Nov 17

Dynamic Program for Subset Sum Problem

<u>Goal:</u> Compute $w(S) = \sum_{j \in S} w_j$ for an optimal S

O_j: be an optimal solution for $1, \dots, j$

$OPT(j) = w(O_j)$

<u>Case 1:</u> $j \notin O_j$ $\quad OPT(j) = OPT(j-1)$

<u>Claim:</u> O_j is also optimal for $w_1 \dots w_{j-1}$

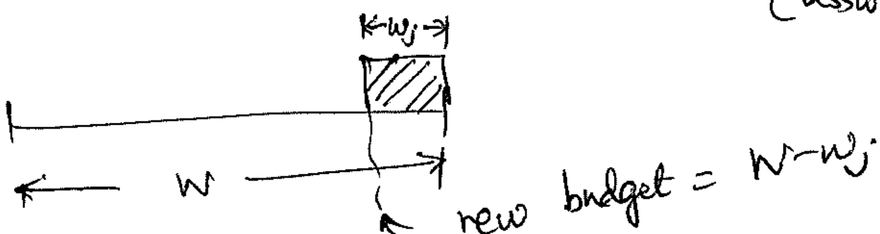
<u>Case 2:</u> $j \in O_j$

Q: What can we say about $O_j \setminus \{j\}$

<u>Hope:</u> $O_j \setminus \{j\}$ is also optimal $w_1, \dots, w_{j'}$ for some $j' < j$

If so, $OPT(j) = w_j + OPT(j')$

(assume $w_j \le W$)



<u>Solution:</u> Keep track of Budget <u>and</u> j

$OPT(B, j)$ = weight of an optimal solution for $w_1, \dots, w_j$ <u>and</u> budget B.

<u>Assume $j \in$ optimal</u> $(w_1, \dots, w_j; B)$ $\quad$ Assume $w_j \le B$

$\Rightarrow OPT(B, j) = w_j + OPT(B - w_j, j-1)$ —— (1)

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

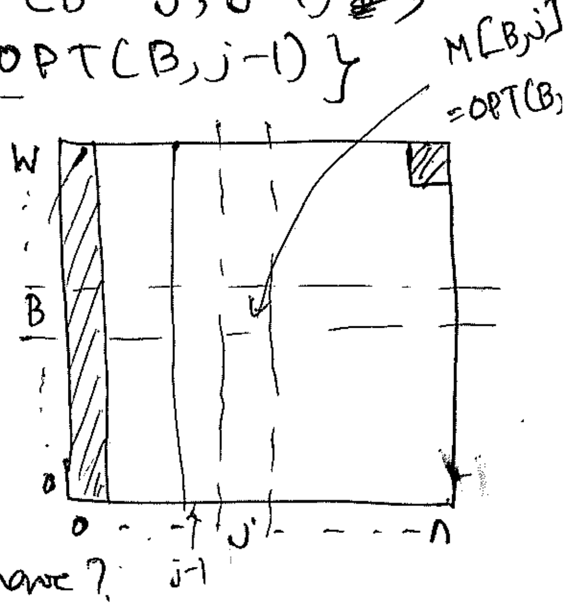
$OPT(B, j) = OPT(B, j-1)$ —— (2)

$w_j > B \Rightarrow OPT(B, j) = OPT(B, j-1)$

<u>Overall recursion:</u>

If $w_j > B \Rightarrow OPT(B, j) = OPT(B, j-1)$

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



Q1) What entry of M is our final output $(w_1, \ldots, w_n; W)$

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

Q2) Initial values:

~~[illegible]~~ $\forall\ 0 \le$ ~~[illegible]~~

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

Q3) How many subproblems do we have?

A3: $(n+1)(W+1) \longrightarrow poly(n)$ If W is $poly(n)$

Q4) Recurrence? **A4)** Done.

Q5) Ordering among subproblems?

A5) Go column by column as knowing $(j-1)^{th}$ column is enough to compute the j^{th} column

---

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

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

   $O(1)$ { if $w_j > B$
   $\quad M[B,j] = M[B,j-1]$
   else
   $\quad M[B,j] = \max\{ w_j + M[B-w_j, j-1], M[B,j-1] \}$
3. Return $M[W,n]$

$O(nW)$ time

Obs: $O(W)$ space if we are only interested in $OPT(W,n)$

But need $\Omega(nW)$ space if we want an actual schedule

$\underline{n=3}$, $w_1 = 1$, $w_2 = 2$, $w_3 = 2$, $W = 3$

	0	1	2	3
3	0	1	3	
2	0	1	2	
B → 1	0	1	1	
0	0	0	0	0

↑
j

$$M[1,1] = \max\{w_1 + M[1-1,0],\ M[1,0]\}$$
$$= \max\{1+0,\ 0\} = 1$$
$$M[2,1] = \max\{w_1 + M[2-1,0],\ M[2,0]\}$$
$$= \max\{1+0,\ 0\} = 1$$
$$M[3,1] = 1$$
$$M[2,2] = \max\{w_2 + M[2-2,0],\ M[2,1]\}$$
$$= \max\{2+0,\ 1\} = 2$$
$$M[3,3] = \max\{2 + M[3-2,2],\ M[3,2]\}$$
$$= \max\{2+1,\ 3\} = 3$$