**SOLUTION**

class Solution {

public:

Solution(){ios::sync\_with\_stdio(false); std::cin.tie(nullptr); std::cout.tie(nullptr);}

int longestCommonSubsequence(string s1, string s2) {

int m=s1.size();

int n=s2.size();

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++)

dp[i][j]=s1[i-1]==s2[j-1]?dp[i-1][j-1]+1:max(dp[i][j-1],dp[i-1][j]);

return dp[m][n];

}

};

**TIME COMPLEXITY: O(M\*N)**

**SPACE COMPLEXITY: O(M\*N)**