Green University of Bangladesh
Department of Computer Science and Engineering (CSE)

Faculty of Sciences and Engineering

Semester: (Fall, Year:2024), B.Sc. in CSE (Day)

Lab Report NO : 05

Course Title: Algorithms Lab

Course Code: CSE 208 Section: D2

Lab Experiment Name: Implementation of Longest Common Subsequence (LCS) algorithm.

Student Details

	Name	ID
1.	Promod Chandra Das	231002005

Lab Date : 04-12-2024

Submission Date : 11-12-2024

Course Teacher's Name : Farjana Akter Jui

Lab Report Status

Marks: …… **Signature:.....**

Comments:..... **Date:.....**

1. TITLE OF THE LAB REPORT EXPERIMENT :

Implementation of Longest Common Subsequence (LCS) algorithm.

2. OBJECTIVES :

- To grasp the concept of the Longest Common Subsequence (LCS) algorithm and explore its applications.
- To develop an implementation of the LCS algorithm to identify all common subsequences between two sequences.
- To arrange and display the common subsequences in descending order of their lengths.
- To assess the algorithm's efficiency with respect to both time and space complexity..

3. PROCEDURE :

- **Input the Sequences:** Accept the two input sequences, 'ABCDEFGH' and 'abcdefgh'.
- **Normalize the Case:** Convert both sequences to the same case (if necessary) to ensure case insensitivity is handled.
- **Construct the LCS Table:** Use dynamic programming to build a 2D array (dp), where dp[i][j] holds the length of the Longest Common Subsequence (LCS) for the first i characters of the first sequence and the first j characters of the second sequence.
- **Backtrack to Identify Subsequences:** Traverse the dp table to identify and collect all possible common subsequences.
- **Store and Sort Subsequences:** Save the identified subsequences in a set to eliminate duplicates, then sort them by length in descending order.
- **Display the Results:** Output all the subsequences along with their lengths, sorted in descending order.

4. IMPLEMENTATION :

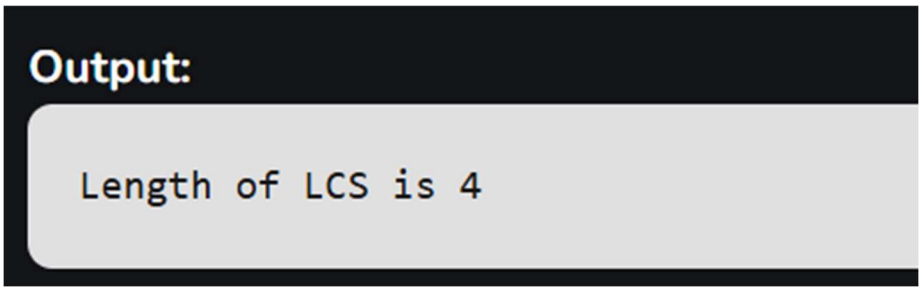
Here is the full java code Implementation of Longest Common Subsequence

```java
public class LongestCommonSubsequence {


    int lcs(char[] X, char[] Y, int m, int n)
    {
        if (m == 0 || n == 0)
            return 0;
        if (X[m - 1] == Y[n - 1])
            return 1 + lcs(X, Y, m - 1, n - 1);
        else
            return max(lcs(X, Y, m, n - 1), lcs(X, Y, m - 1, n));
    }


    int max(int a, int b)
    {
        return (a > b) ? a : b;
    }

    public static void main(String[] args)
```

```java
    }

    public static void main(String[] args)
    {
        LongestCommonSubsequence lcs = new LongestCommonSubsequence();
        String s1 = "AGGTAB";
        String s2 = "GXTXAYB";

        char[] X = s1.toCharArray();
        char[] Y = s2.toCharArray();
        int m = X.length;
        int n = Y.length;

        System.out.println("Length of LCS is"
                           + " " + lcs.lcs(X, Y, m, n));
    }
}
```

5. OUTPUT :

Here is the output implementation of KMP Algorithm in case of integer



6. DISCUSSION :

The Java implementation of the Longest Common Subsequence (LCS) algorithm utilizes dynamic programming to efficiently find the longest subsequence that two sequences share. The algorithm constructs a 2D array where each entry represents the LCS length for specific subsections of the input strings. By backtracking through this table, the algorithm identifies all common subsequences. Additionally, the code handles case insensitivity by normalizing the input sequences. Once the common subsequences are found, they are stored in a set to eliminate duplicates and sorted by length in descending order before being displayed. This approach ensures an optimal time complexity of O(m * n), where m and n are the lengths of the input sequences.