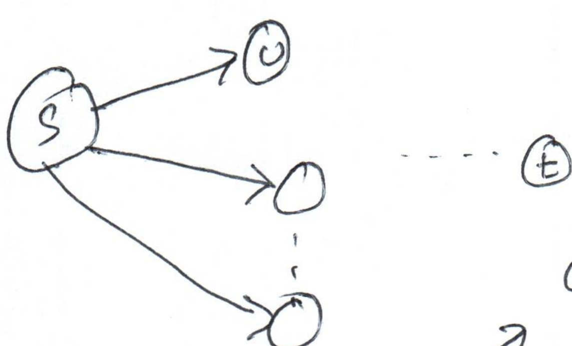
Apr 28

Intuition: If we use edge (s,v) when we compute a shortest s-t path, we do not want to re-use (s,v) when we compute a shortest v-t path.

<u>Solution</u>: Introduce an implicit parameter to define the subproblems.

<u>Attempt 4</u>: (·) $OPT(s, E')$ = cost of a shortest s-t path using only edges in E'.

$$OPT(s, E') = c_{s,v} + OPT(v, E' \setminus \{(s,v)\})$$



In general,

$$OPT(s, E') = \min_{\substack{w:\\ (s,w) \in E}} \{ c_{s,w} + OPT(w, E' \setminus \{(s,w)\}) \}$$

(·) <u>Recurrence</u>

(·) ordering among subproblems: order according to the size of E', i.e., $|E'|$.

(·) # subproblems: $n \cdot 2^m$ ← not polynomial

| # of subsets of a set of size n: 2^n

ISSUE: ~~W~~ ~~we~~ ~~no~~ Keeping track of E' introduces redundancy. We do not have to keep track of the edges that we will not use.

Intuition: Instead of keeping track of all edges in E', keep track of (number) edges that will be a shortest path.

Attempt 5: Bellman-Ford

$OPT(s, i) \doteq$ cost of a shortest s-t path using $\leq i$ edges.

PROP: If G has no negative cycles $\implies \forall s$ $\exists$ a shortest simple path.

Pf. (idea):

→ removing cycle/repeated vertices cannot increase the cost of shortest path.

$OPT(s, i)$

$\forall s \in V$

subproblems:

$n \cdot n = n^2$ * polynomially many subproblems.

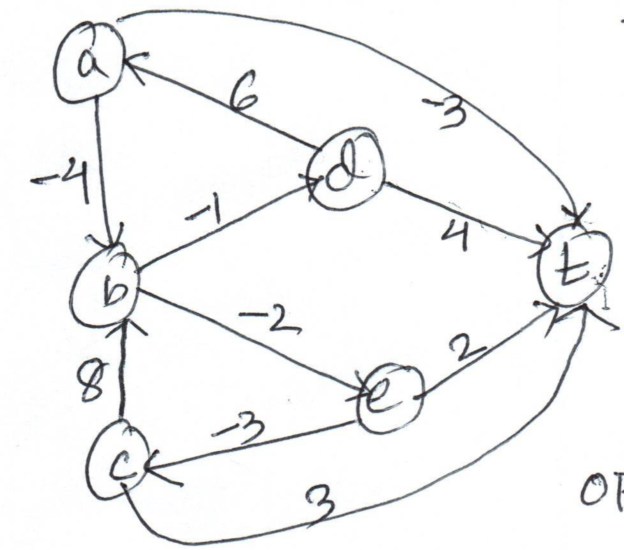
Q. What final OPT value(s) do we want to compute?

A: $OPT(s, n-1)$ $\forall s \in V$

By PROP, s has a shortest path that is simple.

$\implies$ has $\leq n-1$ edges.

Goal: Compute $OPT(s, n-1)$ $\forall s \in V$.



focus on vertex d.

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

$OPT(d,1) = 4$ [d, t]

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

$OPT(d,3) = OPT(d,2) = 3$ [d, a, t]

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

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

$OPT(d,6) = OPT(d,5) = 0$ [as $n=6$, $\Rightarrow n-1 = 5$]

$OPT(d,7) = OPT(d,8) = \cdots = 0$

By PROP

$OPT(s,i)$ = cost of shortest s-t path using $\leq i$ edges.

Recurrence:

$OPT(t,0) = 0$ $OPT(u,0) = \infty$ $\forall u \neq t$

$OPT(u,i)$ for $i > 0$.

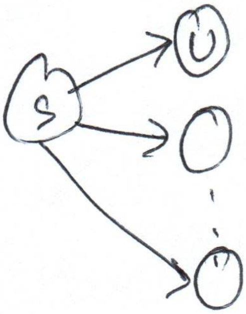
Case 1: $\exists$ a shortest s-t path that uses $\leq i-1$ edges.

$\Rightarrow OPT(u,i) = OPT(u, i-1)$

Case 2: All shortest s-t paths use exactly i edges.

$\Rightarrow$

$OPT(s,i) = c_{s,v} + OPT(v,i-1)$



In general,

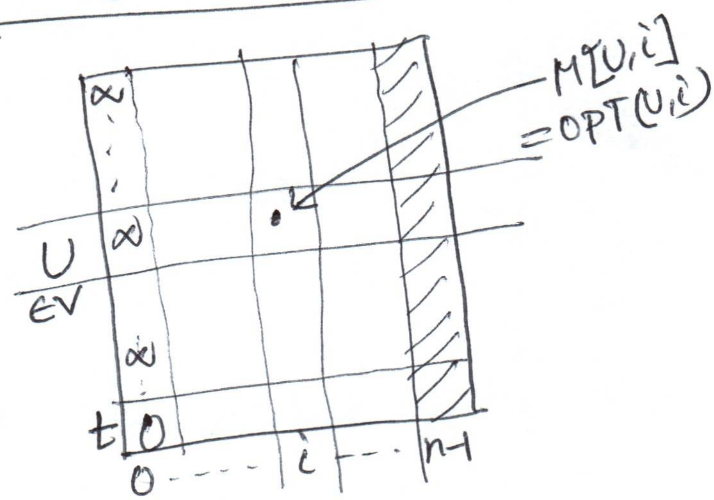
$$opt(s,i) = \min_{\substack{w: \\ (s,w)\in E}} \{ c_{s,w} + \min_{\substack{w: \\ (s,w)\in E}}$$

$$OPT(s,i) = \min_{\substack{w: \\ (s,w)\in E}} \{ c_{s,w} + OPT(w, i-1) \}$$

Overall:

$$OPT(v,i) = \min\{ OPT(v,i-1), \min_{\substack{w: \\ (v,w)\in E}} \{ c_{v,w} + OPT(w,i-1) \} \}$$

(*) Compute $M[s, n-1]$ for $\forall s \in V$



Bellman ford

0. Allocate $n \times n$ matrix M
1. $M[t,0] \leftarrow 0$, $M[v,0] \leftarrow \infty \ \forall v \neq t$
2. for $i = 1 \dots n$

   for $v \in V$

   $M[v,i] = \min\{ M[v,i-1], \min_{\substack{w: \\ (v,w)\in E}} c_{v,w} + M[w,i-1] \}$
3. return $M[s, n-1] \ \forall s \in V$

ordering: column i depends on " $i-1$

compute $M \Rightarrow L \rightarrow R$ column by column.

$$M[v,i] = \min\{ M[v,i-1], \min_{\substack{w: \\ (v,w)\in E}} \{ c_{v,w} + M[w,i-1] \} \}$$