② Recursive formulation $(s \neq t)$ Attempt 3: $OPT(s)$.
= cost of shortest s-t path

Now

$\cdot t$

If shortest s-t path uses edge (s,u)

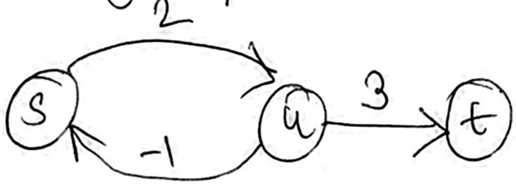
$$OPT(s) = c_{(s,u)} + OPT(u)$$

More generally

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

③ An ordering among problem?

This fails



$$OPT(s) = 2 + OPT(u)$$

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

→ cycle in the dependence ⇒ X total ordering.

Solution: Introduce an (implicit) parameter to the sub-problem definition (not needed earlier)

→ have the parameter keep track of how "close" we are to t

Attempt 4: $OPT(s, E')$ $s \in V$, $E' \subseteq E$

↳ cost of shortest s-t path only using edges in E'.

Recurrence: $OPT(s, E') = \min\limits_{w:(s,w)\in E} \{ c_{(s,w)} + OPT(w, E' \setminus \{(s,w)\}) \}$

Ordering among subproblems: Order in increasing size of $|E'|$

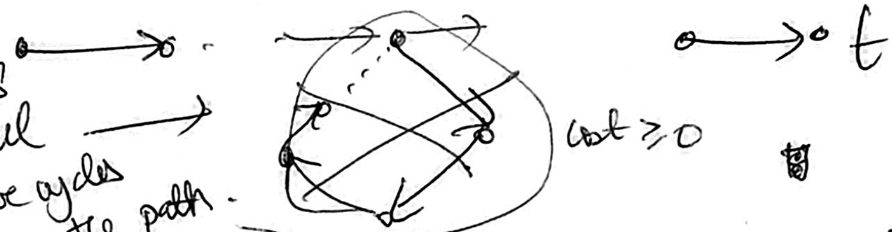
Number of sub-problems: $n \cdot 2^m \leftarrow \# E' \subseteq E$
X not poly.

Attempt 5: Bellman-Ford Algo

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

PROP: If G has no negative cycle $\Rightarrow \forall s\ \exists$ a shortest s-t path that is simple.

Pf(idea): If not



Drop all negative cycles from the path.

∎

Recall: A simple path has $\leq n-1$ edges is it. $\Rightarrow \exists$ a shortest s-t path of len $\leq n-1$

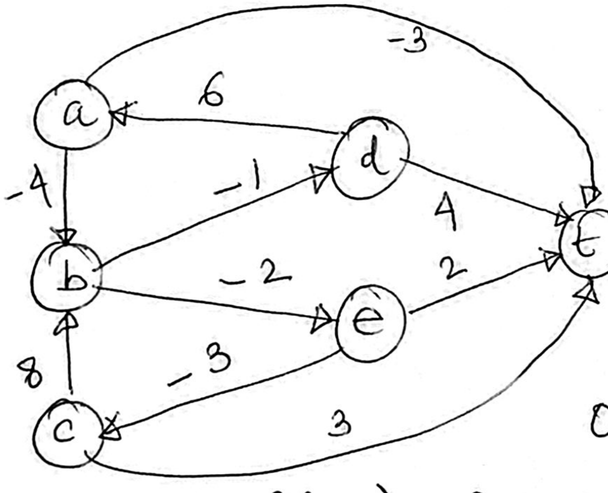
Q: $OPT(s, n-1)$

All set if we can compute $OPT(s,i)\ \forall s \in V$, $\forall\ 0 \leq i \leq n-1$

Q: How many subproblems? $n \cdot n = n^2$

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

$n=6$ Let's focus on d.



$OPT(d,0) = \infty$ [d ≠ t]

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

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

$OPT(d,3) = 3$ [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]

$OPT(d,6) = 0 = OPT(d,7) \ldots$

Recall: $OPT(s,i)$ = cost of shortest s-t path of len $\leq i$

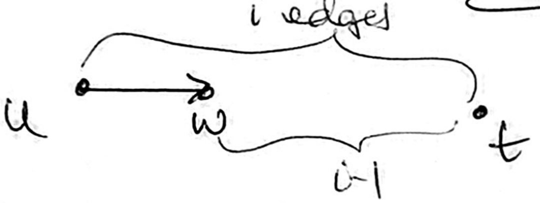
Recurrence: $OPT(t,0) = 0$, $OPT(u,0) = \infty \quad \forall u \neq t$

Consider $OPT(u,i) \quad i > 0$

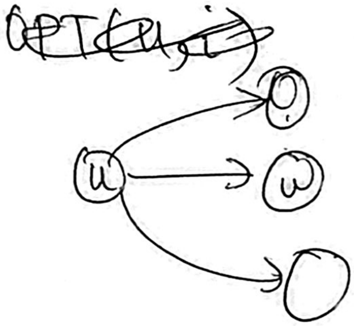
Case 1: $\exists$ a shortest u-t path with $\leq i$ edges that actually uses $\leq i-1$ edges

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

Case 2: All shortest u-t path of len $\leq i$ uses *exactly* i edges



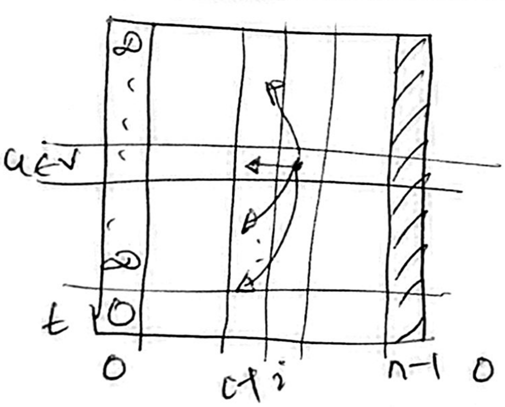
$$OPT(u,i) = c_{(u,w)} + OPT(w,i-1)$$



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

OVERALL: $i > 0$

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



$M[u,i] = OPT(u,i)$

Output: $M[s,n-1] \quad \forall s \in V$

③ ordering

$O(n^3)$ follows $\rightarrow O(nm)$

Bellman-Ford

0. Allocate an $n \times n$ matrix M } $O(n^2)$
1. $M[t,0] \leftarrow 0$, $M[u,0] = \infty \quad \forall u \neq t$ } $O(n)$
2. for $i = 1..n-1$ $\quad O(n)$
   n^2 iterate for $u \in V$
   $M[u,i] \leftarrow \min \{ M[u,i-1], \min_{w:(u,w)\in E} \{ c_{(u,w)} + M[\ldots$
3. Return $M[s,n-1] \quad \forall s \in V$ } $O(n)$