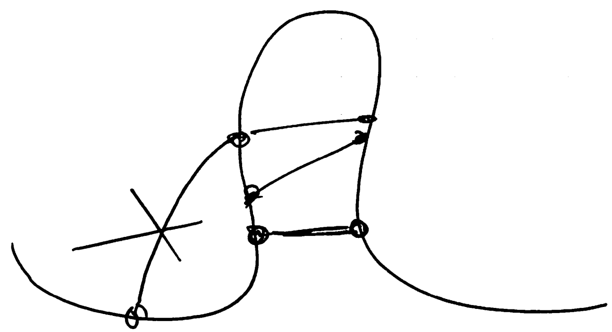
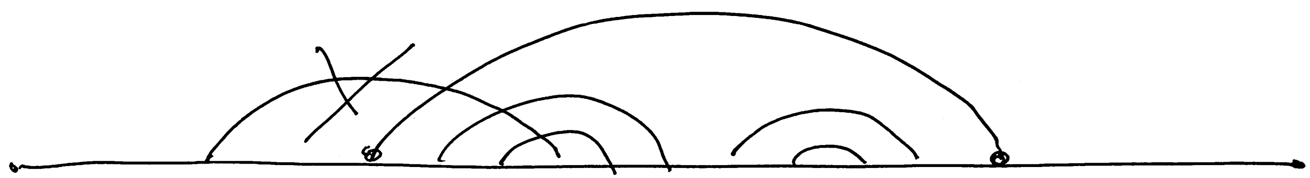
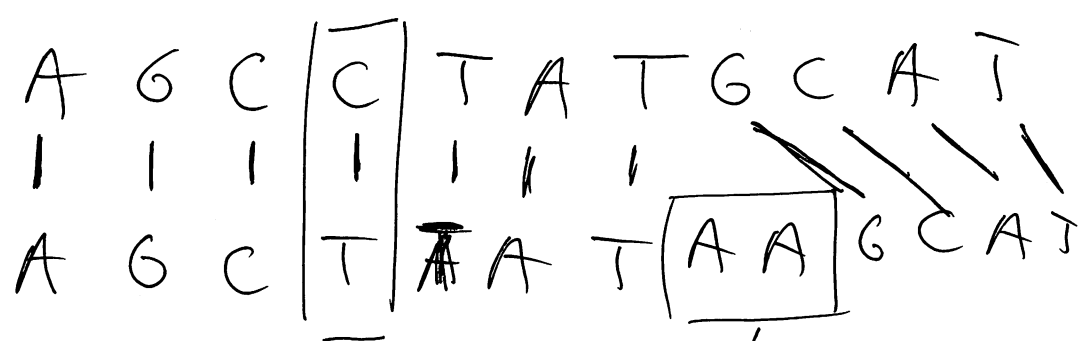






$\alpha(C,T)$ $\quad$ 2δ

First (incorrect) try

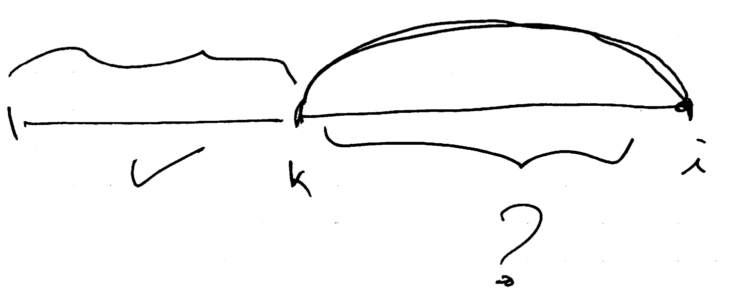
① $M[i]$ = ~~cost~~ size of the best matching for position $1, 2, \ldots, i$ in the string

②

If i is <u>not</u> matched in opt.

$M[i] = M[i-1]$

If i is matched to $k < i-4$



X stuck

Second (correct) try

① $M[i,j] =$ size of the best matching for positions $i, i+1, \ldots, j$

② If j is not matched

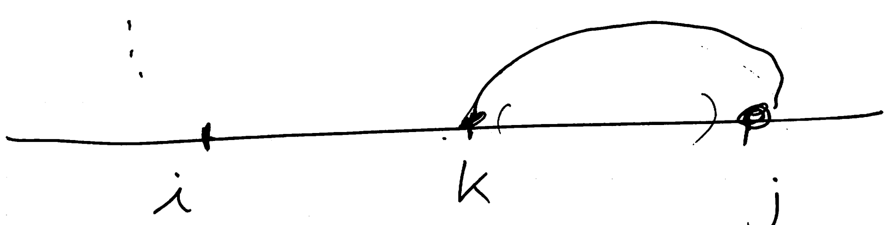
$$M[i,j] = M[i, j-1]$$

Else, j matched to k : $i \le k \le j-4$

$$M[i,j] = M[i, k-1] + M[k+1, j-1] + 1$$

$A[k] = U$ and $A[j] = A$, or

$\vdots$



③ # DP states $= \binom{n}{2} = O(n^2)$

each takes $O(n)$ time

total time is $O(n^3)$

$\Rightarrow$

$$\begin{cases} M[i,j] = \max\left(M[i,j-1], \ \max\limits_{i \le k < j-4} M[i,k-1] + M[k+1,j-1] + 1 \right) \\ M[i,j] = 0 \quad \text{if } j - i \le 4 \end{cases}$$

① $M[i,j]$ = best cost for alignment of $x_1, x_2, \ldots, x_i$ and $y_1, y_2, \ldots, y_j$

③ # DP states = nm

Each takes = $O(1)$ time

---

total time = $O(nm)$ time

②

If i unmatched in OPT

$$M[i,j] = M[i-1,j] + \delta$$

If j unmatched in OPT

$$M[i,j] = M[i,j-1] + \delta$$

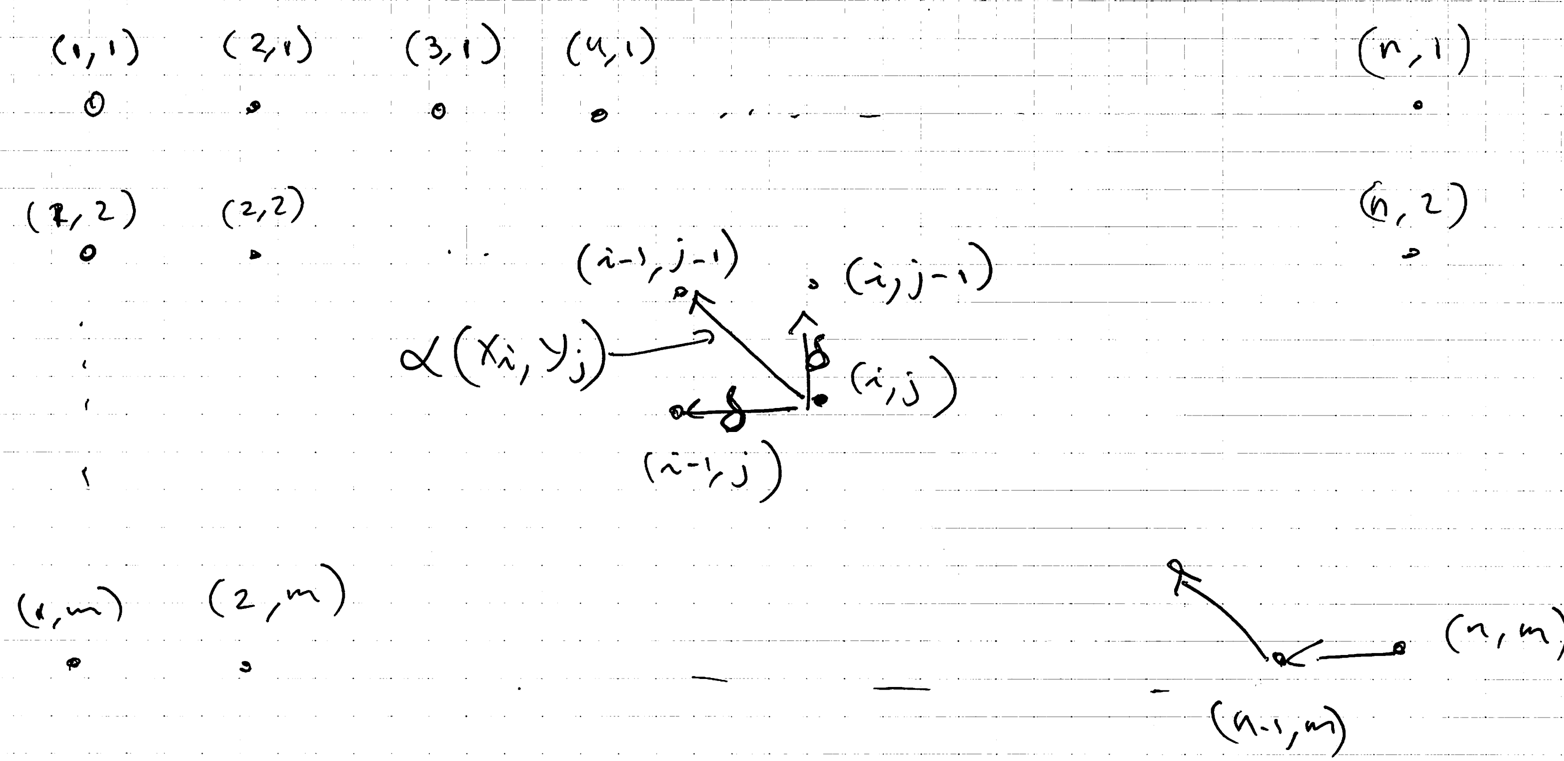
Else x_i — y_j

$$M[i,j] = M[i-1,j-1] + \alpha(x_i, y_j)$$

Assume $\alpha(z,z) = 0$

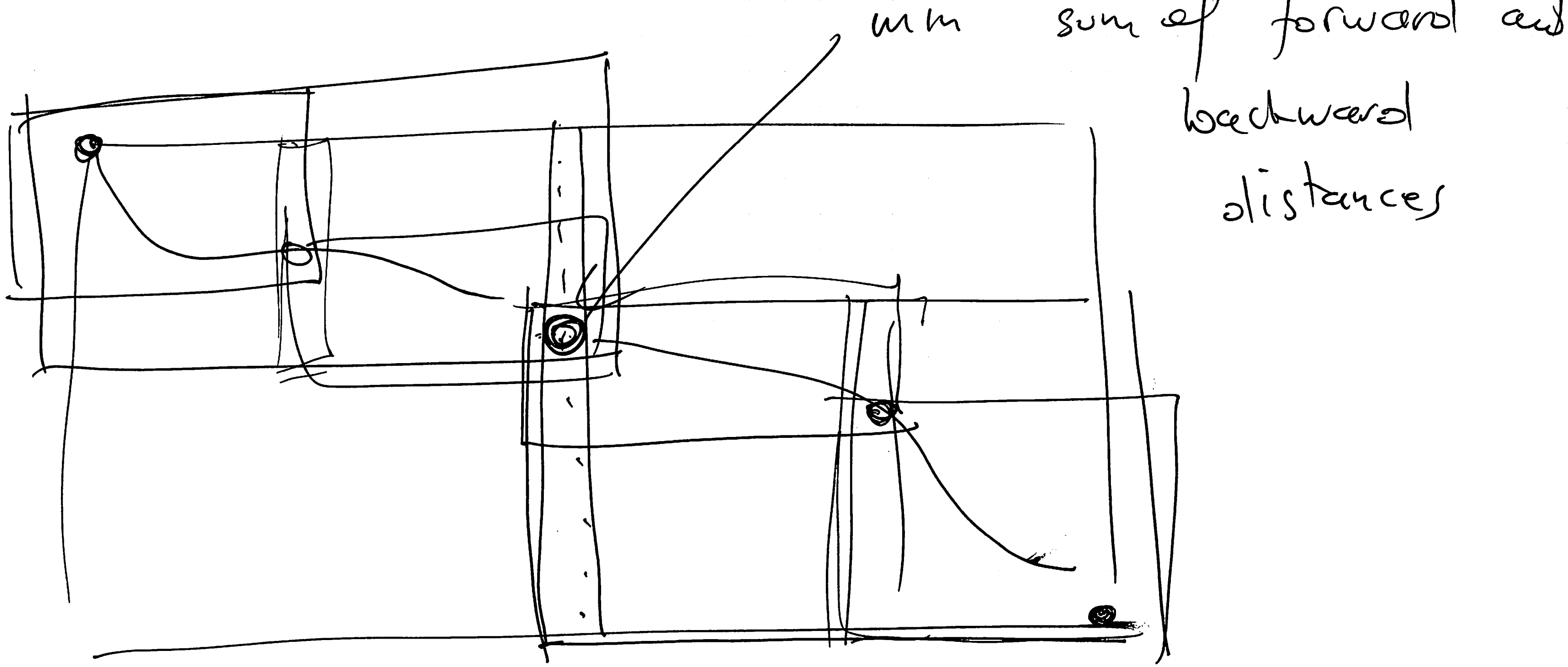
$$\Rightarrow \begin{cases} M[0,j] = \delta j \quad M[i,0] = \delta i \\ M[i,j] = \min\big(M[i-1,j] + \delta, \\ \quad M[i,j-1] + \delta, \\ \quad M[i-1,j-1] + \alpha(x_i, y_j)\big) \end{cases}$$

TITLE: DATE / NO.



$\Rightarrow$ compute $M[n,m]$ in $O(nm)$ time

$O(m)$ space



$\Longrightarrow$ linear space

$\Longrightarrow$ $O(nm)$ time

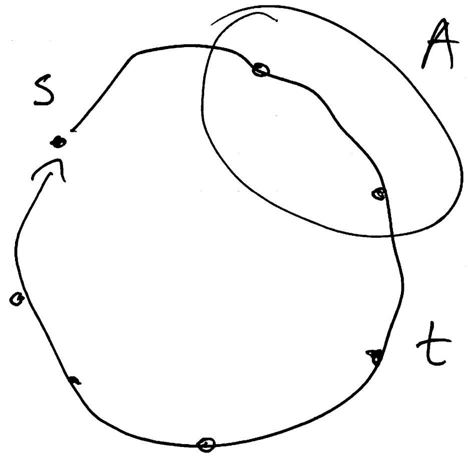
What's the trivial algorithm for TSP?

pre compute distances ~~$\Theta($~~ dist(x_i, x_j) $\forall x_i, x_j \in X$

go over all permutations of X

keep the one minimizing objective

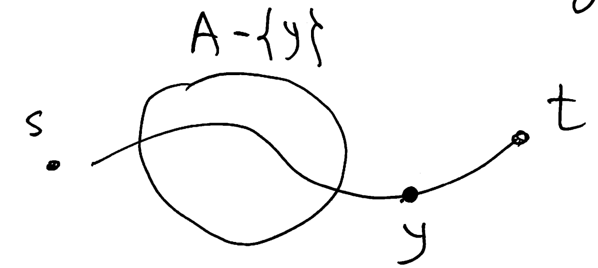
$\Rightarrow$ Time complexity: $O(K!\,K)$ time



$A \subseteq X$

1 $M[A, t] =$ cost of going from s to t through cities in A

2



$$M[A,t] = M[A-\{y\}, y] + dist(y,t)$$

$$\Rightarrow \begin{cases} M[A,t] = \min\limits_{y \in A} \left(M[A-\{y\}, y] + dist(y,t) \right) \\ M[\emptyset, t] = dist(s,t) \end{cases}$$

3 Time complexity

#DP states $= 2^k$ k

each takes $= O(k)$ time

$\Rightarrow O(2^k k^2)$ total time