Code for Longest Common Subsequence Algorithm

import java.util.\*;

public class EXPT\_10\_Longest\_Common\_Subsequence {

    public static void main(String args[]) {

        Scanner in = new Scanner(System.in);

        System.out.print("Enter the first string: ");

        String str1 = in.nextLine();

        System.out.print("\nEnter the second string: ");

        String str2 = in.nextLine();

        int str1\_l = str1.length();

        int str2\_l = str2.length();

        int lcs[][] = new int[str2\_l + 1][str1\_l + 1];

        String backtrack[][] = new String[str2\_l + 1][str1\_l + 1];

        Arrays.fill(lcs[0], 0);

        for (int i = 0; i < lcs.length; i++) {

            lcs[i][0] = 0;

            Arrays.fill(backtrack[i], "?");

        }

        EXPT\_10\_Longest\_Common\_Subsequence obj = new EXPT\_10\_Longest\_Common\_Subsequence();

        long start = System.nanoTime();

        obj.Longest\_Common\_Subsequence(lcs, backtrack, str1, str2);

        long end = System.nanoTime();

        System.out.print("\nLCS Table: ");

        obj.print(lcs, str1, str2);

        System.out.print("\nThe Longest Common Subsequence is ");

        obj.subsequence(backtrack, str2\_l, str1\_l, str2);

        System.out.print("\nThe length of Longest Common Subsequence is " + lcs[str2\_l][str1\_l]);

        System.out.print("\nThe time taken by Longest Common Subsequence Algorithm\nwhen length of first string is "

                + str1\_l + " and\nlength of second string is " + str2\_l + " is " + (end - start) + " nanoseconds.");

    }

    void print(int arr\_2D[][], String str1, String str2) {

        String st1 = '0' + str1, st2 = '0' + str2;

        System.out.print("\n    ");

        for (int i = 0; i < st1.length(); i++)

            System.out.print(String.format("%3s", st1.charAt(i)));

        System.out.println();

        for (int i = 0; i < arr\_2D.length; i++) {

            if (i < st2.length())

                System.out.print(String.format("%3s", st2.charAt(i)));

            for (int j = 0; j < arr\_2D[i].length; j++)

                System.out.print(String.format("%3d", arr\_2D[i][j]));

            System.out.println();

        }

    }

    void Longest\_Common\_Subsequence(int lcs[][], String backtrack[][], String str1, String str2) {

        for (int i = 1; i < lcs.length; i++) {

            for (int j = 1; j < lcs[i].length; j++) {

                if (str2.charAt(i - 1) == str1.charAt(j - 1)) {

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

                    backtrack[i][j] = "~";

                } else if (lcs[i - 1][j] >= lcs[i][j - 1]) {

                    lcs[i][j] = lcs[i - 1][j];

                    backtrack[i][j] = "^";

                } else {

                    lcs[i][j] = lcs[i][j - 1];

                    backtrack[i][j] = "<-";

                }

            }

        }

    }

    void subsequence(String backtrack[][], int i, int j, String str1) {

        if (i == 0 || j == 0)

            return;

        if (backtrack[i][j] == "~") {

            subsequence(backtrack, i - 1, j - 1, str1);

            System.out.print(str1.charAt(i - 1));

        } else if (backtrack[i][j] == "^") {

            subsequence(backtrack, i - 1, j, str1);

        } else {

            subsequence(backtrack, i, j - 1, str1);

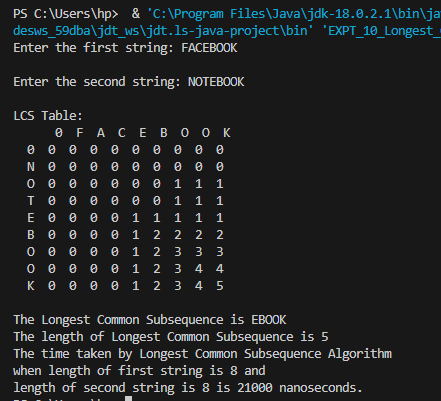
        }

    }

}

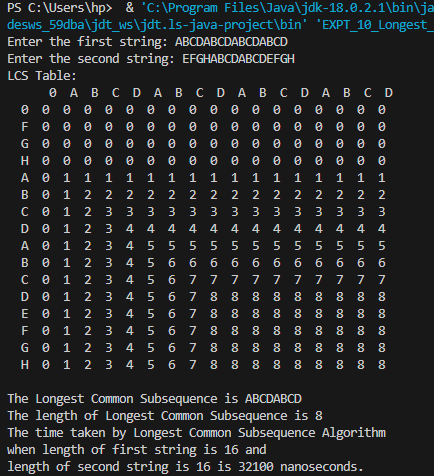
Case 1: When the length of longest string is 8

Time taken is 21,000 nanoseconds.



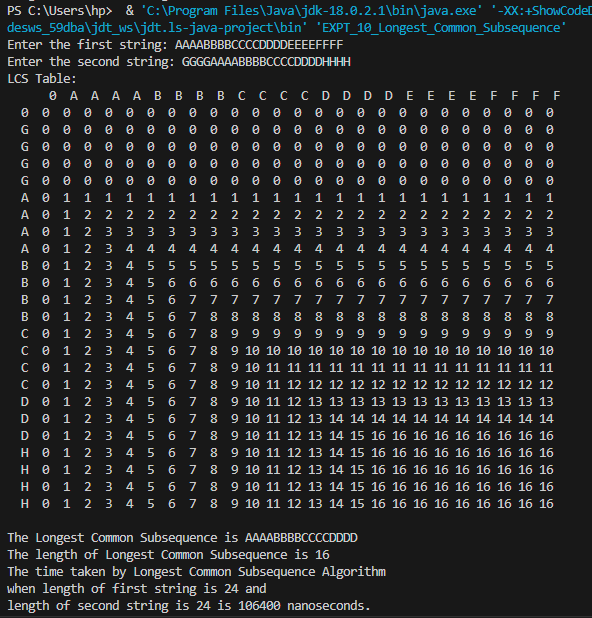
Case 2: When the length of longest string is 16

Time taken is 32,100 nanoseconds.

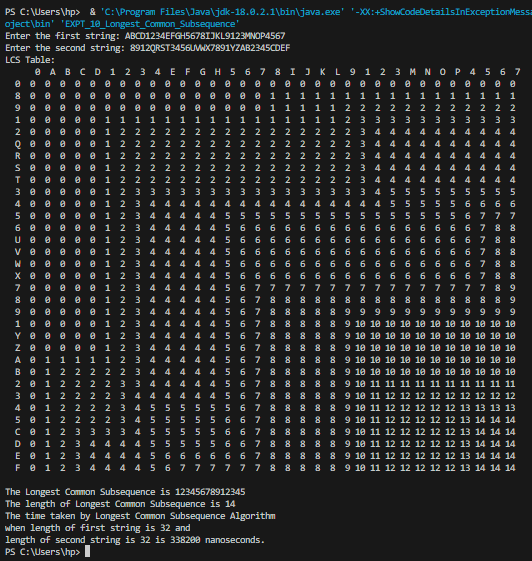


Case 3: When the length of longest string is 24.

Time taken is 106,400 nanoseconds.



Case 4: When the length of longest string is 32.



The time taken is 338,200 nanoseconds.