Nov 16

Subset Sum $(w_1, \dots, w_n; W)$

0. Allocate a $(W+1) \times (n+1)$ matrix M
1. $M[B, 0] \leftarrow 0 \quad \forall\, 0 \le B \le W$
2. for $j = 1 \dots n$
   - for $B = 0 \dots W$
     - if $w_j > B$
       - $M[B, j] \leftarrow M[B, j-1]$
     - else
       - $M[B, j] \leftarrow \max\{ w_j + M[B - w_j, j-1], M[B, j-1] \}$
3. Return $M[\underline{W, n}]$

Prove by induction on j, $\forall\, j, B$ $M[B, j] = OPT(B, j)$

$O(nW)$ runtime $\Rightarrow$ pseudo poly runtime $\rightarrow$ $W = poly(n)$ $\Rightarrow$ poly runtime.

Subset sum problem: Input size $N = n + \log W$

If $W = 2^n \Rightarrow N = \Theta(n)$ but runtime $O(n \cdot 2^n)$

Run of algo $\quad n = 3,\ w_1 = 1,\ w_2 = 2,\ w_3 = 2,\ W = 3$

B \ j	0	1	2	3
3	0	1	3	3
2	0	1	2	2
1	0	1	1	1
0	0	0	0	0

$M[1,1] \xleftarrow{1 = w_1 \not> B = 1} \max\{ w_1 + M[1-1, 0], M[1,0] \} = \{1+0, 0\} = 1$

$M[2,1] \xleftarrow{1 = w_1 \not> B = 2} \max\{ w_1 + M[2-1, 0], M[2,0] \} = \max\{1+0, 0\} = 1$

$OPT(0, j) = 0$

$M[3,1] \xleftarrow{1 = w_1 \le B = 3} \max\{ w_1 + M[3-1, 0], M[3,0] \} = \max\{1+0, 0\} = 1$

$M[1,2] \xleftarrow{w_2 = 2 > B = 1} M[1,1] = 1$

$M[2,2] \xleftarrow{w_2 = 2 \le B = 2} \max\{ w_2 + M[2-2, 0], M[2,1] \} = \max\{2+0, 1\} = 2$

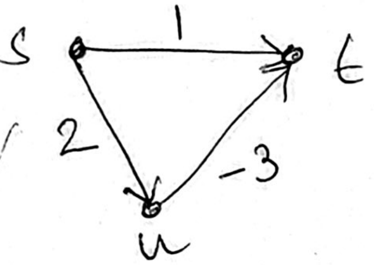
$M[3,2] \xleftarrow{w_2 = 2 \le B = 3} \max\{ w_2 + M[3-2, 1], M[3,1] \} = \max\{2+1, 1\} = 3$

Shortest Path Problem

Input: (•) Directed graph $G = (V, E)$
$\forall e \in E$, c_e ($c_e < 0$ is allowed)
BUT no negative cycle.
(•) $t \in V$

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

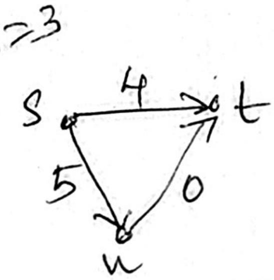
Attempt 1: Run Dijkstra



shortest s-t path: s, u, t
but Dijkstra picks s, t

Attempt 2: Ashton's algo: Add some $\Delta > 0$ to all edge costs so that new costs ≥ 0
→ run Dijkstra

$\Delta = 3$



← shortest s-t path here s, t
→ Reduction doesn't work

Note: There is no known greedy or Divide & Conquer algo for this problem.

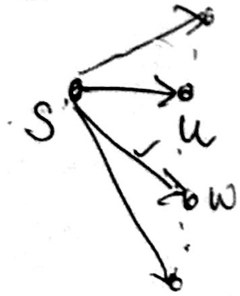
ASSUME (for now) Only care about cost of shortest s-t paths.

Goal: Design a Dynamic Program

Attempt 3: $OPT(s)$ = cost of shortest s-t path.

① How many sub-problems? $OPT(s)$ $\forall s \in V \Rightarrow n$

② Recursive formulation $(s \neq t)$



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

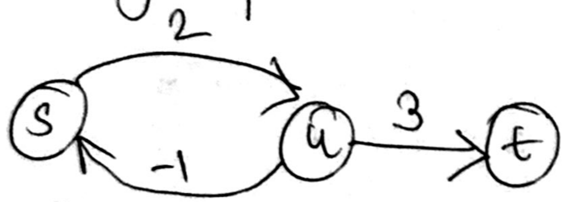
$$OPT(s) = c_{(s,u)} + OPT(u)$$

More generally

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

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 ⇒ ✗ total ordering.