Advanced DP Topics

LIS: Longest Increasing Subsequence

$x_1, \ldots, x_n$

Naive DP:

$LIS(i)$ = LIS ending at i

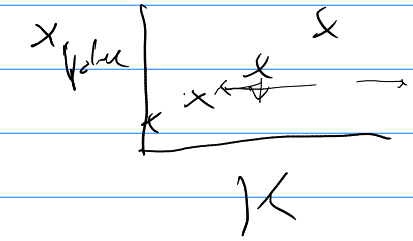
$$LIS(i) = 1 + \max_{j<i \mid x_j \le x_i} LIS(j)$$

$$\text{Answer} = \max_i LIS(i)$$

Time: $O(n^2)$

Fancier:

$Lowest(i, k)$ = smallest last value of length-k IS ending before i.



$$Lowest(i, k) = \min \begin{cases} Lowest(i-1, k) \\ x_i \quad \text{if } \exists \text{ length } k \text{ sequence ending at } i \end{cases}$$

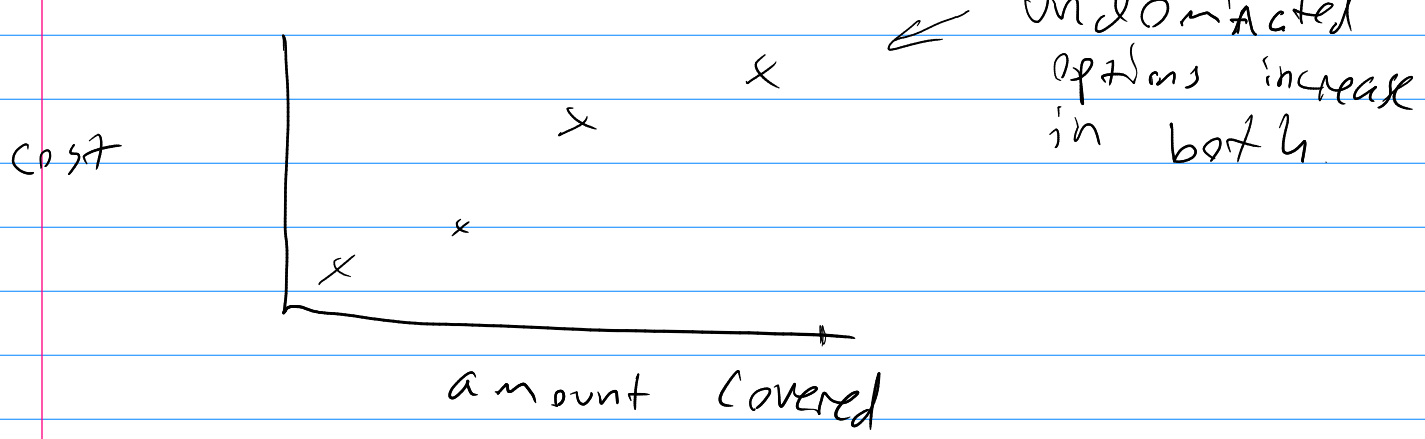
$\Leftrightarrow$ Lpast $(i-1, k-1) < x_i$

Homework:

min cost interval cover.

Sort by finish time.

Build from left



Cost[i]

Cover[i]

For new (s, t, c):

Let j = min j s.t. Cover[j] $\geq$ s.

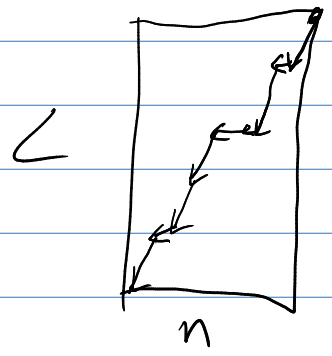
New option (Cost[j] + c, t)

Add to list (undominated since t bigger)

But first, pop till Cost[-1] $\leq$ Cost[j] + c

Sliding Window + finding set

Recall Knapsack:



$val(i, v) = \max(val(i-1, v),\ val(i-1, v-s_i) + v_i)$

$O(nC)$ time

$O(C)$ space if only track 2 columns

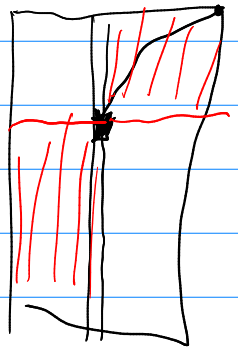
How to find set?

Normally, Back pointers.

But: that's $O(nC)$ space.

just need the path. $O(n)$.

Idea: only ~~keep~~ back pointers into column $n/2$



[create at $i = \frac{n}{2}$; just copy over]

$O(nC)$ time to get 1 point on path.

$C_{n/2}$ = cap. used after first $n/2$.

Recurse:

Knapsack($\{$items $1, \ldots, n/2\}$, $C_{n/2}$)

Knapsack($\{$items $\frac{n}{2}+1, \ldots, n\}$, $C - C_{n/2}$)

will get whole path $(0, 0) \ldots (n, C)$

How long?

$T(n, C) = nC + T(\frac{n}{2}, C_{n/2}) + T(\frac{n}{2}, C - C_{n/2})$

$\geq n$

Time in next round $= \frac{nC}{2}$.

$$nC + \frac{nC}{2} + \frac{nC}{4} + \ldots = O(nC).$$