

Algorithmics	Student information	Date	Number of session
	UO: 275725		5
	Surname: Gómez Menéndez		
	Name: Laura		

Activity 1. [Validation results]

I was not able to fill the table.

GCCCTAGCG and GCGCAATG

```
<terminated> LCSTest [Java Application] C:\Users\usuario\Documents\Plantillas personalizadas de Office\bin\javaw.exe (7 abr. 2021 21:11:25)
DYNAMIC PROGRAMMING:
String1: *GCCCTAGCG
String2: *GCGCAATG
Initializing table...
Filling table...
Print table...
      *      *      G      C      C      C      T      A      G      C      G
* 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0)
G 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0)
C 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0)
G 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0)
C 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0)
A 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0)
A 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0)
T 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0)
G 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0)
Finding longest subsequence...
*GCGCG
Printing longest subsequence...

/*****/

RECURSIVE:
Finding longest subsequence...
GCCTG
Program terminated.
```

GCATGCAT and GAATTCAG

```
<terminated> LCSTest [Java Application] C:\Users\usuario\Documents\Plantillas personalizadas de Office\bin\javaw.exe (7 abr. 2021 21:12:17)
DYNAMIC PROGRAMMING:
String1: *GCATGCAT
String2: *GAATTCAG
Initializing table...
Filling table...
Print table...
      *      *      G      C      A      T      G      C      A      T
* 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0)
G 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0)
A 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0)
A 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0)
T 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0)
T 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0)
C 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0)
A 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0)
G 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0) 0( 0, 0)
Finding longest subsequence...
*GATCA
Printing longest subsequence...

/*****/

RECURSIVE:
Finding longest subsequence...
GATCA
Program terminated.
```

Algorithmics	Student information	Date	Number of session
	UO: 275725		5
	Surname: Gómez Menéndez		
	Name: Laura		

Activity 2. [Experimental time measurements]

n	T_dynamic
100	10.2
200	16.6
400	24.4
800	37.9
1600	67.9
3200	801.9

n	T_recursive
8	1.6
10	5
12	19.2
14	120.7
16	487.4
18	2610.6

Algorithmics	Student information	Date	Number of session
	UO: 275725		5
	Surname: Gómez Menéndez		
	Name: Laura		

Activity 3. [Answer to questions in Section 3 (A, B, C and D)]

A) Determine theoretically complexities (time, memory space and waste of stack) for both implementations, recursive (approximated) and using programming dynamic.

Dynamic programming

Time complexity: $O(n * m)$ (n and m are the lengths of the strings)

Stack memory:

Recursive

Time complexity: $O(2^n)$

Stack memory:

B) Compute theoretical times and compare them with the experimental measurements.

As I said, the recursive one has a complexity of $O(2^n)$, to check it we can calculate its execution time.

$T_2 = C^{(n-1)} * t_1 = 2^{120} * 120.7 = 482.8$ that is very similar to 487.4.

C) Why large sequences cannot be processed with the recursive implementation? Explain why dynamic programming implementation raises an exception for large sequences.

This happens because if it calls itself many times before returning, the space needed may exceed the Java limits, so with a large number it returns a `StackOverflowError`.

Algorithmics	Student information	Date	Number of session
	UO: 275725		5
	Surname: Gómez Menéndez		
	Name: Laura		

D) The amount of possible LCS can be more than one, e. g. GCCCTAGCG and GCGCAATG has two GCGCG and GCCAG. Find the code section that determines which subsequence is chosen, modify this code to verify that both solutions can be achieved.

With the code that I uploaded I get GCGCG, and with this one:

```
int[][] table = new int[size2 + 1][size1 + 1];

    for (int i = 0; i <= size1; i++) {
        for (int j = 0; j <= size2; j++) {
            if (i == 0 || j == 0)
                table[j][i] = 0;
            else if (str2.charAt(j - 1) == str1.charAt(i - 1))
                table[j][i] = table[i - 1][j - 1] + 1;
            else
                table[j][i] = Math.max(table[j - 1][i], table[j][i - 1]);
        }
    }

    int index = table[size2][size1];
    int aux = index;

    char[] lcs = new char[index + 1];
    lcs[index] = '\0';

    int i = size2;
    int j = size1;

    while (i > 0 && j > 0) {
        if (str2.charAt(i - 1) == str1.charAt(j - 1)) {
            lcs[index - 1] = str2.charAt(i - 1);
            i--;
            j--;
            index--;
        }
        else if (table[j - 1][i] > table[j][i - 1])
            j--;
        else
            i--;
    }

    for (int k = 0; k <= aux; k++)
        System.out.print(lcs[k]);
    System.out.println("");
```

I get GCAG but not GCCAG.