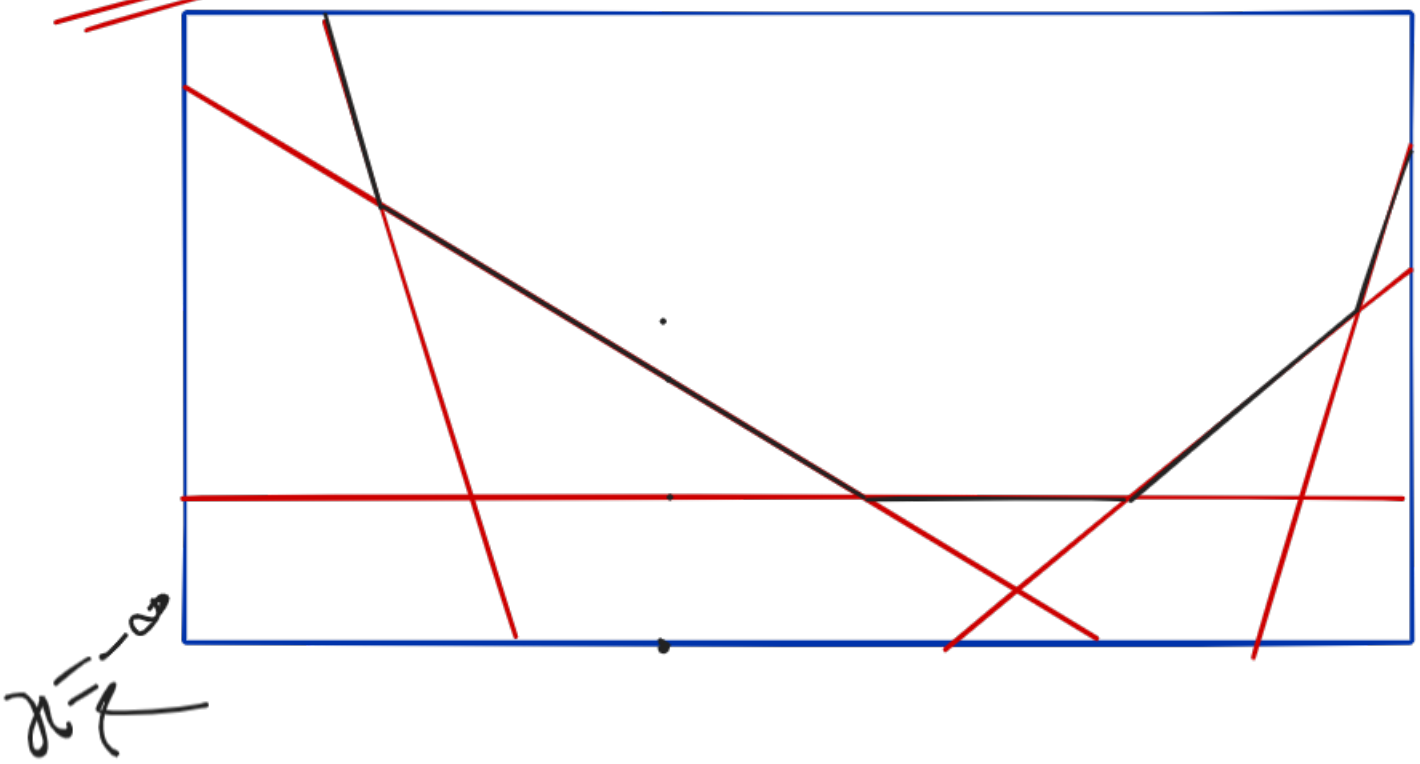
DP optimization

1. CHT - convex hull optimization
2. Li chao tree
3. Divide and conquer optimization
4. Quadrangle inequality
5. 1D/1D optimization

degree → O(n)



n<= 10e5

$y = mx + C$

Obl → Sort m.

O(n) ← Build O(log n) ← Query

DP[i] = MAX { DP[J] + C(j , i) } for (j<i)

$m_j x + (DP(j) + C) \approx mx + c'$.

C(j,i) -> O(1)

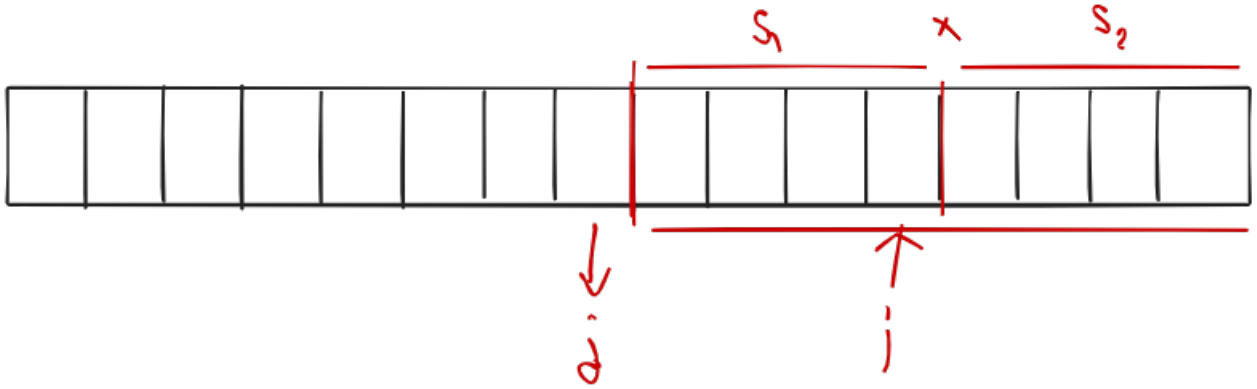
m → monotonic

Classical dp -> O(n^2)

$C(j,i) = mi + C$

$m = g_1(j)$

$c = g_2(j)$



$dp[i][k] = \max_{j<i} \{ DP[j][k-1] + C(j,i) \}$

↓ i

$C = S_1 * S_2 = (P_n - P_i)(P_i - P_j)$

$= P_n P_i - P_i^2 - P_n P_j + P_i P_j + DP_j$

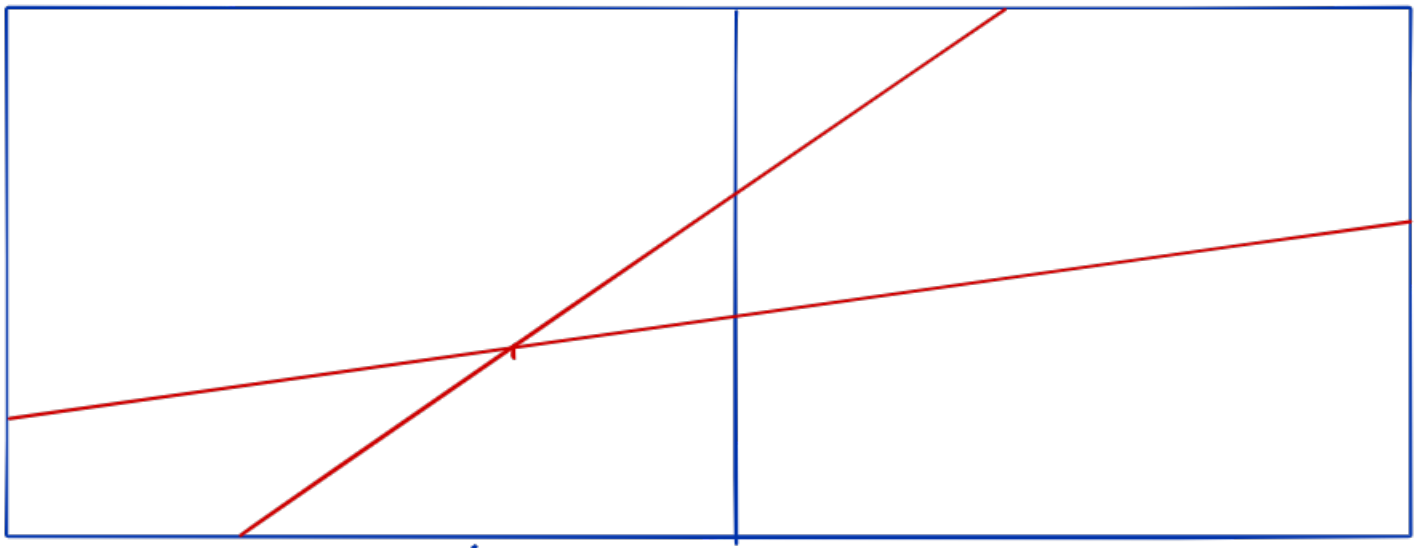
$= -P_i^2 + (P_n + P_j) P_i - P_n P_j + DP_j$

m x + C

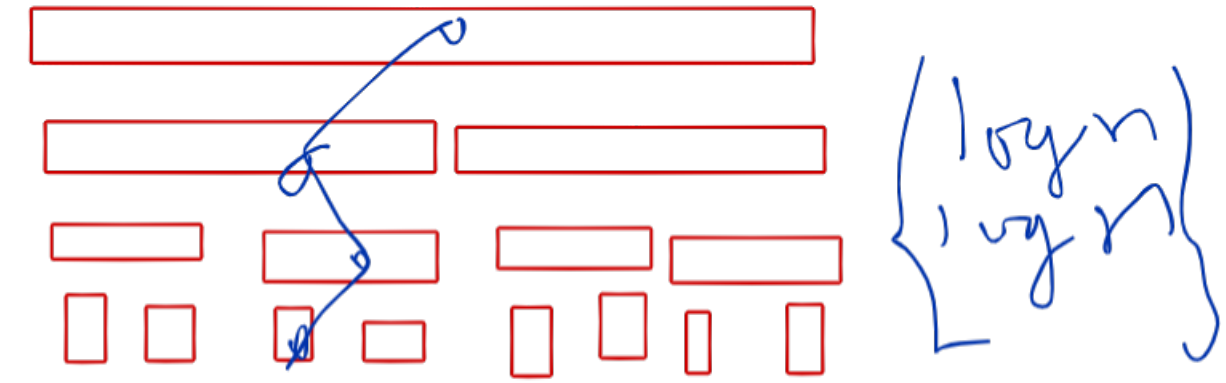
$k \rightarrow O(n \log n)$

$\rightarrow O(nk \log n)$

Li Chao tree :



$(m \geq 0, c = -\infty)$



D&C optimization :

Dp[i] = MIN { DP[J] + C(j,i) } for j<i

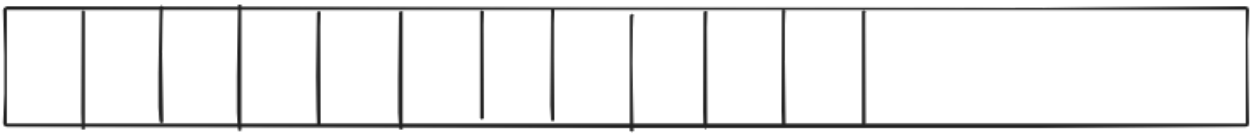
opt(i) = lowest j which give optimal DP(i)

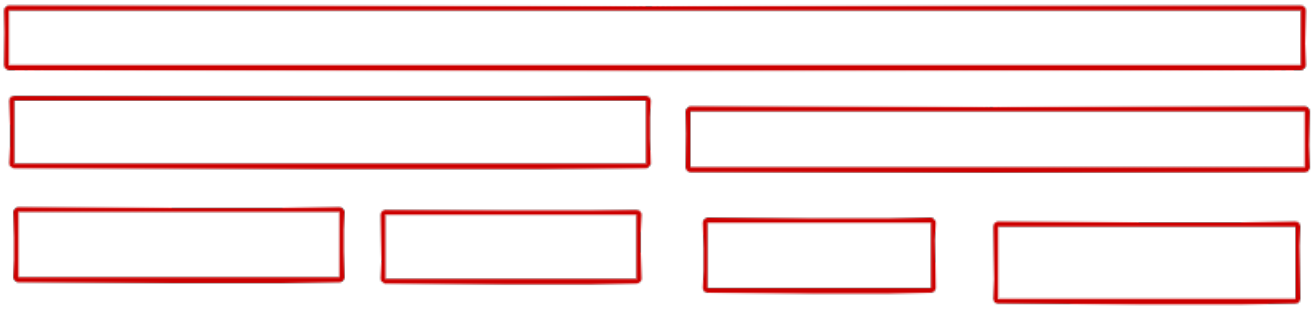
DP[i] = DP[opt(i)] + C(opt(i) , i) ✓

opt(i) → Non - decreasing

-> Segmennt [L,R]

1. mid = (l+r)/2
2. opt(mid) in 0



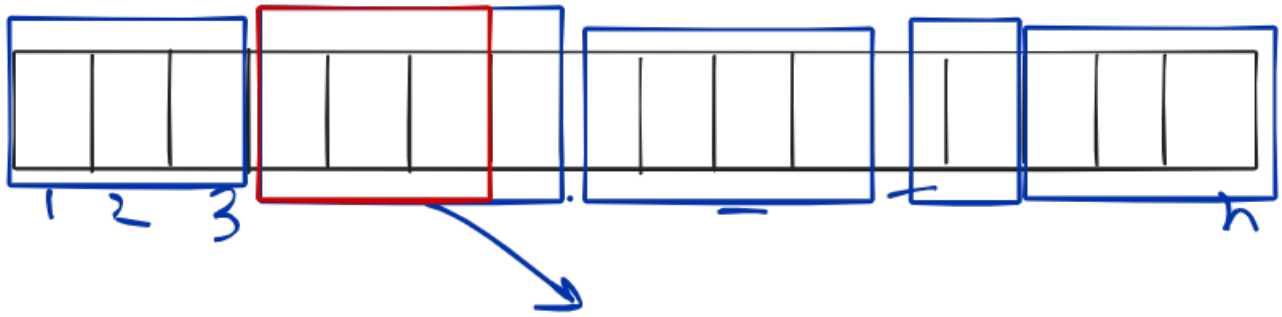


Dp[i] = MIN { DP[J] + C(j,i) } for j<i

Quadrangle inequality : C(i , j+1) - C(i , j) ≥ C(i+1 , j+1) - C(i+1 , j)

opt(i) ≤ opt(i+1)

Non decreasing



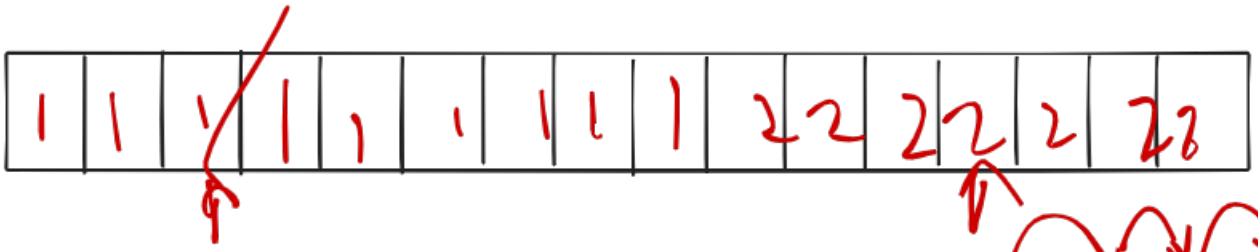
$$\begin{cases} U(i,j) \geq 0 \\ U(i,i) = 0 \end{cases} \quad \sum U(1,j)$$

Min [Total]

DP[I][K] = MIN { DP[J][K-1] + C(j,i) }

$$C(i,j+1) - C(i,j) \geq C(i+1,j+1) - C(i+1,j)$$

$$[i,j] \rightarrow j+1 \quad \geq \quad [i+1,j] \rightarrow j^*$$



-> quadrangle inequality

$$DP[i] = Min\{DP[j] + C(j,i)\}$$

$$DP(i) = Min\{A[j] + C(j,i)\}$$