Chapter 26: Edit Distance Dynamic Algorithm

Section 26.1: Minimum Edits required to convert string 1 to string 2

The problem statement is like if we are given two string str1 and str2 then how many minimum number of operations can be performed on the str1 that it gets converted to str2.The Operations can be:

1. **Insert**
2. **Remove**
3. **Replace**

For Example

```
Input: str1 = "geek", str2 = "gesek"
Output: 1
We only need to insert s in first string

Input: str1 = "march", str2 = "cart"
Output: 3
We need to replace m with c and remove character c and then replace h with t
```

To solve this problem we will use a 2D array dp[n+1][m+1] where n is the length of the first string and m is the length of the second string. For our example, if str1 is **azcef** and str2 is **abcdef** then our array will be dp[6][7]and our final answer will be stored at dp[5][6].

```
          (a) (b) (c) (d) (e) (f)
+---+---+---+---+---+---+---+
| 0 | 1 | 2 | 3 | 4 | 5 | 6 |
+---+---+---+---+---+---+---+
(a)| 1 |   |   |   |   |   |   |
+---+---+---+---+---+---+---+
(z)| 2 |   |   |   |   |   |   |
+---+---+---+---+---+---+---+
(c)| 3 |   |   |   |   |   |   |
+---+---+---+---+---+---+---+
(e)| 4 |   |   |   |   |   |   |
+---+---+---+---+---+---+---+
(f)| 5 |   |   |   |   |   |   |
+---+---+---+---+---+---+---+
```

For **dp[1][1]** we have to check what can we do to convert **a** into **a**.It will be **0**.For **dp[1][2]** we have to check what can we do to convert **a** into **ab**.It will be **1** because we have to **insert b**.So after 1st iteration our array will look like

```
          (a) (b) (c) (d) (e) (f)
+---+---+---+---+---+---+---+
| 0 | 1 | 2 | 3 | 4 | 5 | 6 |
+---+---+---+---+---+---+---+
(a)| 1 | 0 | 1 | 2 | 3 | 4 | 5 |
+---+---+---+---+---+---+---+
(z)| 2 |   |   |   |   |   |   |
+---+---+---+---+---+---+---+
(c)| 3 |   |   |   |   |   |   |
+---+---+---+---+---+---+---+
(e)| 4 |   |   |   |   |   |   |
+---+---+---+---+---+---+---+
(f)| 5 |   |   |   |   |   |   |
+---+---+---+---+---+---+---+
```

For iteration 2

For **dp[2][1]** we have to check that to convert **az** to **a** we need to remove **z**, hence **dp[2][1]** will be **1**.Similary for **dp[2][2]** we need to replace **z** with **b**, hence **dp[2][2]** will be **1**.So after 2nd iteration our **dp[]** array will look like.

```
         (a) (b) (c) (d) (e) (f)
+---+---+---+---+---+---+---+
| 0 | 1 | 2 | 3 | 4 | 5 | 6 |
+---+---+---+---+---+---+---+
(a)| 1 | 0 | 1 | 2 | 3 | 4 | 5 |
+---+---+---+---+---+---+---+
(z)| 2 | 1 | 1 | 2 | 3 | 4 | 5 |
+---+---+---+---+---+---+---+
(c)| 3 |   |   |   |   |   |   |
+---+---+---+---+---+---+---+
(e)| 4 |   |   |   |   |   |   |
+---+---+---+---+---+---+---+
(f)| 5 |   |   |   |   |   |   |
+---+---+---+---+---+---+---+
```

So our **formula** will look like

```
if characters are same
    dp[i][j] = dp[i-1][j-1];
else
    dp[i][j] = 1 + Min(dp[i-1][j], dp[i][j-1], dp[i-1][j-1])
```

After last iteration our dp[] array will look like

```
          (a) (b) (c) (d) (e) (f)
+---+---+---+---+---+---+---+
| 0 | 1 | 2 | 3 | 4 | 5 | 6 |
+---+---+---+---+---+---+---+
(a)| 1 | 0 | 1 | 2 | 3 | 4 | 5 |
+---+---+---+---+---+---+---+
(z)| 2 | 1 | 1 | 2 | 3 | 4 | 5 |
+---+---+---+---+---+---+---+
(c)| 3 | 2 | 2 | 1 | 2 | 3 | 4 |
+---+---+---+---+---+---+---+
(e)| 4 | 3 | 3 | 2 | 2 | 2 | 3 |
+---+---+---+---+---+---+---+
(f)| 5 | 4 | 4 | 2 | 3 | 3 | 3 |
+---+---+---+---+---+---+---+
```

Implementation in Java

```java
public int getMinConversions(String str1, String str2){
    int dp[][] = new int[str1.length()+1][str2.length()+1];
    for(int i=0;i<=str1.length();i++){
        for(int j=0;j<=str2.length();j++){
            if(i==0)
```

```
                dp[i][j] = j;
            else if(j==0)
                dp[i][j] = i;
            else if(str1.charAt(i-1) == str2.charAt(j-1))
                dp[i][j] = dp[i-1][j-1];
            else{
                dp[i][j] = 1 + Math.min(dp[i-1][j], Math.min(dp[i][j-1], dp[i-1][j-1]));
            }
        }
    }
    return dp[str1.length()][str2.length()];
}
```

Time Complexity

```
O(n^2)
```