Minimum Cost To Make Two Strings Identical

Given two strings x and y, and two values costX and costY, the task is to find the minimum cost required to make the given two strings identical. You can delete characters from both the strings. The cost of deleting a character from string X is costX and from Y is costY. The cost of removing all characters from a string is the same.

Example 1:

Input: x = "abcd", y = "acdb", costX = 10

costY = 20.

Output: 30

Explanation: For Making both strings

identical we have to delete character

'b' from both the string, hence cost

will be = 10 + 20 = 30.

Example 2:

Input : x = "ef", y = "gh", costX = 10

costY = 20.

Output: 60

Explanation: For making both strings

identical, we have to delete 2-2

characters from both the strings, hence

cost will be = 10 + 10 + 20 + 20 = 60.

Your Task:

You don't need to read or print anything. Your task is to complete the function findMinCost() which takes both strings and the costs as input parameters and returns the minimum cost.

Expected Time Complexity: O(|x|\*|y|)

Expected Space Complexity: O(|x|\*|y|)

Constraints:

1 ≤ |x|, |y| ≤ 1000

1<= costX, costY <= 105

Code :

class Solution {

public:

int findMinCost(string x, string y, int costX, int costY) {

int n = x.size();

int m = y.size();

vector<int> dp0(m+1, 0), dp1(m+1,0);

for(int i{1};i<=n;i++)

{

for(int j{1};j<=m;j++)

{

if(x[i-1] == y[j-1]) dp1[j]=dp0[j-1]+1;

else dp1[j] = max(dp1[j-1], dp0[j]);

}

dp0 = dp1;

}

return costX \* (n-dp1[m]) + costY \* (m-dp1[m]);

}

};

Link : https://www.geeksforgeeks.org/problems/minimum-cost-to-make-two-strings-identical1107/1