Q3 Implement Longest Common Subsequence. Analyze its time complexity.

TIME COMPLEXITY: O(m * n)

SPACE COMPLEXITY: O(m * n)

where m and n are the length of the input strings.

```
#include <iostream>
#include <vector>
#include <random>
#include <chrono>

using namespace std;


int lcsDP(string & s1, string & s2)
{
    int m = int(s1.size()), n = int(s2.size());
    int ** dp = new int*[m + 1];
    for(int i = 0; i < m + 1; i++)
        dp[i] = new int[n + 1];
    for(int i = 0; i < m + 1; i++)
        dp[i][0] = 0;
    for(int i = 0; i < n + 1; i++)
        dp[0][i] = 0;
    for(int i = 1; i < m + 1; i++)
    {
        for(int j = 1; j < n + 1; j++)
        {
            if(s1[m - i] == s2[n - j])
                dp[i][j] = 1 + dp[i - 1][j - 1];
            else dp[i][j] = max(dp[i - 1][j], dp[i][j - 1]);
        }
    }
    int i = m, j = n;
    string lcs = "";
    while(i > 0 && j > 0)
    {
        if(s1[m - i] == s2[n - j])
        {
            lcs.push_back(s1[m - i]);
            i--;
            j--;
        }
        else if(dp[i - 1][j] > dp[i][j - 1])
            i--;
        else j--;
    }
    cout << "LONGEST COMMON SUBSEQUENCE: " << lcs << endl;
    cout << "LENGTH OF THE LCS: " << lcs.size() << endl;
    return dp[m][n];
}
```

```
int main()
{
    string s1 =
     "nohgdazaxavkhylqvghqpifijohudenozotejoxavkfkzcdqnoxydynavwdyl";
    string s2 = "nohgdazargdcbmbgjcvqpcbadujkxaxujudmbejcrevuvcdobo";
    cout << "S1: " << s1 << endl;
    cout << "S2: " << s2 << endl;
    lcsDP(s1, s2);
}
```

OUTPUT:

```
S1: nohgdazaxavkhylqvghqpifijohudenozotejoxavkfkzcdqnoxydynavwdyl
S2: nohgdazargdcbmbgjcvqpcbadujkxaxujudmbejcrevuvcdobo
LONGEST COMMON SUBSEQUENCE: nohgdazavqpjudeevcdo
LENGTH OF THE LCS: 20
Program ended with exit code: 0
```