Apr 17

Simpler Q: max |S| (instead of w(S))

Greedy alg: Sort in increasing order of w_i
& pick as many as you can without
exceeding the budget

Ex. Prove this is optimal (Greedy stays ahead)

Original problem : w(S)

Try: Greedy alg : counter example

$w_A = 1$
$w_B = 3$ $W = 6$
$w_C = 3$

Greedy picks {A, B} but optimal is {B, C}

Note: no known greedy alg.

Dynamic Program for Subset Sum Problem

Goal: compute $w(S) = \sum_{j \in S} w_j$ for an optimal S

O_j : be an optimal solution for $1, \dots, j$ (don't have to be sorted)

$OPT(j) = w(O_j)$

Case 1: $j \notin O_j$ $\quad OPT(j) = OPT(j-1)$

Claim: O_j is also optimal for $w_1, w_2, \dots w_{j-1}$

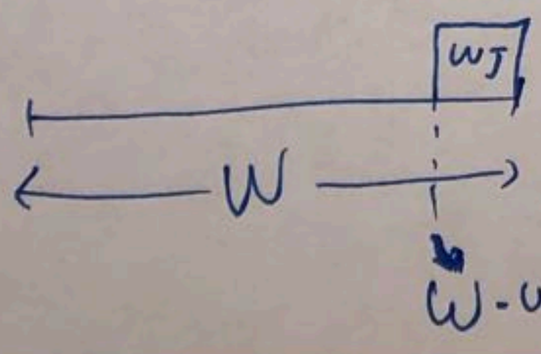
Case 2: $j \in O_j$

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

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

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

(assume $w_j \leq W$)



Solution: Keep track of budget and J

$OPT(B, J)$ = weight of an optimal solution for $w_1, \ldots w_J$ and budget B

Assume $J \in$ optimal$(w_1, \ldots w_J ; B)$ Assume $w_J \leq B$

$\Rightarrow OPT(B, J) = w_J + OPT(B - w_J, J-1)$ — (1)

$J \notin$ optimal$(w_1, \ldots w_J ; B)$

$OPT(B, J) = OPT(B, J-1)$ —— (2)

$w_J > B \Rightarrow OPT(B, J) = OPT(B, J-1)$

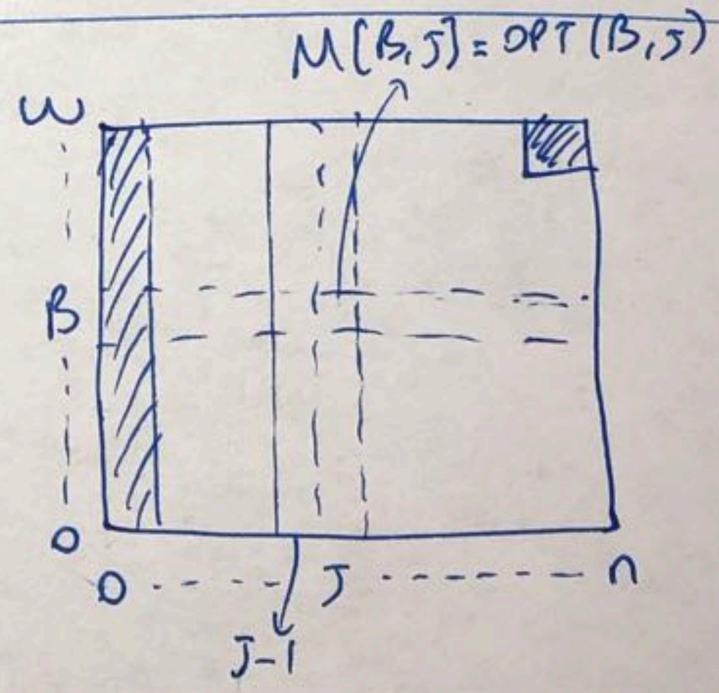
Overall recursion

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 $(w_1, w_2 \ldots w_n ; W)$?

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



Q2: Initial values?

A2: $M[B, 0] = 0 \quad \forall \; 0 \leq B \leq W$

Q3: How many subproblems do we have?

A3: $(n+1)(W+1) \rightarrow$ 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 J^{th} column

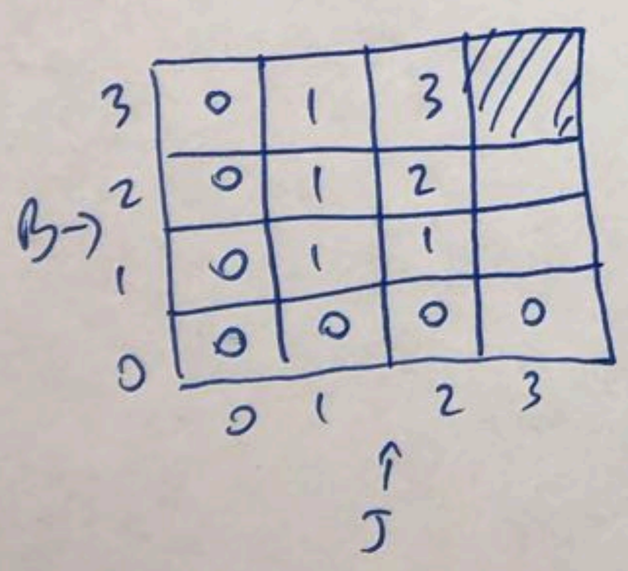
SubsetSum $(w_1, \ldots, w_n ; W)$

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

Obs. $O(W)$ space if we are only interested in $OPT(W, n)$
But need $\Omega(n \cdot W)$ space if we want an actual subset

---

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



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

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

$\vdots$

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