Review (Contd)

Subset problem:

Items: $1, \ldots, n$
Weights: $w_1, \ldots, w_n$

Find a subset of maximal weight, under W.

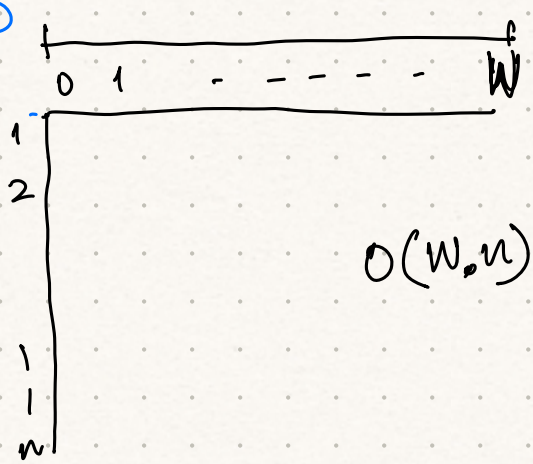
We want a subset $\{i_1, \ldots, i_k\}$ s.t $\sum_{j=1}^{k} w_{i_j} \leq W$ and $\sum_j w_{i_j}$ is maximised & maximal

Either item 1 is a part of Opt
↳ $Opt([2,n], W-w_1)$
or not.
↳ $Opt([2,n], W)$

} 2-param DP

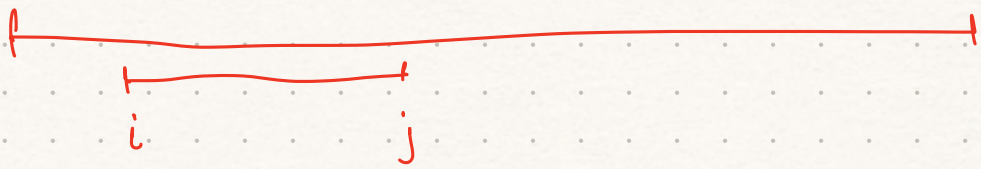
We assumed that the wts are int.

$Opt([i,n], \tilde{W})$



$O(W,n)$

Qn: Is there a subset that sums up to A.
↓
Contiguous



Array Partitioning Problem:
(Sundar Vishwanathan's notes, IITB)

Contiguous partition
Input is a Penalty matrix nxn

- Each partition involves a cost c.

Want: Divide the array into parts s.t each part has contiguous elements and total partition cost is minimized.

→ Sum of Penalties + cost of parts

→ If there is only one part → $P(1,n)$

→ $P(1,i) + c + P(i+1,n)$

→ $i_1, \ldots, i_k$ → $c \cdot N + P(1,i_1) + P(i_1+1, i_2) + \ldots + P(i_k+1, n)$.

$$\text{Cost}([1,n]) = \min\left\{ \min_k \{ \underbrace{\text{Cost}([1,k])}_{P[1,n]}, + c + \text{Cost}([k+1,n]) \} \right.$$

Say there is an i^* in opt part s.t $[1,i^*]$ is a part ~ $[i^*+1, k^*]$, $[k^*+1, n]$

← Space $O(n^2)$ time $O(n^2 \cdot n)$

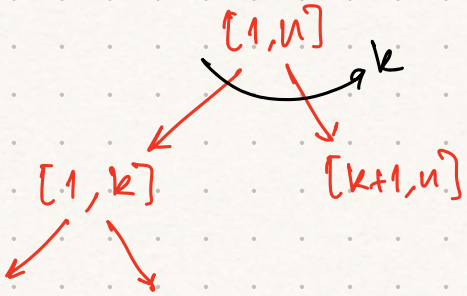
$$\text{MinCost}([1,n]) = \left\{ \min_{1 \le i < n} \{ P[1,i] + c + \text{MinCost}[i+1,n] \}, \right.$$

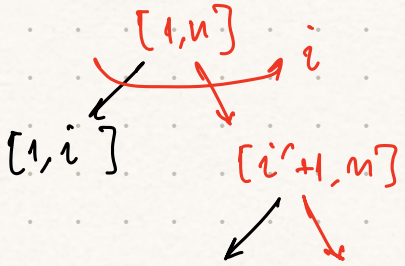
$$\left. P[1,n] \right\}.$$

Space $O(n)$ time $O(n^2)$

$\text{MinCost}([i,n])$

$$= \min\left(\{ P[i,k] + c + \text{Mincost}[k+1,n] \mid i \le k < n \} \cup \{P[i,n]\} \right).$$





Interval Scheduling:

← Max no. of jobs that ~~can be scheduled w/o overlaps.~~ (Before)

Jobs/Reqs	J_1, . . . ,	J_n	
Start	s_1	s_n	← Want: The duration of scheduling to be maximized as well. (Now)
Finish	f_1	f_n	

→ Sort your reqs in incr order of start times.

J_1, . . . , J_n: index of first non-overlapping job

$|J_1| + Opt(First[1], n)$ $\qquad$ $Opt(2, n)$

$$Opt(1,n) = \max \begin{cases} Opt(2,n) \\ |f_1 - s_1| + Opt(First[1], n) \end{cases}$$

$$Opt(i,n) = \max \begin{cases} Opt(i+1,n) \\ |f_i - s_i| + Opt(First[i], n) \end{cases}$$