WEEK 5, LECTURE 10 ON 22 SEPTEMBER 2021 CS1.301.M21 ALGORITHM ANALYSIS AND DESIGN

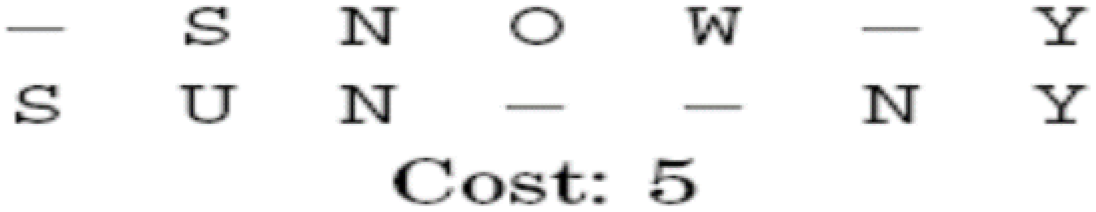
EDIT DISTANCE

Problem

Possible operations that count as an edit: Insert, Delete or Overwrite.
The Edit Distance between two strings is the minimal number of operations required to convert one string to the other.

One intuitive way of measuring the edit distance is by observing how many characters in each string match up or *align* after adding some blanks in the words.
And so the edit distance will be the number of columns that are mismatched in the alignment with least possible mismatches.

Eg.

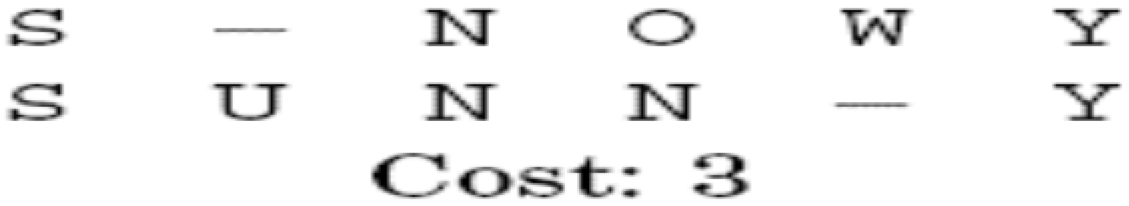


—	S	N	O	W	—	Y
S	U	N	—	—	N	Y

Cost: 5

Here, a blank-char mismatch is an insert, a char-blank mismatch is a delete and a char-char mismatch is an overwrite.

This example can be aligned with less mismatches as shown:



S	—	N	O	W	Y
S	U	N	N	—	Y

Cost: 3

Answer

We can use dynamic programming to solve this.

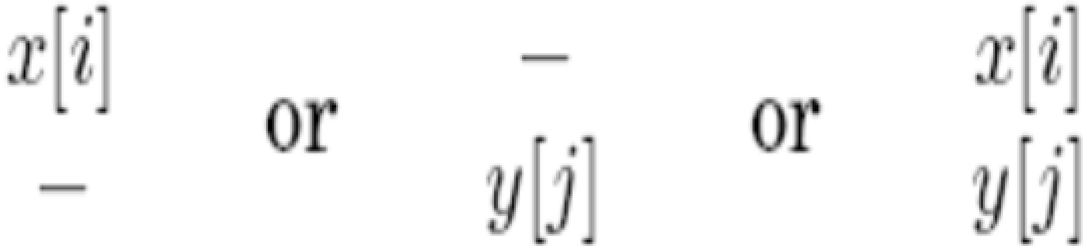
So what are the subproblems?

The goal being to find the edit distance between the strings $x[1 \ldots m]$ and $y[1 \ldots n]$

Suppose a subproblem $E(i, j)$ as the edit distance between certain prefixes of x and y ending in i and j.

An observation:

The rightmost column can only be one of the following:



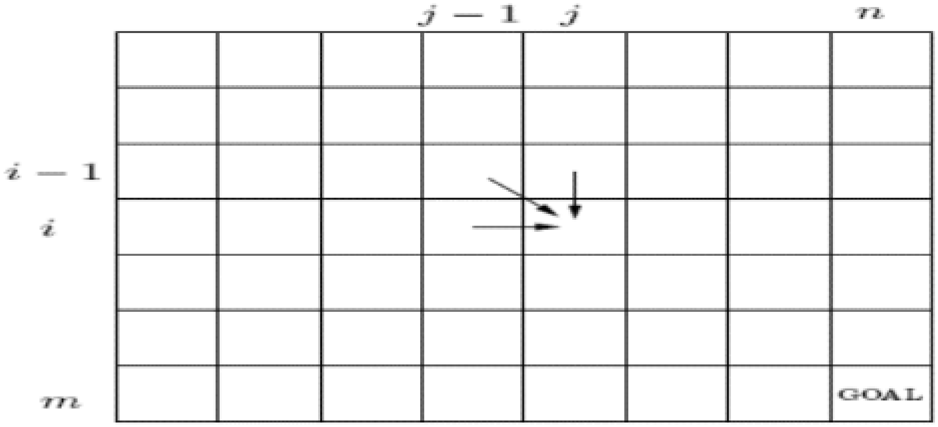
$$\begin{matrix} x[i] \\ - \end{matrix} \quad \text{or} \quad \begin{matrix} - \\ y[j] \end{matrix} \quad \text{or} \quad \begin{matrix} x[i] \\ y[j] \end{matrix}$$

So we can recursively define $E(i, j)$ as

$$E(i, j) = \min\{1 + E(i-1, j), 1 + E(i, j-1), \text{diff}(i, j) + E(i-1, j-1)\}$$

> $\text{diff}(i, j)$ is 0 if the characters are equal and 1 otherwise

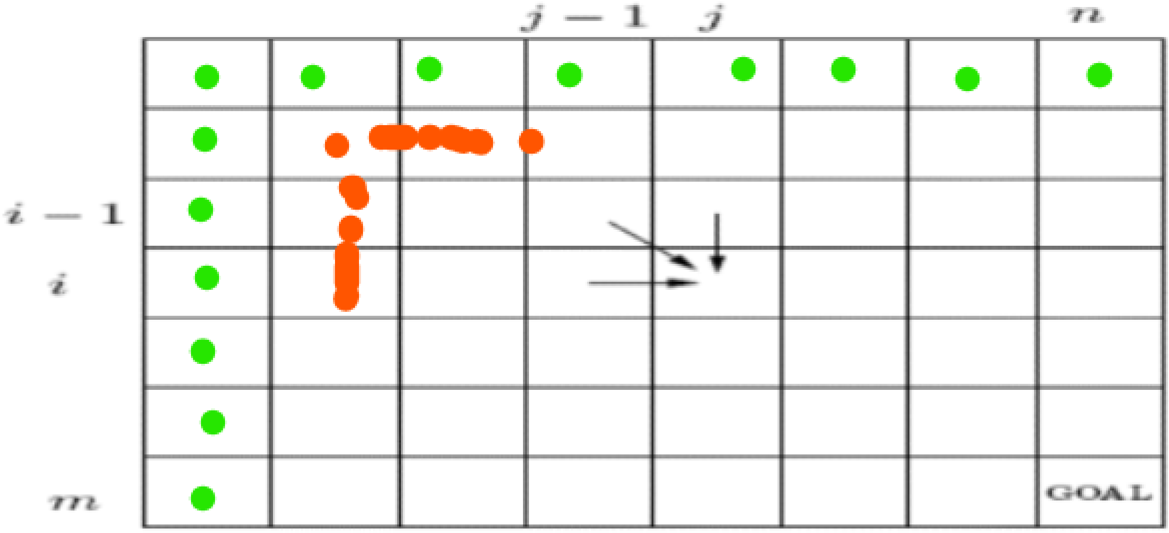
The DP table is:



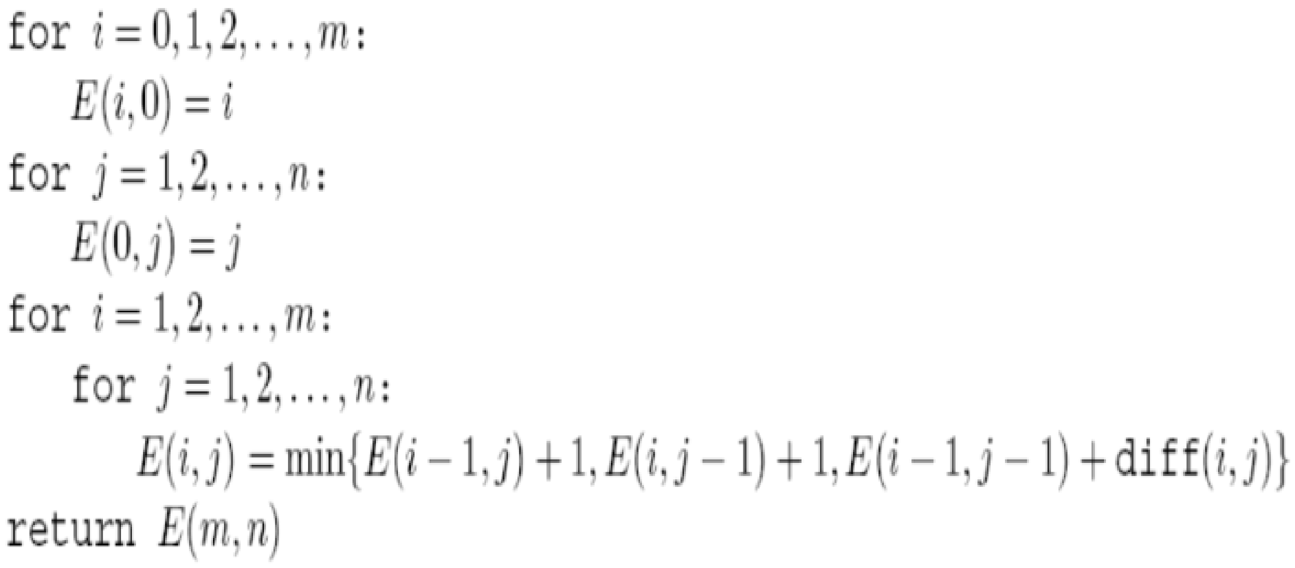
So to find the final answer we recursively find the answers for the subproblems.

Now the question becomes: In what order do we solve the subproblems?

We know that the entire column and row of $[0, j]$ and $[i, 0]$ are i and j. So we can start at $[1, 1]$ and solve row-wise or column-wise till the final answer. The DP table becomes:

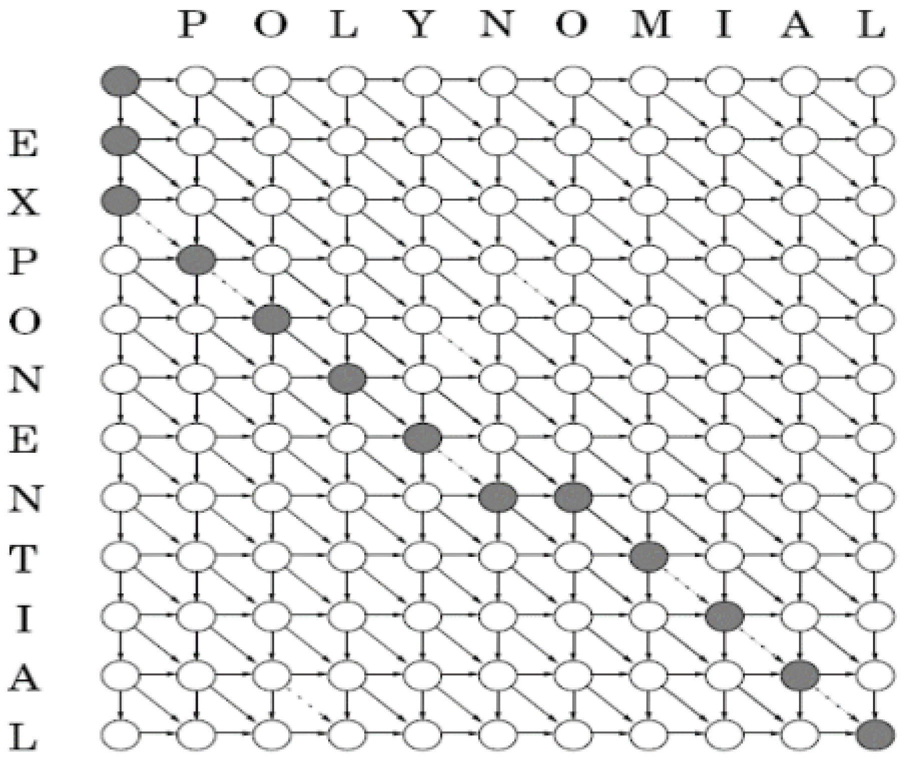


The algorithm then is:



```
for $i = 0,1,2,\ldots,m$:
    $E(i,0) = i$
for $j = 1,2,\ldots,n$:
    $E(0,j) = j$
for $i = 1,2,\ldots,m$:
    for $j = 1,2,\ldots,n$:
        $E(i,j) = \min\{E(i-1,j)+1, E(i,j-1)+1, E(i-1,j-1)+\mathtt{diff}(i,j)\}$
return $E(m,n)$
```

An example DAG for `POLYNOMIAL` and `EXPONENTIAL` is:



So in total the cost of the algorithm is $O(mn)$

PROJECT

1. The direction of benefit taken by project/course/life/job/whatever. Do you wish to:
    - Get something
    - Give something
    - Get More by giving
    - Give more by getting
2. The driving force finds out what your motivation is behind a decision. Why did you take this course:
    - To learn about algorithms
    - To become an algorithmist
    - To get done with the course
    - I like (spending time on) algorithms
3. The Qualitative purpose question tells us what qualities we expect. What do you expect from this course:
    - The Ability/Skill to solve problems
    - Exposure and opportunity to contribute
    - To prepare your mind for creativity
    - Nothing (i just like algorithms)
4. The executive operation question shows how we are going about something. How would you (preferably) execute this course:
    - Solve/implement (a lot of) problems
    - Think a lot and discover
    - Read a lot of literature
    - Listen to a lot of lectures
5. The philosophical question. Life and existence appear to you as:
    - Steady evolution of real chemical scum
    - Chancy superposition of abstract wave-functions
    - Unending splendor of surreal beauty
    - Virtual simulation of Universal Intelligence