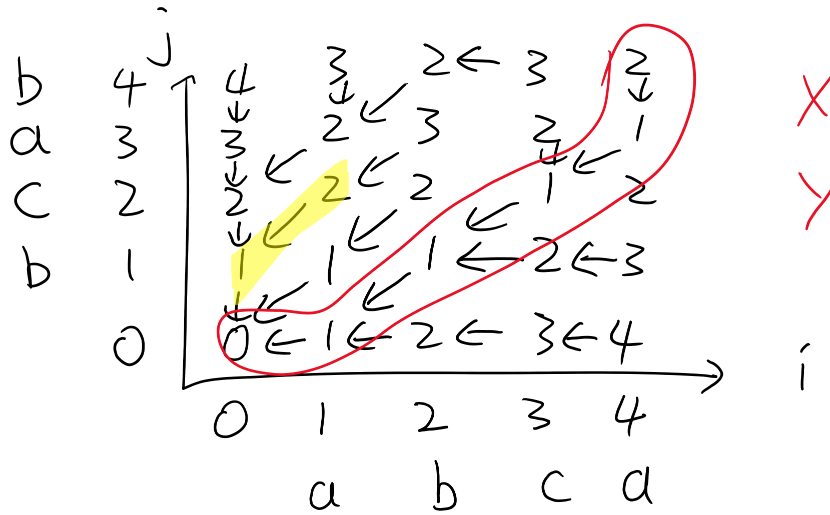
Edit distance

Let $d(i,j)$ = the edit distance between string $x_1, x_2, \ldots, x_i$ and string $y_1, y_2, \ldots, y_j$

Then,

$$d(i,j) = \begin{cases} j & \text{if } i = 0 \\ i & \text{if } j = 0 \\ \min\left(d(i-1,j-1) + 1_{x_i \neq x_j},\ d(i-1,j)+1,\ d(i,j-1)+1 \right) \end{cases}$$



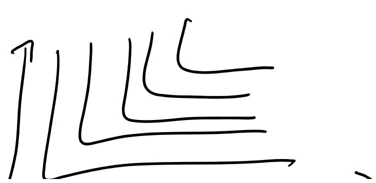
X_i : a b c a ␣

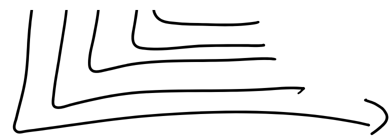
Y_j : ␣ b c a b

Runtime: $O(mn)$ time, $O(mn)$ space $m = len(x)$, $n = len(y)$

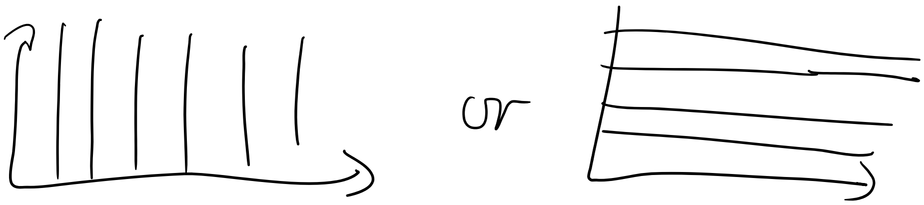
If we just want edit distance,

- Can do $O(m+n)$ space

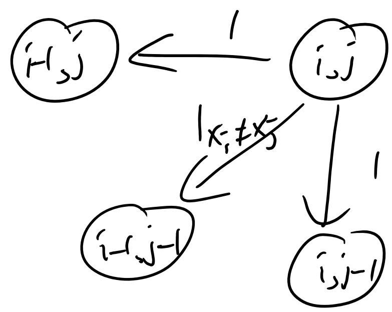




- can do $O(\min(m,n))$

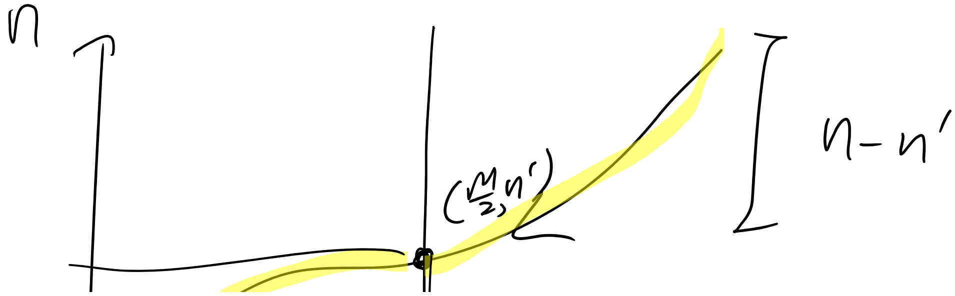


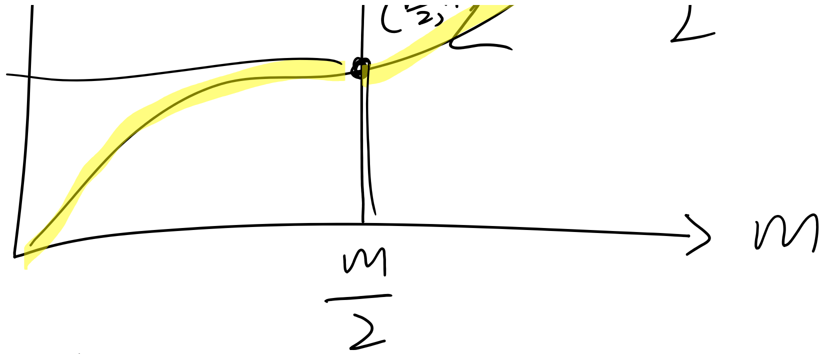
DP for edit distance is computing shortest path for the directed graph from (m,n) to $(0,0)$ with $(m+1)(n+1)$ vertices



Question: How to find alignment (shortest path) in $O(mn\log(n))$ time $O(m+n)$ space.

Ans: DP + divide and conquer.

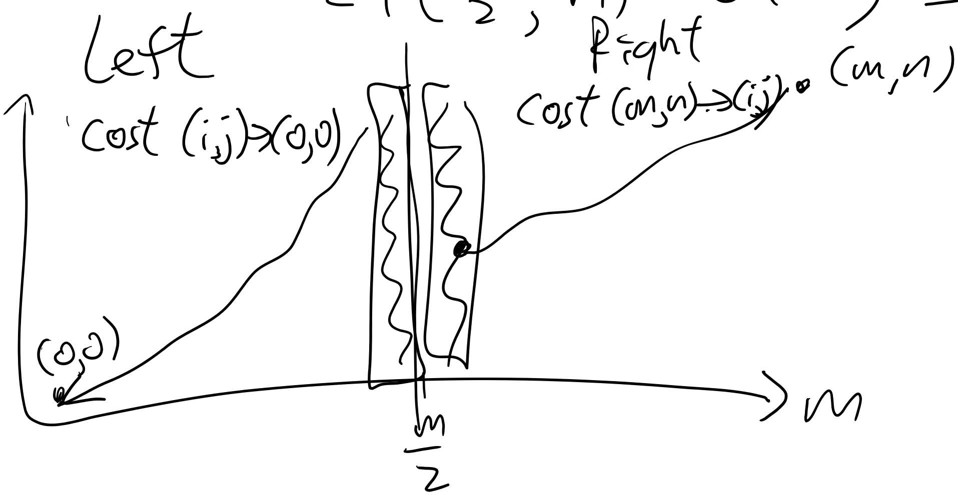


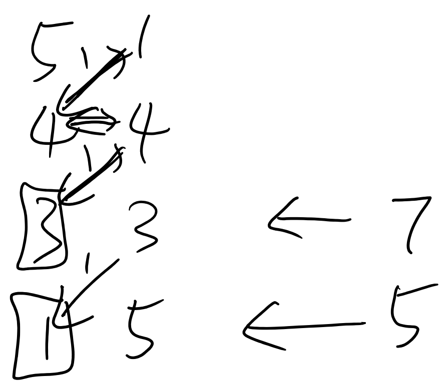


Let $T(m,n)$ = cost finding shortest path $(m,n) \to (0,0)$

$$T(m,n) = T\left(\frac{m}{2}, n-n'\right) + T\left(\frac{m}{2}, n'\right) + \underbrace{\boxed{\text{cost computing } n'}}_{= O(mn)}$$

$$\le 2T\left(\frac{m}{2}, n\right) + O(mn) = O(mn \log m)$$





Longest path in DAG

Longest Path in DAG

- Order vertices $(i,j) \in E \Rightarrow i < j$

$\boxed{\text{Wrong}}$ $1 \quad 2 \quad 3 \quad \cdots \quad n$

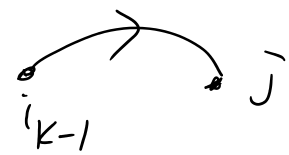
$OPT(j)$ = length of long path on $\{1, 2, \ldots, j\}$

Case 1: longest path doesn't pass j

$$OPT(j) = OPT(j-1)$$

Case 2: longest path end at j

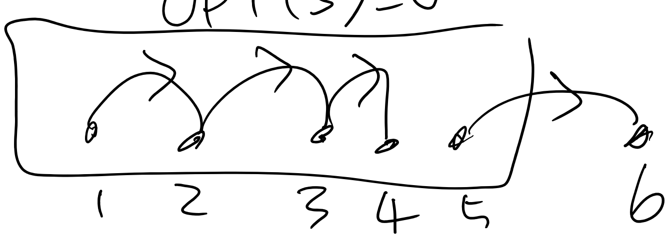
Let $i_1, i_2, \ldots, i_{k-1}, j$



$$OPT(j) = OPT(i_{k-1}) + 1$$

$$OPT(j) = \begin{cases} 0 & \text{if } j \text{ is source} \\ \max\left(OPT(j-1),\ 1 + \max\limits_{(i,j)\in E} OPT(i)\right) & \text{else} \end{cases}$$

$OPT(5) = 0$



$OPT(j)$ = longest path ends at j

$$OPT(j) = \begin{cases} 0 & j \text{ is source} \\ 1 + \max\limits_{(i,j)\in E} OPT(i) & \text{else} \end{cases}$$

$$\text{Ans} = \max_{1\le i\le n} OPT(i)$$