Algorithmics	Student information	Date	Number of session
	UO:275688	02/03/21	5
	Surname: Fernández Esparta		
	Name: Mikel		

Escuela de Ingeniería Informática
Universidad de Oviedo

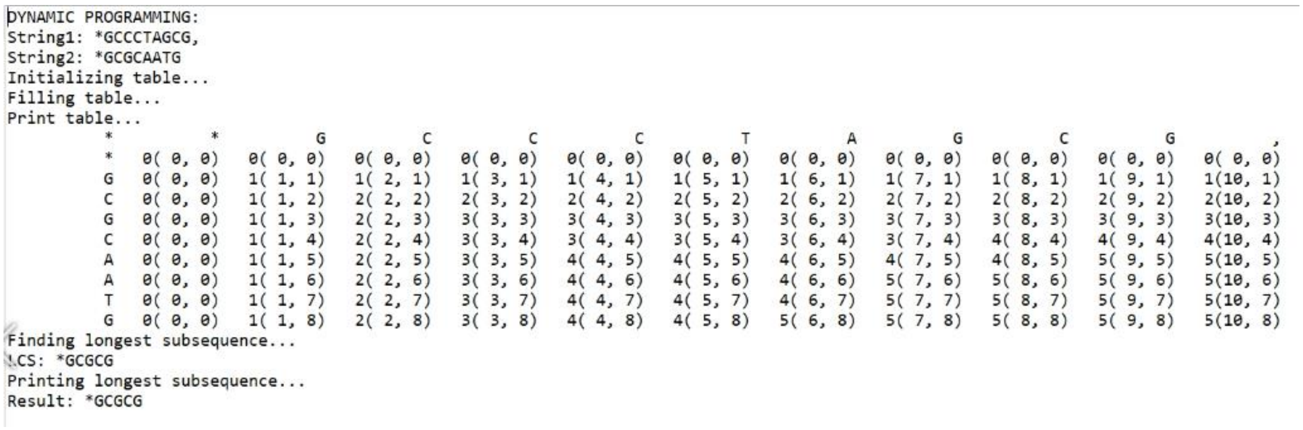
Universidad de Oviedo
Universidá d'Uviéu
University of Oviedo

Activity 1. [Validation results]

After implementing the methods for the following classes, LCSRec and LCS, we must show the output given an example in this case, in my case, I have decided to use this one: GCCCTAGCG and GCGCAATG.

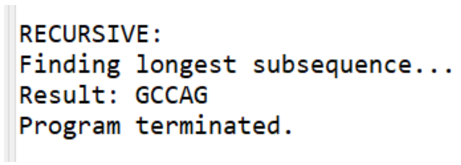
This is the ouput shown in the console after running the LCSTest class with the previous parameters. In which you can see both inputs, the table that is used for dynamic programming and the result which is the longest subsequent possible.

Here we are using dynamic programming:



```
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)   1( 1, 1)   1( 2, 1)   1( 3, 1)   1( 4, 1)   1( 5, 1)   1( 6, 1)   1( 7, 1)   1( 8, 1)   1( 9, 1)   1(10, 1)
         C   0( 0, 0)   1( 1, 2)   2( 2, 2)   2( 3, 2)   2( 4, 2)   2( 5, 2)   2( 6, 2)   2( 7, 2)   2( 8, 2)   2( 9, 2)   2(10, 2)
         G   0( 0, 0)   1( 1, 3)   2( 2, 3)   3( 3, 3)   3( 4, 3)   3( 5, 3)   3( 6, 3)   3( 7, 3)   3( 8, 3)   3( 9, 3)   3(10, 3)
         C   0( 0, 0)   1( 1, 4)   2( 2, 4)   3( 3, 4)   3( 4, 4)   3( 5, 4)   3( 6, 4)   3( 7, 4)   4( 8, 4)   4( 9, 4)   4(10, 4)
         A   0( 0, 0)   1( 1, 5)   2( 2, 5)   3( 3, 5)   4( 4, 5)   4( 5, 5)   4( 6, 5)   4( 7, 5)   4( 8, 5)   5( 9, 5)   5(10, 5)
         A   0( 0, 0)   1( 1, 6)   2( 2, 6)   3( 3, 6)   4( 4, 6)   4( 5, 6)   4( 6, 6)   5( 7, 6)   5( 8, 6)   5( 9, 6)   5(10, 6)
         T   0( 0, 0)   1( 1, 7)   2( 2, 7)   3( 3, 7)   4( 4, 7)   4( 5, 7)   4( 6, 7)   5( 7, 7)   5( 8, 7)   5( 9, 7)   5(10, 7)
         G   0( 0, 0)   1( 1, 8)   2( 2, 8)   3( 3, 8)   4( 4, 8)   4( 5, 8)   5( 6, 8)   5( 7, 8)   5( 8, 8)   5( 9, 8)   5(10, 8)
Finding longest subsequence...
LCS: *GCGCG
Printing longest subsequence...
Result: *GCGCG
```

In this case, we are using the recursive implementation.



```
RECURSIVE:
Finding longest subsequence...
Result: GCCAG
Program terminated.
```

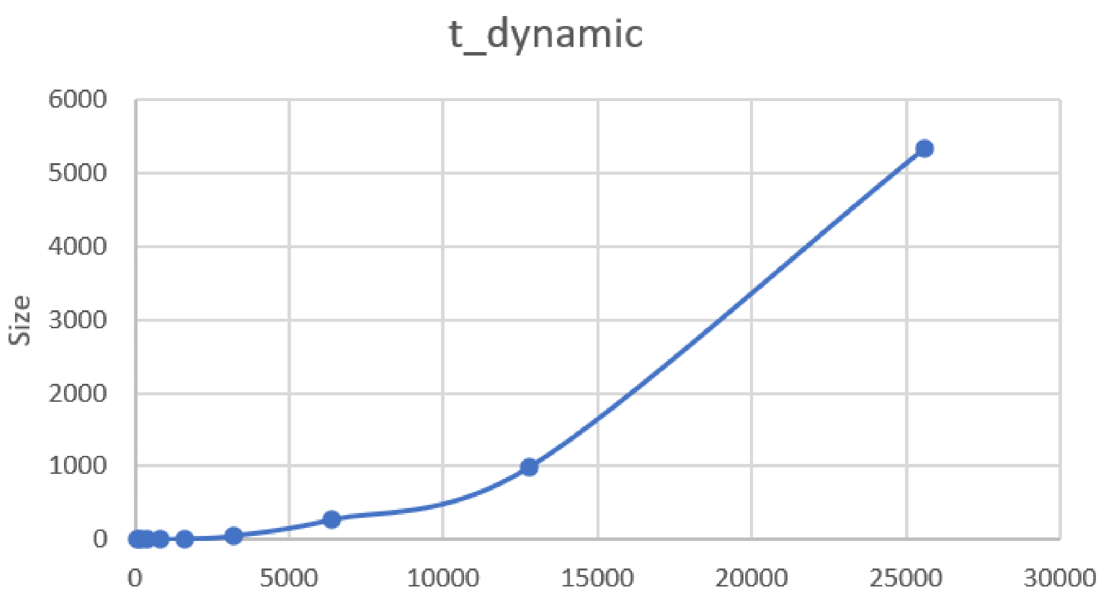
Activity 2. [Experimental time measurements]

These are the measured times that show the time taken to execute each of both algorithms for different size problems.

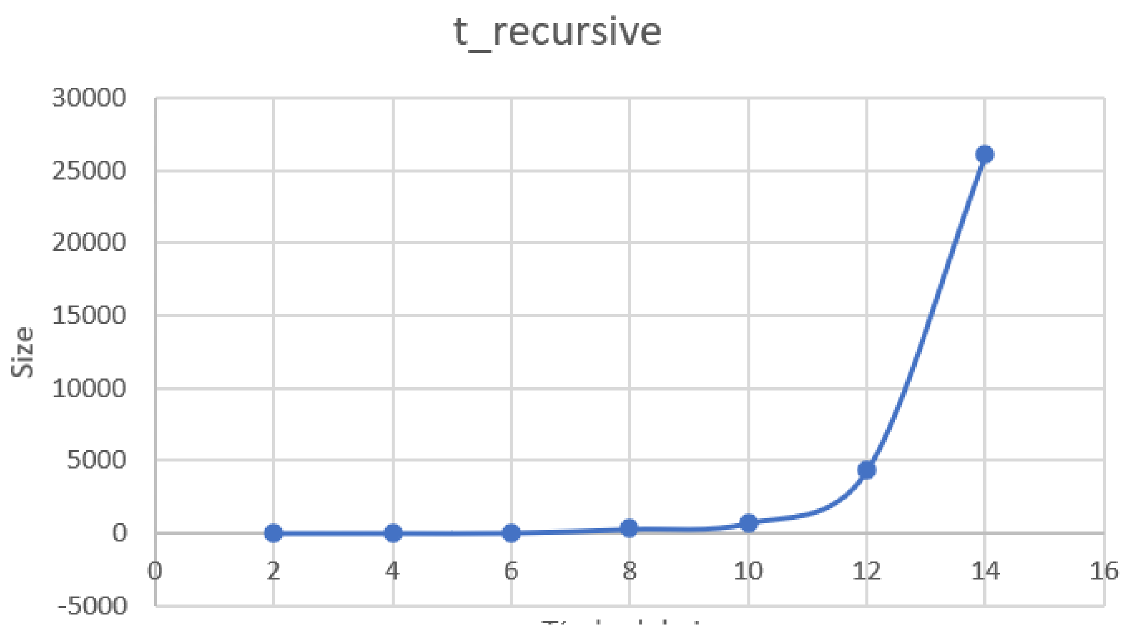
Algorithmics	Student information	Date	Number of session
	UO:275688	02/03/21	5
	Surname: Fernández Esparta		
	Name: Mikel		

n	t_dynamic
100	0,1
200	0,6
400	1,3
800	3,4
1600	14
3200	55,5
6400	278,6
12800	996,8
25600	5336,7

n	t_recursive
2	0,2
4	2,4
6	14,9
8	282,4
10	688,3
12	4304,52
14	26086,35



Algorithmics	Student information	Date	Number of session
	UO:275688	02/03/21	5
	Surname: Fernández Esparta		
	Name: Mikel		



Activity 3. [Questions]

A) Determine theoretically complexities (time, memory space and waste of stack) for both implementations, recursive (approximated) and using programming dynamic.
The time complexity for the recursive algorithm is O(2^n), while the time complexity for dynamic programming is O(size1 * size2) being size1 = str1.length and size2 = str2.length, which is much better than the worst-case scenario of a recursive implementation for this kind of problems.

B) Compute theoretical times and compare them with the experimental measurements.
In dynamic programming we may have size1 = 200 with t1 = 0,6 and size2 = 400 then, t2 would equal 1,2 a bit lower when comparing the value with our experimental measurements which is 1,3.
O the recursive implementation for n1= 2 t1 = 0,2 and n2 = 4 then t2 would equal 0,8 which is much lower than our expected value.

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

Algorithmics	Student information	Date	Number of session
	UO:275688	02/03/21	5
	Surname: Fernández Esparta		
	Name: Mikel		

It cannot be processed with the recursive implementation due to its high time complexity, O(n^2), this is the worst-case and for that it cannot process large sequences.
It probably will raise exceptions for large sequences because of the size of that sequence being really large.