Nov 17

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

0. Allocate an $(W+1) \times (n+1)$ matrix M
1. $M[B,0] \leftarrow 0 \quad \forall\ 0 \le B \le W$
2. for $j = 1..n$

   for $B = 0..W$

   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 $M[W,n]$

$M[B,j] = OPT(B,j)$

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

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

$W = 2^n \rightarrow$ input size $= \Theta(n)$ runtime $= O(n \cdot 2^n)$

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

N				
3	0	1	3	3
2	0	1	2	2
1	0	1	1	1
0	0	0	0	0
	0	1	2	3

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

$= \max\{1+0, 0\} = 1$

$M[2,1] \xleftarrow{1 = w_1 < 2 = B} \max\{w_1 + M[2-1, 0], M[2,0]\}$

$= \max\{1+0, 0\} = 1$

$\ldots$ $M[3,1] \leftarrow \cancel{0}\ 1$

$\boxed{j=2}$

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

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

$= \max\{2+0, 1\} = 2$

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

$= \max\{2+1, 1\} = 3$

$\vdots$

Shortest path problem

Input: (•) Directed graph $G = (V, E)$

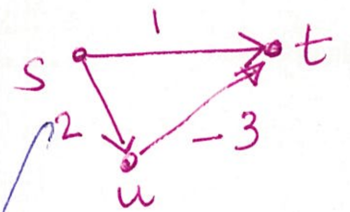
$\forall e \in E$, c_e (note: $c_e < 0$ is allowed)

BUT no negative cycle.

(•) $t \in V$

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

Attempts 1: Run Dijkstra of each $s \in V$



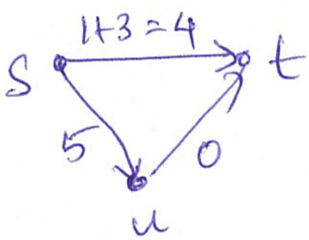
← shortest s-t path: s, u, t

But if you run Dijkstra

it picks s, t ✗

Attempt 2: Add some $\Delta > 0$ to all edges so that all new costs are ≥ 0

$\Delta = 3$



← shortest path w/ new costs $= s, t$ ⇒ Reduction doesn't work

---

→ No known greedy / divide & conquer algo for this problem

ASSUME: Only interested (for now) in cost of shortest s-t paths.

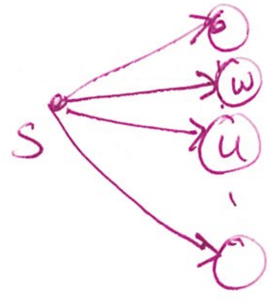
Goal: Design a dynamic program.

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

(1) Poly many subproblems: $OPT(s)$ $\forall s \in V$

n ✓

② Recurrence relation



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

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

In general

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

③ An ordering among sub-problems $\{ OPT(s) \mid s \in V \}$

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

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

Issue: OPT(s) depends on OPT(u)
OPT(u) ——— OPT(s) X

Solution: Introduce an implicit parameter to define your sub-problems.

Intuition: If we use edge (s,u) then when thinking of shortest u-t path we do not want to re-use (s,u)

Attempt 4: $OPT(s, E')$ → cost of shortest s-t path only using edges in E'.
$E' \subseteq E$

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