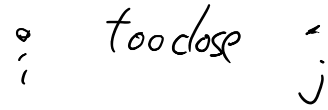
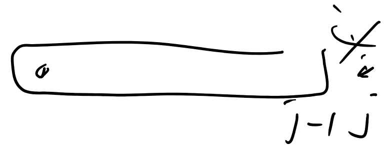
Let $OPT(i,j)$ = max # base pairs for substring $b_i, \ldots, b_j$

Case 1: $j - i \le 4$



$OPT(i,j) = 0$.

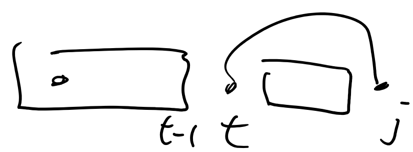
Case 2: j is not matched



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

Case 3: j is matched with t $\quad i \le t < j-4$.

$$OPT(i,j) = 1 + OPT(i, t-1) + OPT(t+1, j-1)$$



ALG:

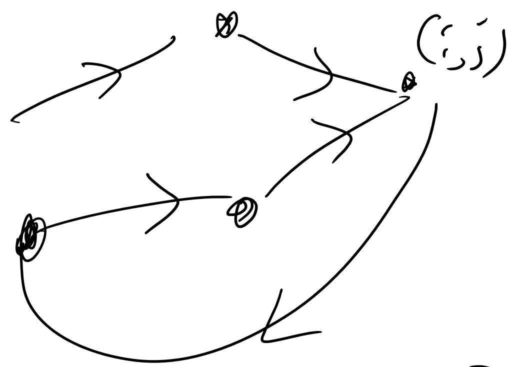
$$OPT(\boxed{i,j}) = \begin{cases} 0 & j - i \le 4 \\ \max\Big(\overset{\text{Case 2}}{OPT(i, j-1)}, \\ \quad \max\limits_{i \le t < j-4}\big(1 + OPT(i, t-1) + OPT(t+1, j-1)\big)\Big) \end{cases}$$

$i = 1, \ldots, n$

Runtime: # subproblems $= O(n^2)$

each subproblem takes $O(n)$ time.

So, runtime $= O(n^3)$



Why there's no loop in recursion?

because 2nd pointer is strictly decreasing.

or because $j-i$ is strictly decreasing

Let $OPT(i,j)$ be the edit distance between $x_1 \ldots x_i$ and $y_1, \ldots, y_j$

$\underbrace{x_1 \ldots \quad x_i}$ $\boxed{x_n}$ X

$\underbrace{y_1 \ldots \quad y_j}$ Y

Case 1: x_i match with y_j

$$OPT(i,j) = OPT(i-1, j-1) + \begin{cases} 1 & \text{if } x_i \neq y_j \\ 0 & \text{else} \end{cases}$$

Case 2: x_i is matched with ␣

$$OPT(i,j) = 1 + OPT(i-1, j)$$

Case 3: y_j is matched with ␣

$$OPT(i,j) = 1 + OPT(i, j-1)$$

	x_1	x_2	x_3	x_4
X	A	B	C	D
	|		|	*
Y	A	␣	C	E
	y_1		y_2	y_3

ALG:

$\Big\lceil$ j if $i=0$

i if $j=0$ or if $x_i \neq y_j$

ALG:

$$OPT(i,j) = \begin{cases} i & \text{if } j=0 \\ \min\Big(OPT(i-1,j-1) + \begin{cases} 1 & \text{if } x_i \neq y_j \\ 0 & \text{else} \end{cases}, \\ \quad 1+OPT(i-1,j),\ 1+OPT(i,j-1)\Big). \end{cases}$$

a b ␣ c   abc

13   12

a b c ␣   abc

runtime: n^2 subproblems $\times$ $O(1)$ / subproblem.

$= O(n^2)$

No loop: $i+j$ is strictly decreasing.