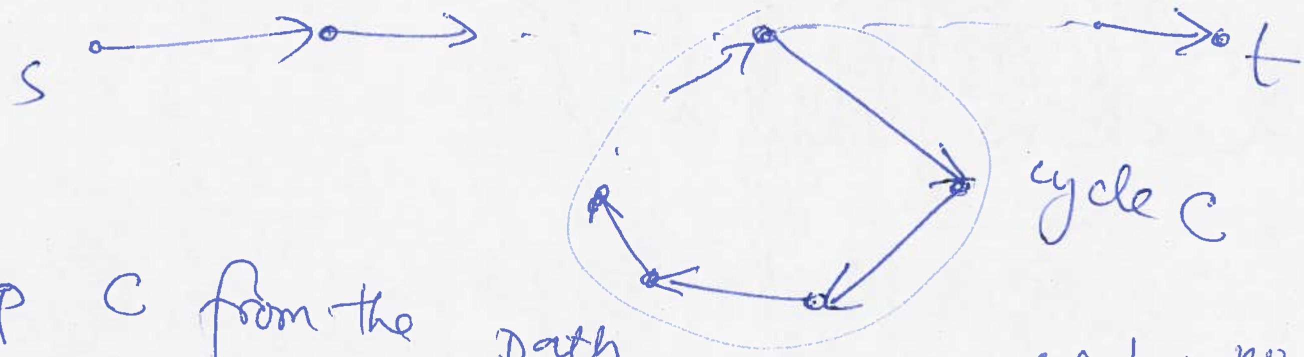
Nov 19, 2014

Prop: If G does not have a −ve cycle, then $\forall$ s, t, $\exists$ a shortest s-t path that is simple (ie no repeated vertex)

Pf:



Drop C from the path

→ s & t still connected

→ but cost of path does not go up (G has no −ve cycle)

$\Rightarrow \forall$ s, t $\exists$ a shortest s-t path with $\leq n-1$ edges

---

Assume: Compute <u>cost</u> of shortest s-t paths

OPT (u, i) — : cost of the shortest u-t path that uses $\leq i$ edges.

$u \in V$

$0 \leq i \leq n-1$

Q: cost of the shortest s-t path

↳ OPT(s, n−1) (from Prop).

~~OPT(u, i+1)~~

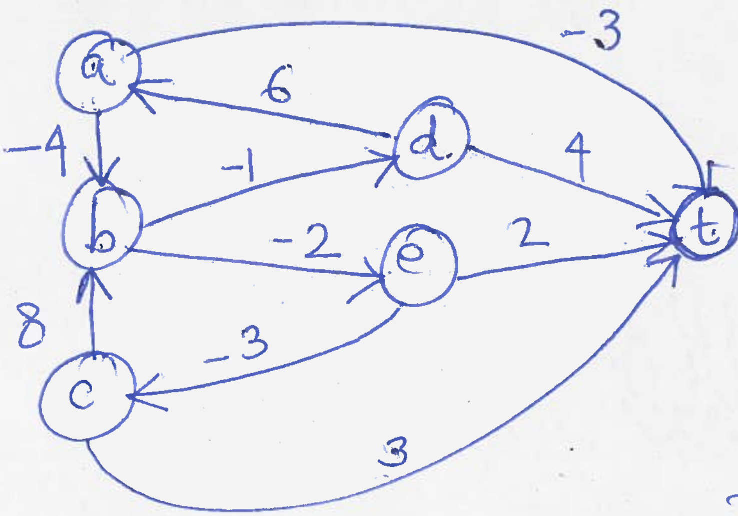
$OPT(u, i+1) \leq OPT(u, i)$

Note if we compute OPT(u, i) $\forall u \in V$, $\forall 0 \leq i \leq n-1$ ⇒ done

sub-problems = $n \times n = \boxed{n^2}$ ← poly many sub-problems

(n = |V|; n = # distinct i)

Example: $n = 6$



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

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

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

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

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

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

$OPT(d,6) = 0$

$\dots$ $OPT(d,100) = 0$

optimal d-t path cost as $n-1=5$

Goal #2 (for Dynamic Prog) $\rightarrow$ Compute a recursive formula

---

$OPT(u,i)$ $\quad i \geq 1$

$OPT(t,0) = 0$

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

say

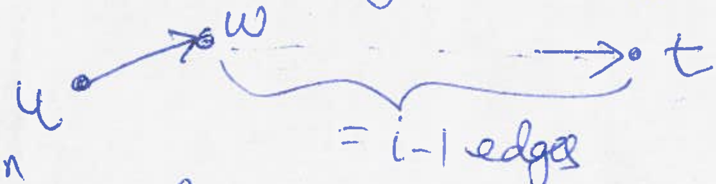
$$\underbrace{u \rightarrow w \; \cdots \cdots \rightarrow t}_{\leq i \text{ edges}}$$

Case 1: $\exists$ a short u-t path ($\leq i$ edges) that uses $\leq i-1$ edges

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

(e.g $OPT(d,3) = OPT(d,2)$ above)

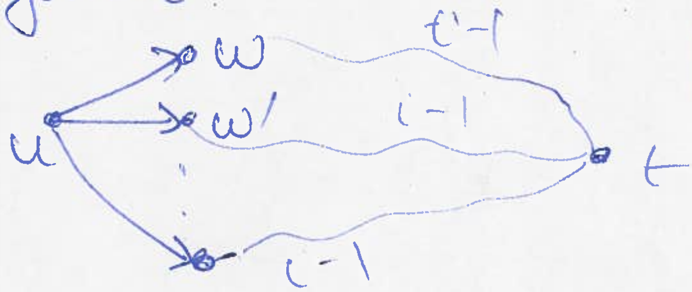
Case 2: All shortest u-t paths ($\le i$ edges) has exactly i edges



$\boxed{G=(V,E)}$

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

in general:



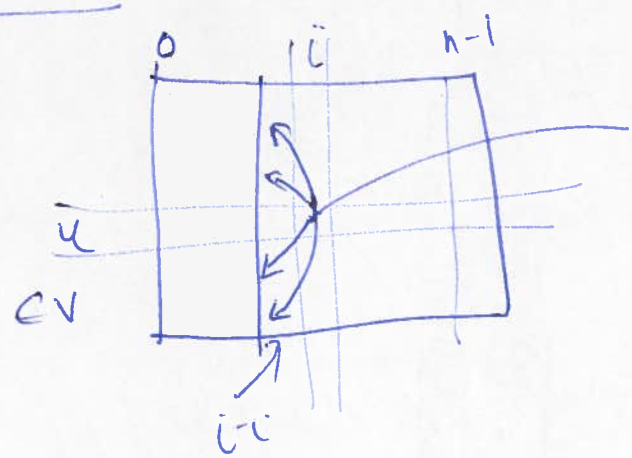
$$\boxed{OPT(u,i) = \min_{(u,w)\in E} c_{u,w} + OPT(w,i-1)}$$

Overall

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

Goal 2: recursive formulation ✓

Goal 3: natural order among subprobs.



Order: Build column by column.