Nov 14

SUBSET SUM problem

Input: n integers $w_1, .., w_n$; $w_i > 0$; $\forall i$

Budget: $W \geq 0$

Output: A subset $S \subseteq [n]$ s.t.

(i) $w(S) \overset{def}{=} \sum_{i \in S} w_i \leq W$ (ii) maximize $w(S)$ {among all valid such S}

Simpler Q: max $|S|$ instead of $w(S)$ in (ii) above?

Q: Can you think of a (greedy) algo for above problem?

→ Sort the w_i's in increasing order & pick as many as possible w/o exceeding W

Ex: Prove the greedy algo does indeed max $|S|$ Hint: Greedy stays ahead

Our problem 2.0 Compute max $w(S)$

↳ Greedy doesn't work (Ex: $W = 6$ $\underbrace{w_1 = 1, \overbrace{w_2 = 3}^{optimal}}_{Greedy}, w_3 = 3$)

Note: No known greedy algos

Dynamic Program

Attempt 1: O_j be optimal solution $w_1, .., w_j$

$$OPT(j) = w(O_j)$$

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

(Ex) Claim: O_j is still opt for $w_1, .., w_{j-1}$

Case 2: $j \in O_j$

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

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

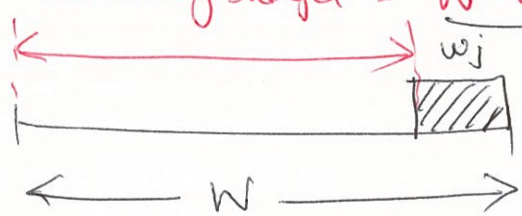
If so, $OPT(j) = w_j + OPT(j')$

Q: What goes wrong in the above? Hint: Something is "missing" in our sub-problem def.

→ What is the actual subproblem if we pick j

Numbers left: $w_1, \dots, w_{j-1}$

Remaining budget = $W - w_j$



→ Define subproblem that includes $w_1, \dots, w_j$; B

Def: $OPT(B, j)$ = wt of an opt. soln on $w_1, \dots, w_j$ with budget B

$\boxed{\text{Assume } w_j \le B}$

Case 1: $j \notin \underbrace{opt(w_1, \dots, w_j; B)}_{w(\ ) = OPT(B,j)}$

$\Rightarrow OPT(B, j) = OPT(B, j-1)$ formal pf: Ex

Case 2: $j \in opt(w_1, \dots, w_j; B)$

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

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

Q: What if $w_j > B \Rightarrow j$ is not in $opt(w_1, \dots, w_j; B)$

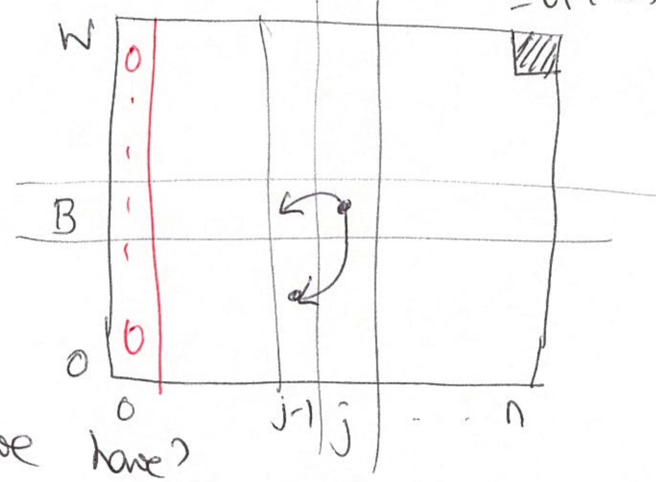
↳ $OPT(B, j) = OPT(B, j-1)$ ← uses the fact that $w_i > 0$

Overall:

$$\boxed{\begin{array}{l} \text{If } w_j > B \text{ then} \\ \quad OPT(B, j) = OPT(B, j-1) \\ \text{else} \\ \quad OPT(B,j) = \max\{ w_j + OPT(B - w_j, j-1), \\ \qquad\qquad\qquad\qquad OPT(B, j-1) \} \end{array}} \quad (*)$$

$OPT(B,0) = 0 \ \forall B$

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



Q1) Given i/p $w_1, \dots, w_n ; W$
have compute $M[B,j]$ correctly for all entries, what is the final answer?

→ $M[W, n] = OPT(W, n)$

Q2) Initialization

Q3) How many subproblems do we have?

$(W+1)(n+1) = O(nW) \leftarrow$ poly If W is poly(n)

Assume this!

Q4) Recurrence? (*)

Q5) Ordering among the sub-problem?

→ Knowing the $(j-1)^{th}$ column is enough to compute j^{th} column

→ $M[:,0], M[:,1] \dots M[:,n]$

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

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

overall $= O(nW)$

Ex: Prove by induction on j that $M[B,j] = OPT(B,j)$