Nov 28

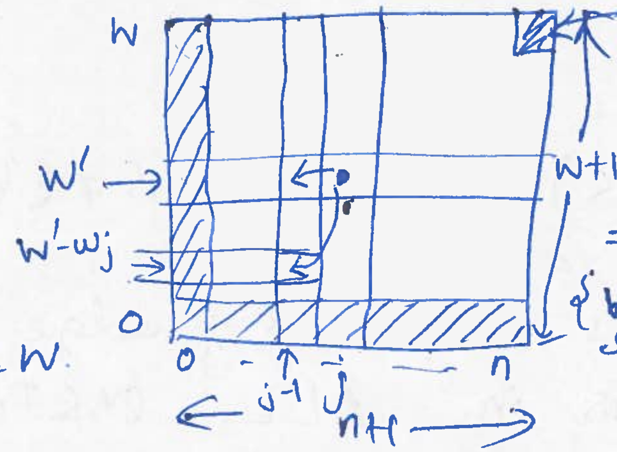
(*)
If $w_j > W'$
$OPT(j, W') = OPT(j-1, W')$
else
$OPT(j, W') = \max\{ OPT(j-1, W'), w_j + OPT(j-1, W' - w_j) \}$

j is not in O_j ← optimal soln for $1..j$

$j \in O_j$

Q1: What entry of M are we interested in?

$OPT(n, W) \sim M[W][n]$



interested in this entry.

$M[W'][j] = OPT(j, W')$

{ rows & columns switched from book

Q2: Shaded values?

~~OPT~~ $M[0][j] = 0 \quad \forall\ 0 \le j \le n$

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

Q3: How many problems?

\# entries in the matrix $= (n+1)(W+1) = \Theta(nW)$

Q4: Recursive formula?

A: See (*)

Q5: Ordering among sub-problems

Obs: Only need to know $(j-1)^{th}$ column to compute the j^{th} column

⇒ Compute the matrix column by column.

Subset-Sum (n, W)

0. Allocate $M[W+1][n+1]$
1. $M[W'][0] = 0 \quad \forall \quad 0 \le W' \le W$.
2. for $j = 1..n$
   for $W' = 0..W$
   Compute $M[W'][j]$ using (*)
3. Return $M[W][n]$.

(*)
If $w_j > W'$
$M[W'][j] = M[W'][j-1]$
else
$M[W'][j] = \max\{M[W'][j-1], w_j + M[W' - w_j][j-1]\}$

Run of algo:

$W = 4$ $\qquad n = 3$

$w_1 = 1, \; w_2 = 2, \; w_3 = 2$

w' \ j	0	1	2	3
4	0	1	3	4
3	0	1	3	3
2	0	1	2	2
1	0	1	1	1
0	0	0	0	0

$w' = 4$, $w_1 = 1$

$M[1][1] = \max\{M[1][0], 1 + M[0][0]\}$ $\quad w'=1, \; w_1 = 1$

$= 1$

$M[2][1] = \max\{M[2][0], 1 + M[1][0]\}$ $\quad w' = 2, \; w_1 = 1$

$= 1$

$M[3][1] = \max\{M[3][0], 1 + M[2][0]\}$ $\quad w' = 3, \; w_1 = 1$

$= 1$

$M[4][1] = \max\{M[4][0], 1 + M[3][0]\}$ $\quad w' = 4, \; w_1 = 1$

$= 1$

$M[3][2] = \max\{\underset{1}{M[3][1]}, \; 2 + \underset{1}{M[1][1]}\}$ $\quad w' = 3, \; w_2 = 2$

$= 3$

→ Compute the optimal ~~schedule~~ subset

If $j \in$ optimal soln for $OPT(W', j)$

$M[W'][j] = \max\{\underbrace{M[W'][j-1]}, \; \underbrace{w_j + M[W' - w_j][j-1]}\}$

$j \in O_j \Leftarrow \quad <$

Ex: Recursively / Iteratively compute the optimal subset

→ Knapsack problem