The Longest Common Subsequence

by PRASHANTB1984

Problem Submissions Leaderboard Discussions

A subsequence is a sequence that can be derived from another sequence by deleting some elements without changing the order of the remaining elements. Longest common subsequence (*LCS*) of 2 sequences is a subsequence, with maximal length, which is common to both the sequences.

Given two sequence of integers, $A = [a_1, a_2, \ldots, a_n]$ and $B = [b_1, b_2, \ldots, b_m]$, find **any one** longest common subsequence.

In case multiple solutions exist, print any of them. It is guaranteed that at least one non-empty common subsequence will exist.

Recommended References

This Youtube video tutorial explains the problem and its solution quite well.

Lec 15 | MIT 6.046J / 18.410J Introduction to Algorithms (SMA 5...

Input Format

First line contains two space separated integers, n and m, where n is the size of sequence A, while m is size of sequence B. In next line there are n space separated integers representing sequence A, and in third line there are m space separated integers representing sequence B.

```
n m
A1 A2 … An
B1 B2 … Bm
```

Constraints

$1 \leq n \leq 100$
$1 \leq m \leq 100$
$0 \leq a_i < 1000, where\ i \in [1, n]$
$0 \leq b_j < 1000, where\ j \in [1, m]$

Output Format

Print the longest common subsequence and each element should be separated by at least one white-space. In case of multiple answers, print any one of them.

Sample Input

```
5 6
1 2 3 4 1
3 4 1 2 1 3
```

Sample Output

```
1 2 3
```

Explanation

There is no common subsequence with length larger than 3. And "1 2 3", "1 2 1", "3 4 1" are all correct answers.

Tested by Khongor

Solved score: 55.00pts

Submissions: 8167
Max Score: 55
Difficulty: Medium

Rate This Challenge:

More

Current Buffer (saved locally, editable) Java 8

```
import java.io.*;
import java.util.*;

public class Solution {

    public static void main(String[] args) throws IOException {

        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

        String[] numbers = br.readLine().split("\\s");

        int[] n = new int[Integer.parseInt(numbers[0])];
        int[] m = new int[Integer.parseInt(numbers[1])];

        numbers = br.readLine().split("\\s");

        for(int i = 0 ; i < n.length ; i++){
            n[i] = Integer.parseInt(numbers[i]);
        }

        numbers = br.readLine().split("\\s");

        for(int i = 0 ; i < m.length ; i++){
            m[i] = Integer.parseInt(numbers[i]);
        }

        int arr[][] = new int[n.length][m.length];

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

            for(int j = 0 ; j < m.length ; j++){

                if(i == 0 && j == 0){
                    arr[i][j] = n[i] == m[j] ? 1 : 0;
```

```
                }
                else if(i == 0 && j > 0){
                    arr[i][j] = n[i] == m[j] ? 1 : arr[i][j - 1];
                }
                else if(i > 0 && j == 0){
                    arr[i][j] = n[i] == m[j] ? 1 : arr[i - 1][j];
                }
                else{

                    arr[i][j] = n[i] == m[j] ? arr[i - 1][j - 1] + 1 : Math.max(arr[i - 1][j], Math.max(arr[i - 1][j - 1],arr[i][j - 1]));

                }
            }
        }

        int row = n.length - 1;
        int col = m.length - 1;

        ArrayList<Integer> list = new ArrayList<Integer>();

        while(true){

            int tmp = arr[row][col];

            if(tmp == 0){
                break;
            }


            if(row > 0 && tmp == arr[row - 1][col]){
                row = row - 1;
            }
            else if(col > 0 && tmp == arr[row][col - 1]){
                col = col - 1;
            }
            else{
                list.add(n[row]);
                row = row - 1;
                col = col - 1;
            }

            if(row < 0 || col < 0){
                break;
            }
        }

        for(int i = list.size() - 1 ; i > - 1 ; i--){
            System.out.print(list.get(i) + " ");
        }

    }
}
```

Line: 1 Col: 1

Upload Code as File Test against custom input Run Code Submit Code