Nov 30

Shortest path problem

Input: • Directed $G=(V,E)$

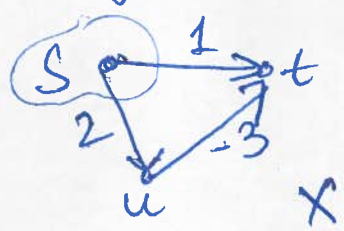
$\forall e \in E$, cost c_e (can be <0)

but NO $-$ve cost cycle.

• $t \in V$

Output: $\forall s \in V$, output a shortest s-t path

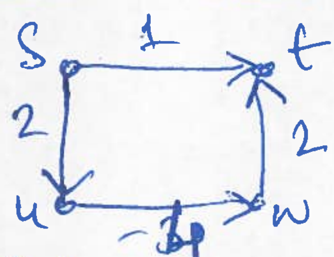
Attempt 1: Run Dijkstra's as is: You start Dijkstra on s.



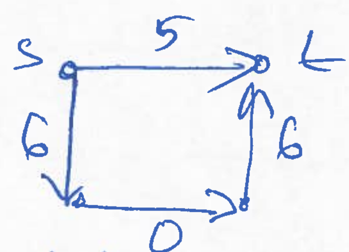
$d'(t)=1$, $d'(u)=2$ will assert dist from s to t is 1. X as s-u-t is shortest path of total cost -1.

Attempt 2: Add a large enough # to all edge costs so the new edge costs ≥ 0 & run Dijkstra

Counter-example:



Add 4 to all $\longrightarrow$



shortest s-t path: s-u-w-t shortest s-t path: (s,t)

No known greedy algo.

Bellman-Ford

Prop: If G has no $-$ve cost cycle $\Rightarrow$ $\forall s,t$ $\exists$ a shortest s-t path that is simple (i.e. no node repeats)

Pf (idea): By contradiction. Say there is no simple shortest s-t path.

throw away the cycle.

If the cycle T let $P' = P \setminus T$

$$c(P') = c(P) - c(T)$$

$$\underset{\text{as } c(T) \geq 0}{\longrightarrow} \leq c(P)$$

Repeat if ~~necessary~~ necessary to end up with a simple s-t path that as same cost as $c(P)$

COROLLARY: $\forall\, s,t$ $\exists$ a shortest path $\leq n-1$ edges.

Assume: Only interested is cost of a shortest path

Attempt 1: Let $OPT(S)$ be cost of a shortest s-t path

if a shortest s-t path uses the edge (s,u)

$$OPT(s) = c(s,u) + OPT(u)$$

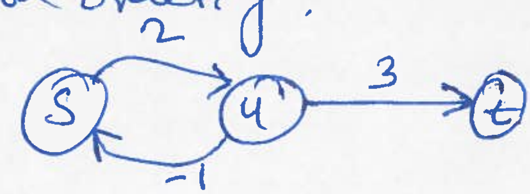
In general

$$OPT(S) = \min_{(s,u) \in E} c(s,u) + OPT(u)$$

(1) n subproblem ✓

(2) ↓ recursive formula

(3) Good ordering?



$$(*)\quad OPT(S) = \cancel{c(s,u)}\ 2 + OPT(u)$$

$$OPT(u) = \min\{3 + OPT(t),\ -1 + \underset{(**)}{OPT(S)}\}$$

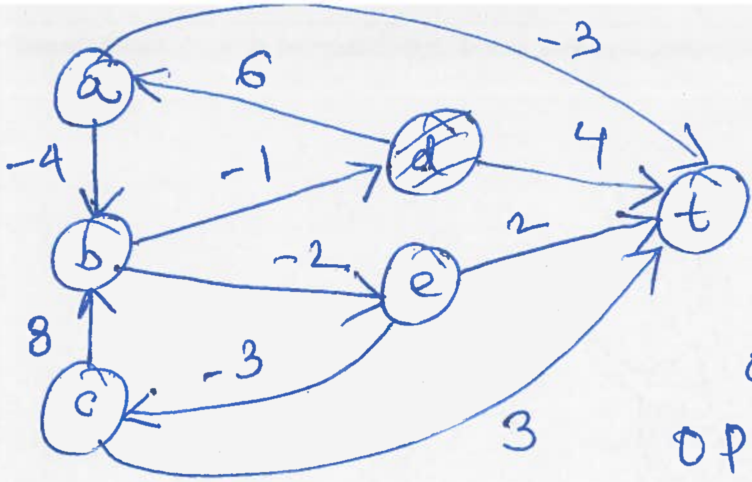
Problem: $OPT(S)$ depends on $OPT(u)$
AND $OPT(u)$ — $OPT(S)$ } no hope of a total order.

Q: How do we fix this?

Q: Any difference in $(*)$ & $(**)$ if we somehow also take ~~use~~ # edges into account.

Idea: $OPT(u,i)$ = cost of shortest u-t path with $\leq i$ edges.

↑ different from subset sum since this parameter is not explicitly part of the problem.



focus on d

$OPT(d,0) = \infty$ (as $d \neq t$

$OPT(d,1) = 4$ (d-t)

$OPT(d,2) = 6-3=3$ (d-a-

$OPT(d,3) = 3$ (d-a-t)

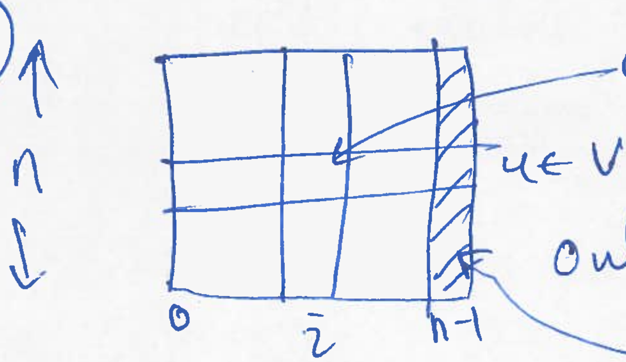
$OPT(d,4) = 6-4-2+2 = 2$ (d-a-b-e

$OPT(d,5) = 6-4-2-3+3=0$ (d-a-b-e-c-t)

$\underbrace{OPT(d,6) = 0 \quad OPT(d,100) = 0}_{\text{true by COROLLARY.}}$

Note: (1) $OPT(u,i) \leq OPT(u,i-1)$ (by defn)

(2)



subproble $= n^2$

output: $OPT(s, n-1)$