# Practical No. 5

## Name: Rana Chavan
## Section: A4 – B2
## Roll no. 26

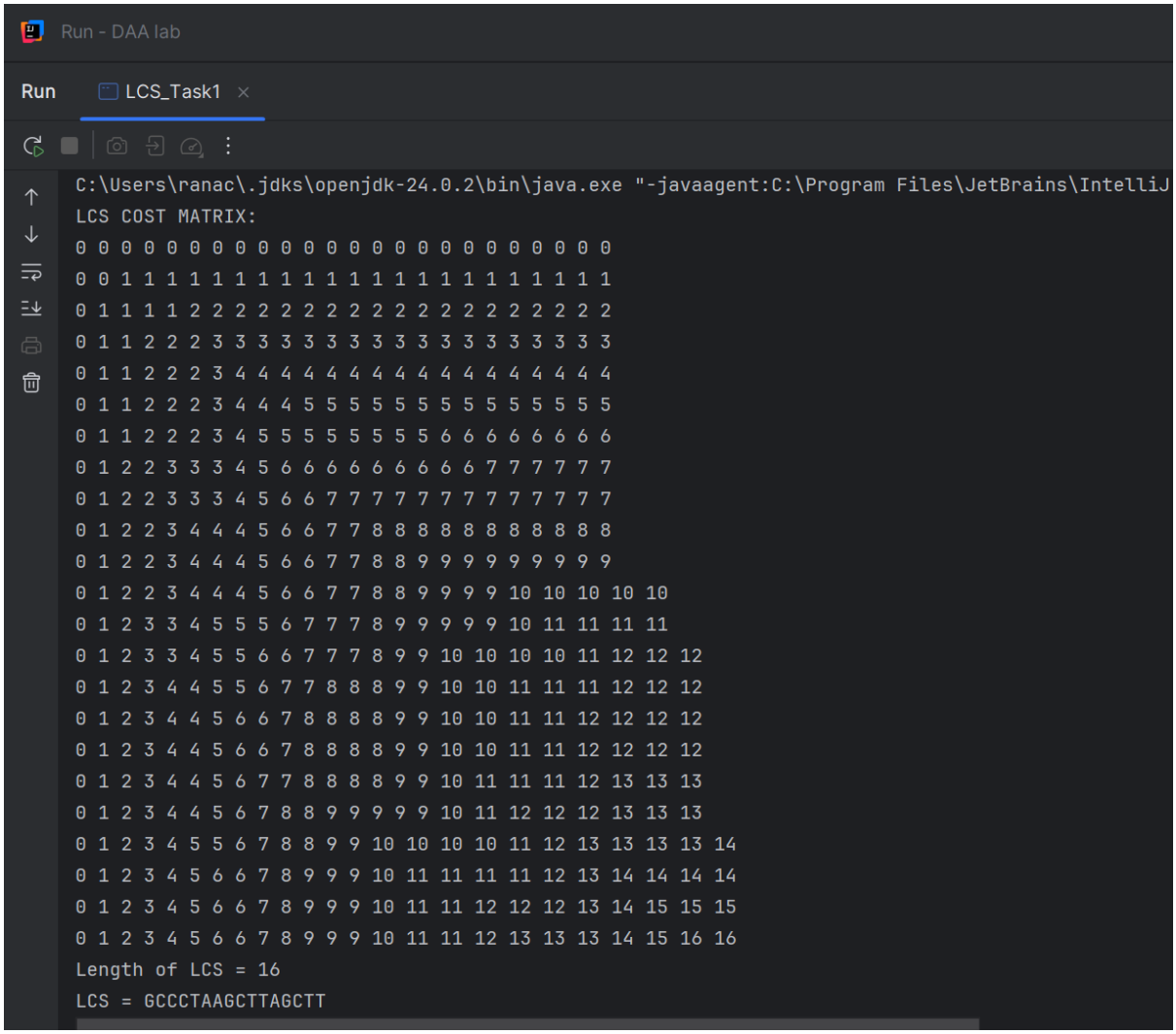
**Aim:** Implement a dynamic algorithm for Longest Common Subsequence (LCS) to find the length and LCS for DNA sequences.

**Task 1:**

**CODE:**

```
public class LCS_Task1 {
    public static void main(String[] args) {
        String X = "AGCCCTAAGGGCTACCTAGCTT";
        String Y = "GACAGCCTACAAGCGTTAGCTTG";

        int m = X.length();
        int n = Y.length();
        int[][] dp = new int[m + 1][n + 1];

        for (int i = 1; i <= m; i++) {
            for (int j = 1; j <= n; j++) {
                if (X.charAt(i - 1) == Y.charAt(j - 1))
                    dp[i][j] = dp[i - 1][j - 1] + 1;
                else
                    dp[i][j] = Math.max(dp[i - 1][j], dp[i][j - 1]);
            }
        }

        System.out.println("LCS COST MATRIX:");
        for (int i = 0; i <= m; i++) {
            for (int j = 0; j <= n; j++) {
                System.out.print(dp[i][j] + " ");
            }
            System.out.println();
        }

        System.out.println("Length of LCS = " + dp[m][n]);

        int i = m, j = n;
        String lcs = "";
        while (i > 0 && j > 0) {
            if (X.charAt(i - 1) == Y.charAt(j - 1)) {
                lcs = X.charAt(i - 1) + lcs;
                i--;
                j--;
            } else if (dp[i - 1][j] > dp[i][j - 1])
                i--;
            else
                j--;
        }

        System.out.println("LCS = " + lcs);
    }
}
```

**OUTPUT:**




**Task 2:**

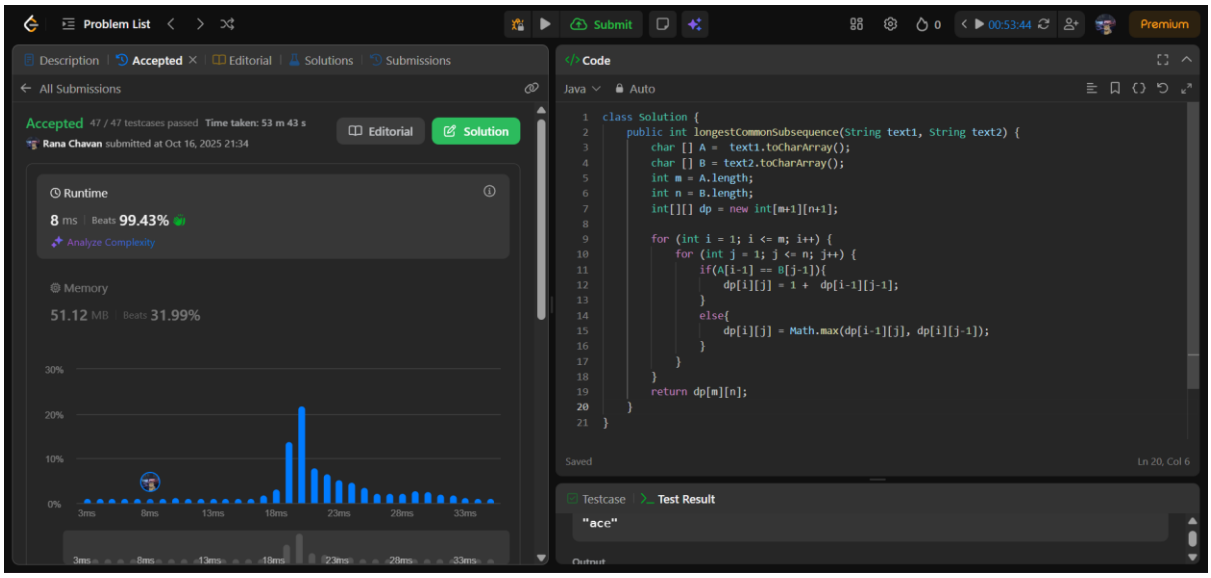
**CODE:**

```
public class LRS_Task2 {
    public static void main(String[] args) {
        String S = "AABCBDC";
        int n = S.length();
        int[][] dp = new int[n + 1][n + 1];

        for (int i = 1; i <= n; i++) {
            for (int j = 1; j <= n; j++) {
                if (S.charAt(i - 1) == S.charAt(j - 1) && i != j)
                    dp[i][j] = dp[i - 1][j - 1] + 1;
                else
                    dp[i][j] = Math.max(dp[i - 1][j], dp[i][j - 1]);
            }
        }
```

```
        System.out.println("LRS MATRIX:");
        for (int i = 0; i <= n; i++) {
            for (int j = 0; j <= n; j++) {
                System.out.print(dp[i][j] + " ");
            }
            System.out.println();
        }

        System.out.println("Length of LRS = " + dp[n][n]);

        int i = n, j = n;
        String lrs = "";
        while (i > 0 && j > 0) {
            if (S.charAt(i - 1) == S.charAt(j - 1) && i != j) {
                lrs = S.charAt(i - 1) + lrs;
                i--;
                j--;
            } else if (dp[i - 1][j] > dp[i][j - 1])
                i--;
            else
                j--;
        }

        System.out.println("LRS = " + lrs);
    }
}
```

**OUTPUT:**

**LeetCode Assesment:**




**Profile Link:** https://leetcode.com/u/ranachavan/

**Submission detail:** https://leetcode.com/submissions/detail/1803509703/