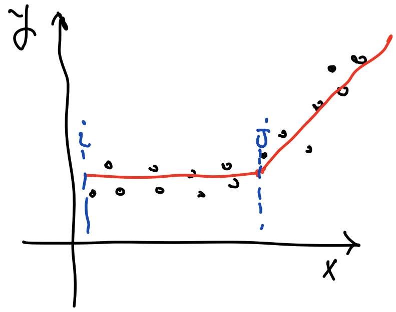
Segmented L-S lines



Min $E + C L$

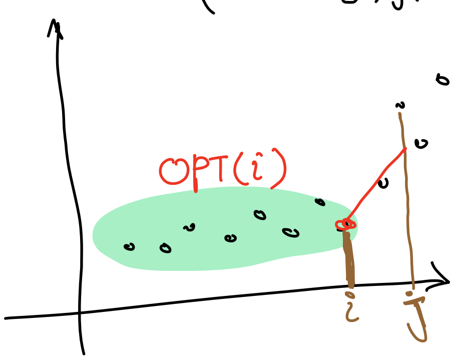
Optimum

$$E_{ij} = \frac{\sum_{k=i}^{j} (y_k - a x_k - b)^2}{j - i}$$

$$a = \frac{(j-i) \sum x_k y_k - \sum x_k - \sum y_k}{(j-i) \sum x_k^2 - \left(\sum x_k\right)^2}$$

$$b = \frac{\sum y_k - a \sum x_k}{(j-i)}$$

$$OPT(j) = \begin{cases} 0 & j = 0 \\ \min\{OPT(i) + C + E_{ij}\} \quad \forall i \in [0, j) \end{cases}$$



$M[j] = OPT(j)$

$M[0] = 0$

// Compute the errors

for $j = 1$ to n

for $i = 0$ to $j-1$

e_{ij} = Compute error (i, j)

$O(n)$

$O(n^3)$

// Solve the problem

for $j = 1$ to n

$M[j] = \boxed{\min\left(OPT(i) + C + e_{ij}\right) \ \forall i \in [0, j)}$

$O(n)$

$O(n^2)$

return $M[n]$

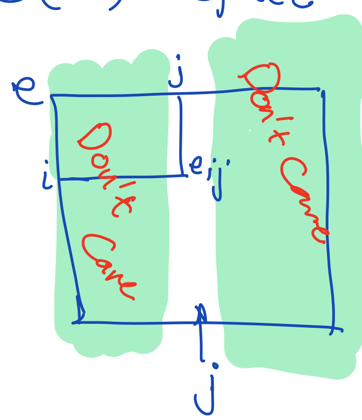
$O(n^3)$ time

$O(n^2)$ Space

Can you make it

$O(n^2)$ time

$O(n)$ Space



$$Y_t = \sum_{k=\ell}^{t} y_k$$

$$X_t = \sum_{k=\ell}^{t} x_k$$

$$XY_t = \sum x_k y_k$$

$$X2_t = \sum_{k=\ell}^{t} x_k^2$$

$$X_t = X_{t-1} + x_t$$

$$Y_t = Y_{t-1} + y_t$$

...

$M[0] = \emptyset$

for $j = 1$ to n

  // Compute e_{ij}

  $X = 0;\ Y = 0; \ldots$

  for $t = 1$ to $j-1$

    $X = X + x_t$; $Y = Y + y_t$

    ...

    $e_{j-t,j} = \text{Error}_{\text{C-time}}(X, Y, XY, X^2)$ $O(1)$

  // Compute $M[j]$

  $M[j] = \min(M[i] + C + e_{ij})$

  $\forall i \in [0, j)$

return $M[n]$

$O(n)$ $O(n)$ $O(n^2)$

Space: $O(n)$

0/1 Knapsack

Capacity : C

item: 1 2 .... n

value: v_1 v_2 .... v_n

weight: w_1 w_2 .... w_n

Goal: Select a Subset of items that

(1) Fit in the backpack

$$\sum w_i \leq C$$

(2) $\text{Max} \sum v_i$

Greedy Doesn't work:

$C = 5$

	1	2	3
v_i	4.1	3	2
w_i	4	3	2

Greedy: $\{1\}$, $\sum v_i = 4.1$

Optimal: $\{2, 3\}$, $\sum v_i = 5$

$OPT(i, c')$

items available

Capacity left

$OPT(n, C)$

output

$$OPT(i, c') = \begin{cases} 0 & i = 0 \text{ OR } c' = 0 \\ \text{Max}\left(\overbrace{OPT(i-1, c'-w_i) + v_i}^{\text{Select } i},\ \overbrace{OPT(i-1, c')}^{\text{not Select } i}\right) \end{cases}$$

$M[i, j] = OPT(i, j)$

for $i = 1$ to n $\quad M[i, 0] = 0$

for $j = 1$ to C $\quad M[0, j] = 0$

for $i = 2$ to n

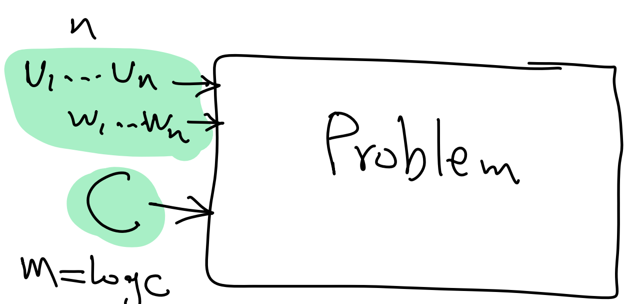
  for $j = 2$ to C

    $M[i, j] = \text{Max}(M[i-1, \underbrace{j - w_i}_{\text{if } j - w_i \geq 0}] + v_i,\ M[i-1, j])$

return $M[n, C]$

Space: $O(nC)$

Time: $O(nC)$



$O(n 2^m)$ ← Pseudo Polynomial

0/1 Knapsack is NP-Complete