot d = 9$