Longest Common Subsequence

Link:- https://leetcode.com/problems/longest-common-subsequence/description/

Given two strings text1 and text2, return *the length of their longest* ***common subsequence****.* If there is no **common subsequence**, return 0.

A **subsequence** of a string is a new string generated from the original string with some characters (can be none) deleted without changing the relative order of the remaining characters.

* For example, "ace" is a subsequence of "abcde".

A **common subsequence** of two strings is a subsequence that is common to both strings.

**Example 1:**

**Input:** text1 = "abcde", text2 = "ace"

**Output:** 3

**Explanation:** The longest common subsequence is "ace" and its length is 3.

class Solution {

    public int longestCommonSubsequence(String text1, String text2) {

int x=text1.length();

int y=text2.length();

        int ar[][]=new int[x+1][y+1];

        for(int i=1;i<x+1;i++){

            for(int j=1;j<y+1;j++){

                if(text1.charAt(i-1)==text2.charAt(j-1)){

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

                }

                else{

                    ar[i][j]=(int)Math.max(ar[i-1][j],ar[i][j-1]);

                }

            }

        }

        return ar[x][y];

    }

}