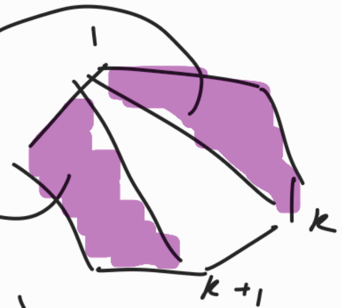
Cost of triangulation

$\binom{n}{2}$ diagonals.

<u>Idea</u> 1: $Opt([1,n]) = \min\limits_{\substack{\text{diagonals}\\\text{available}}} \left(Opt(A_1) + Opt(A_2) \right)$ → Not all diagonals are compatible with each other.

<u>Idea</u> 2 : Triangle $(1, k, k+1)$

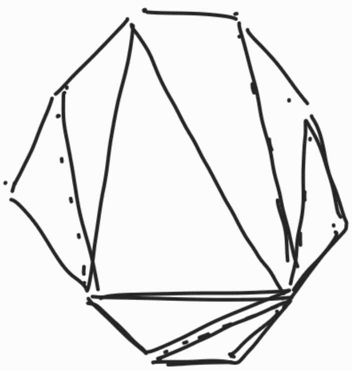


Total cost = Cost + Cost + Perimeter of $(1, k, k+1)$

↳ Iterate over all s^{ks}. Get the min.

$$Cost([1,n]) = \min_{k \in S} \left\{ cost([1,k]) + cost([1 \to k+1, n]) + (w_{ik} + w_{k,k+1} + w_{k+1,1}) \right\}$$

$[1,n] \setminus [2,k]$



<u>Regular</u>

$$cost(A[1,n]) = \min_{k \in [2,n]} \left\{ cost(A[1,k]) + cos(A([1,n] \setminus [2,k])) + (w_{1,k} + w_{k,k+1} + w_{k+1,1}) \right\}$$

<u>Non regular</u>

$$cost(A[1,n]) = \min_{i,k} \left\{ cost(A[i,k] + cost([1,n] \setminus [i+1, k]) + w_{i,k} + w_{k,k+1} + w_{k+1,1} \right\}$$

JeffE - DP

DP

- Optimal substructure (Recursive)
- Memory

Recursion $\longrightarrow$ Recursive statement + base case.

$A[i,j]$ = cost of triangulation with vertices

$i, i+1, \ldots, j-1, j$.

$A[i,k]$ $\qquad$ $A[k,j]$.

$i, i+1, \ldots, k$ $\qquad$ $k, \ldots, j$

$i<j$

$$A[i,j] = \min_k \{ A[i,k] + A[k,j] + \underbrace{w_{i-k} + w_{k-j} + w_{j-i}}\}$$

Returns 0 $\leftarrow$ $j=i+1$ $\longrightarrow$ Line $\longrightarrow$ Zero diagonals = 0

Returns sum of wts. $\leftarrow$ $j=i+2$ $\longrightarrow$ Triangle $\longrightarrow$ One diagonal $\longrightarrow$ Part of each Δle.

$j=i+3$ $\longrightarrow$

$\hookrightarrow$ Will depend on cases with $j-i \leq 2$.

$\hookrightarrow$ Already computed (& Memoized).

Time complexity

$\rightarrow$ For every i,j $\rightarrow$ No. of lookups $j-i$

$$\sum_{i,j\,:\,i<j} (j-i) = \sum_{i=1}^{n} \sum_{j>i}^{n} (j-i) = \sum_i \left(1+2+\ldots+(n-i)\right)$$

$\Downarrow$ $j=i+1$

$$= \sum_i \left(\frac{(n-i)}{2}\left(2+(n-i-1)\right)\right)$$

$$= \sum_{i=1}^{n} (n-i) + \sum_{i=1}^{n} (n-i)(n-i-1)$$

$$= (0+1+\dots+n-1) + \sum_{i=1}^{n} (n^2+n(-2i-1)+i^2-i)$$

$$= \frac{n}{2}(n-1)$$

- **Edit Distance**

Q: Min. entries to be changed to get the other word.

$a_1, a_2, \dots, a_m$

$b_1, b_2, \dots, b_n$.

A R U N
A R U L . Dist. = 1

A A R O N
A R O N —

3 + 1 dist.

(OR)

A A R O N
A R O N

1 dist.

OPTIMAL SEQUENCE ALIGNMENT

C H A N D R A
S A N D R A

Dist = 2

→ $A[1,m]$, $B[1,n]$ → How to edit these both such that the edit distance is minimum.

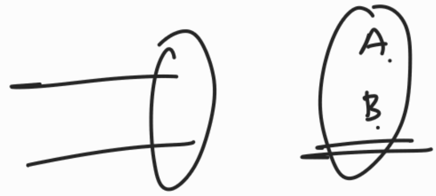
Base case :

α
β

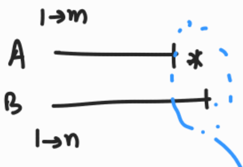
If $\alpha = \beta$, edit = 0

$\alpha \neq \beta$. edit = 1

Look at the last char.

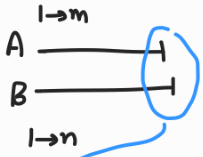


Move A to left



$\text{Edit}(A[1,m], B[1,n-1])$ $+1$

Match end pts.

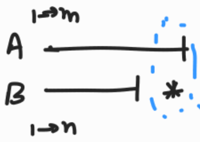


$\text{Edit}(A[1,m-1], B[1,n-1])$ $+b$

where $b=1$ if $A_m \neq B_n$

$b=0$ if $A_m = B_n$

Move A to right



$\text{Edit}(A[1,m-1], B[1,n]) + 1$

$$\Delta(A[1,m], B[1,n]) = \min \begin{cases} \Delta(A[1,m], B[1,n-1]) + 1 \\ \Delta(A[1,m-1], B[1,n]) + 1 \\ \Delta(A[1,m-1], B[1,n-1]) + b \end{cases}$$

Match ends

C H A N D R A

S   A N D R A

Dist = 2

CH / S   CH / S

Here we're taking care of suffix & then subproblem on prefix.

(Right to left).

We can do left to right also

Left to right:

A L G A E

A L G O S

A | L G A E
_ | A L G O S
+1

L G A E
L G O S.

_ | A L G A E
A | L G O S
+1

Right to left: