Subset Sum Problem:

Nov 27

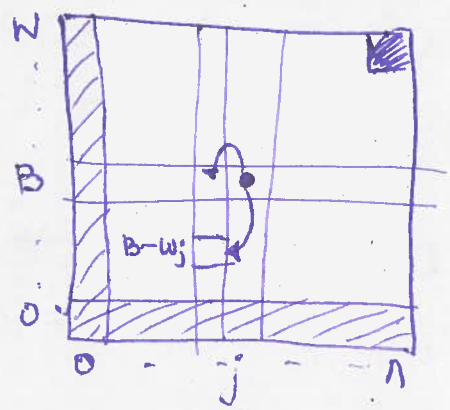
(*) If $w_j > B$
$$OPT(j, B) = OPT(j-1, B)$$
else
$$OPT(j,B) = \max\{OPT(j-1, B),\ w_j + OPT(j-1, B-w_j)\}$$

$j \in$ optimal for $(w_1, \dots, w_j, B-w_j)$

$j \notin O_j$ (optimal $w_1, \dots, w_j$, B)

Q1) Which entry of M are we interested in? (Recall: $w_1, \dots, w_n$; W)

A1) $M[W][n] = OPT(n, W)$



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

Q2): What should the initial values be?

A2: $M[0][j] = 0 \quad \forall\ 0 \le j \le n$

$M[B][0] = 0 \quad \forall\ 0 \le B\ W$

Q3) How many sub-problems do we have? **A3** # entries of M $= (n+1)(W+1)$

Note: if poly $\longrightarrow$ if $W = poly(n)$.

Q4) Recursive formula?

A4) See (*)

Q5) Ordering among sub-problems?

Obs: Only need to know $(j-1)$th column to compute column j

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

O(nW) runtime

0. Allocate $(W+1)(n+1)$ matrix M
1. $M[B][0] = 0 \quad \forall\ 0 \le B \le W$
2. for $j = 1..n$
   for $B = 0..W$
   O(1) { If $w_j > B$ then $M[B][j] = M[B][j-1]$
   else $M[B][j] = \max\{M[B][j-1],\ w_j + M[B-w_j][j-1]\}$
3. return $M[W][n]$.

Obs 1: Only need $O(W)$ space if we only care about $OPT(n,W)$.
But need $\Omega(nW)$ ——— want to compute the optimal set S.

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

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

$M[1][1] = \max\{M[1][0], w_1 + M[1-1][0]$
$(B=1, w_1=1) \quad = \max\{0, 1+0\} = 1$

$M[2][1] = \max\{M[2][0], w_1 + M[2-1][$
$(B=2, w_1=1) \quad = \max\{0, 1+0\} = 1$

$M[3][1] = \max\{M[3][0], w_1 + M[3-1][$
$(B=3, w_1=1) \quad = \max\{0, 1+0\} = 1$

$M[1][2] = M[1][1] = 1$
$(B=1, w_2=2)$

$\vdots$

$M[3][3] = \max\{M[3][2], 2 + M[\underbrace{B-w_3}_{1}][2]\} = \max\{3, 2+$
$(B=3, w_3=2)$ $\qquad B - w_3 = 3-2$
$= 3.$

Compute optimal subset:

If $w_j > B \Rightarrow j \notin$ optimal for $1..j; B$

else if $w_j + M[B-w_j][j-1] \geq M[B][j-1]$

$\Rightarrow j \in$ optimal for $1..j; B$

else

$j \notin$ ______

Ex: Recursively compute the optimal subset (like the weighted interval scheduling problem).

Knapsack problem: i^{th} item (w_i, v_i); W

Output $S \subseteq [n]$ s.t. $\sum_{i \in S} w_i \leq W$ AND $\max \sum_{i \in S} v_i$

(Subset sum problem: $v_i = w_i$)