Practical 5

Name-Huzaif Baig

Class-A-4

Batch-B3

Aim-

(i) DNA sequences can be viewed as strings of A, C, G, and T characters, which represent nucleotides. Finding the similarities between two DNA sequences are an important computation performed in bioinformatics.

[Note that a subsequence might not include consecutive elements of the original sequence.]

TASK 1: Find the similarity between the given X and Y sequence.

X=AGCCCTAAGGGCTACCTAGCTT

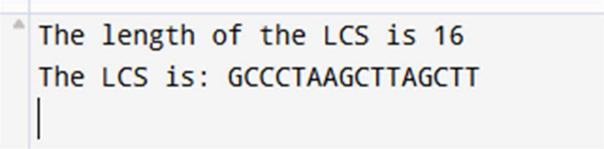
Y= GACAGCCTACAAGCGTTAGCTTG

Output: Cost matrix with all costs and direction, final cost of LCS and the LCS.

Length of LCS=16

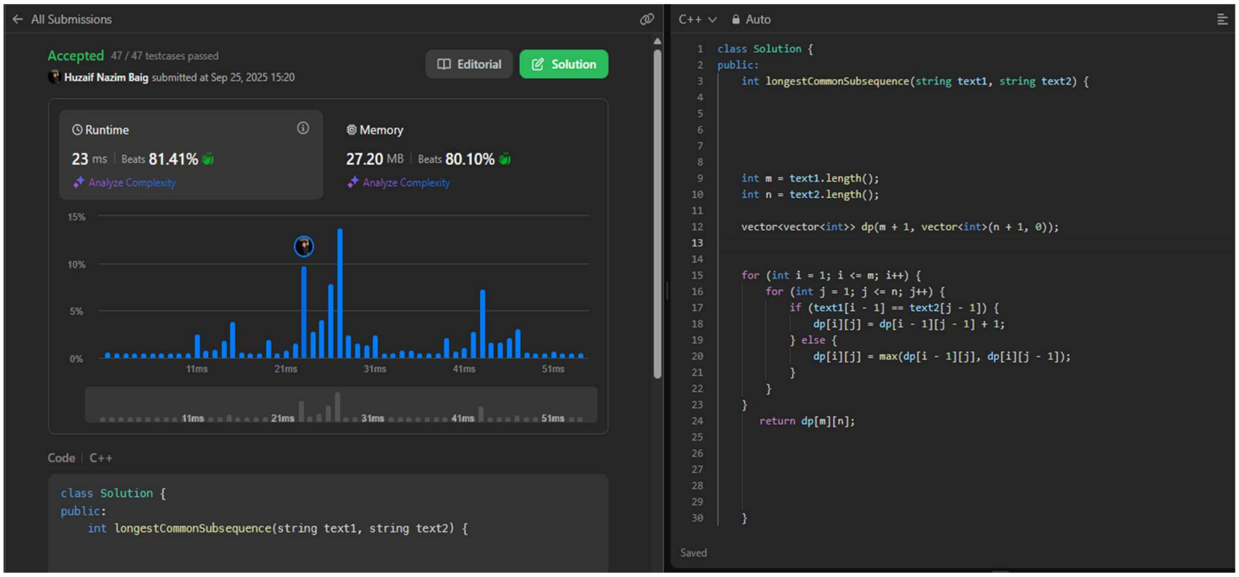
```
#include <iostream>
#include <string>
using namespace std;
#include <iostream>
#include <string>
using namespace std;

string printLCS(string x, string y) {
    int m = x.length();
    int n = y.length();
    int dp[100][100] = {0};

    // Fill dp table
    for (int i = 1; i <= m; i++) {
        for (int j = 1; j <= n; j++) {
            if (x[i - 1] == y[j - 1]) {
                dp[i][j] = dp[i - 1][j - 1] + 1;
            } else {
                dp[i][j] = max(dp[i - 1][j], dp[i][j - 1]);
            }
        }
    }

    // Reconstruct LCS from dp table
    int i = m, j = n;
    string lcsStr = "";

    while (i > 0 && j > 0) {
        if (x[i - 1] == y[j - 1]) {
            lcsStr = x[i - 1] + lcsStr;
            i--;
            j--;
        } else if (dp[i - 1][j] > dp[i][j - 1]) {
            i--;
        } else {
            j--;
        }
    }

    return lcsStr;
}

int main() {
    string x = "AGCCCTAAGGGCTACCTAGCTT";
    string y = "GACAGCCTACAAGCGTTAGCTTG";
    string lcsStr = printLCS(x, y);
    cout << "The length of the LCS is " << lcsStr.length() << endl;
    cout << "The LCS is: " << lcsStr << endl;


    return 0;
}
```

Output



```
The length of the LCS is 16
The LCS is: GCCCTAAGCTTAGCTT
```

Leetcode Problem-



Accepted 47 / 47 testcases passed

Huzaif Nazim Baig submitted at Sep 25, 2025 15:20

Runtime 23 ms | Beats 81.41%

Memory 27.20 MB | Beats 80.10%

Code | C++

```
class Solution {
public:
    int longestCommonSubsequence(string text1, string text2) {
```

```
class Solution {
public:
    int longestCommonSubsequence(string text1, string text2) {





    int m = text1.length();
    int n = text2.length();

    vector<vector<int>> dp(m + 1, vector<int>(n + 1, 0));


    for (int i = 1; i <= m; i++) {
        for (int j = 1; j <= n; j++) {
            if (text1[i - 1] == text2[j - 1]) {
                dp[i][j] = dp[i - 1][j - 1] + 1;
            } else {
                dp[i][j] = max(dp[i - 1][j], dp[i][j - 1]);
            }
        }
    }
      return dp[m][n];






    }
```

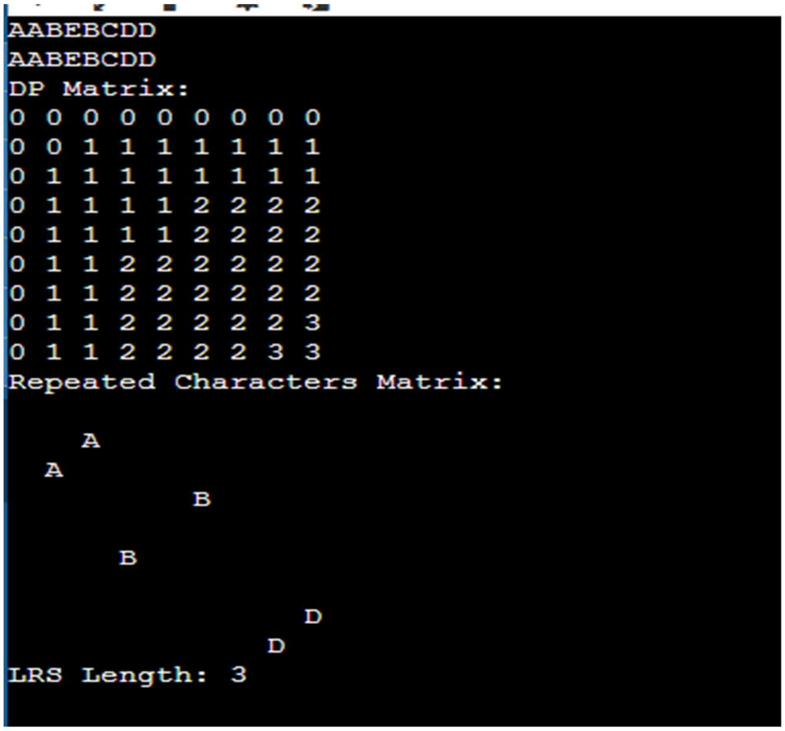
Saved

LRS Part 2

Code-

```
#include <iostream>

#include <vector>

#include <string>

using namespace std;


int main() {

    string s1, s2;

    cin >> s1 >> s2;


    int n = s1.length();

    vector<vector<int>> dp(n + 1, vector<int>(n + 1, 0));

    vector<vector<char>> result(n + 1, vector<char>(n + 1, ' '));


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

        for (int j = 1; j <= n; j++) {

            if (s1[i-1] == s2[j-1] && i != j) {

                dp[i][j] = 1 + dp[i-1][j-1];

                result[i][j] = s1[i-1];

            } else {

                dp[i][j] = max(dp[i-1][j], dp[i][j-1]);

                result[i][j] = ' ';

            }

        }

    }


    cout << "DP Matrix:" << endl;

    for (int i = 0; i <= n; i++) {

        for (int j = 0; j <= n; j++) {

            cout << dp[i][j] << " ";
        }
        cout << endl;
    }

    cout << "Repeated Characters Matrix:" << endl;
    for (int i = 0; i <= n; i++) {
        for (int j = 0; j <= n; j++) {
            cout << result[i][j] << " ";
        }
        cout << endl;
    }

    cout << "LRS Length: " << dp[n][n] << endl;

    return 0;
}
```

Output-



```
AABEBCDD
AABEBCDD
DP Matrix:
0 0 0 0 0 0 0 0 0
0 0 1 1 1 1 1 1 1
0 1 1 1 1 1 1 1 1
0 1 1 1 1 2 2 2 2
0 1 1 1 1 2 2 2 2
0 1 1 2 2 2 2 2 2
0 1 1 2 2 2 2 2 2
0 1 1 2 2 2 2 2 3
0 1 1 2 2 2 2 3 3
Repeated Characters Matrix:

    A
  A
          B

      B

                D
              D
LRS Length: 3
```