

## Práctica 5

### Mediciones realizadas en:

Intel(R) Core(TM) i78550U  
RAM 16,0 GB

**A) Determina las complejidades (tiempo, memoria y uso de la pila) para ambas implementaciones, recursiva (de forma aproximada) y usando programación dinámica.**

Complejidad DV  $\rightarrow O(2^n)$  (Exponencial)

Complejidad programación dinámica  $\rightarrow O(n^2)$  (Polinómica)

**B) Calcula los tiempos teóricos y compáralos con las medidas experimentales.**

Hecho más adelante, en la sección TIEMPOS.

**C) ¿Por qué no se pueden procesar cadenas largas con la implementación recursiva? Explica por qué la implementación con programación dinámica genera una excepción para secuencias largas.**

El problema de divide y vencerás es que falla al descomponer el problema en un número alto de subproblemas, lo que sucede en cadenas largas. A ello se le puede sumar el caso de que se generen subproblemas repetidos.

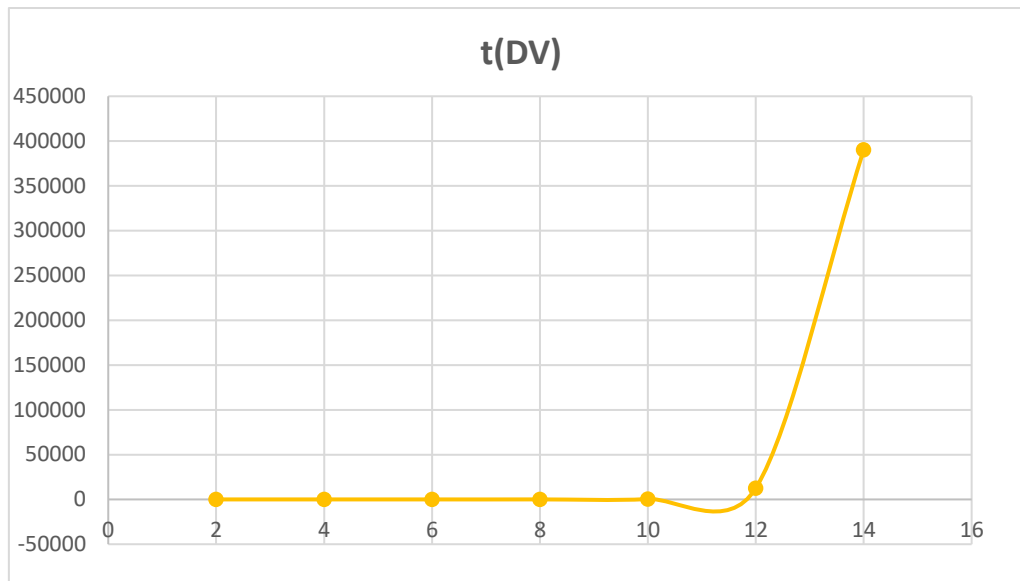
En la programación dinámica se genera una excepción stackoverflow porque llena la pila por acumulación.

**D) Puede haber más de una MSC, p. e. GCCCTAGCG y GCGCAATG tiene dos GCGCG y GCCAG. Encuentra la sección de código que determina qué subsecuencia es elegida, modifícalo y verifica que se pueden encontrar ambas soluciones.**

```
if(str1.charAt(i - 1) == str2.charAt(j - 1)) { // CASO ESPECIAL DE L3
    table[i][j].value++;
}
```

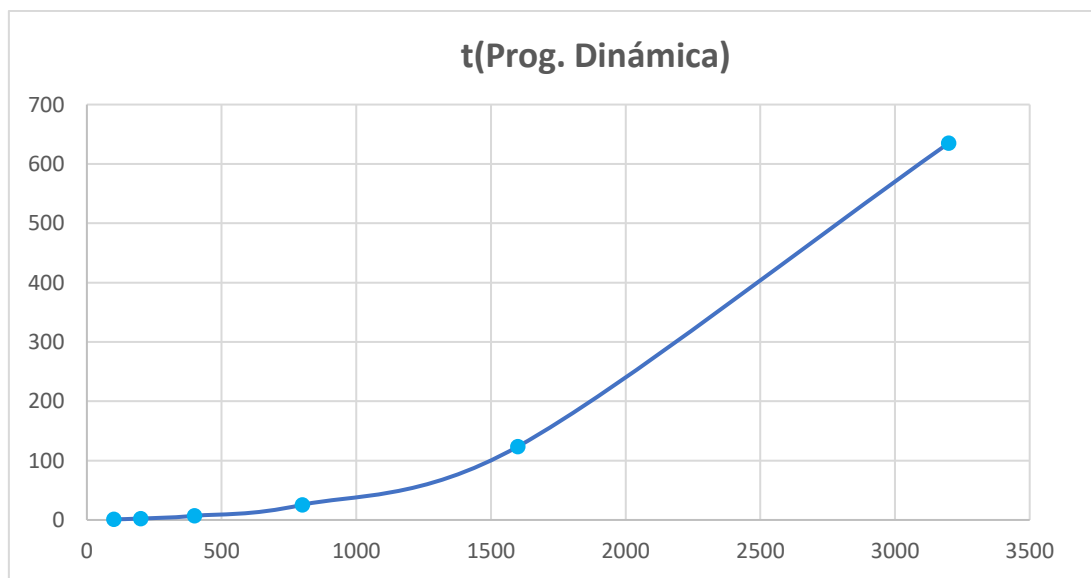
### TIEMPOS

n	t(DV)	nVeces (DV)
2	0,002	nVeces = 100000
4	0,01662	
6	0,43483	
8	14,4318	nVeces = 10000
10	426,66	nVeces = 100
12	12597,9	nVeces = 10
14	390211	nVeces = 1
Complejidad:	$O(2^n)$	



Sí se corresponde con una exponencial, por su rápido crecimiento.

n	t(Prog. Dinámica)	nVeces (Prog. Dinámica)
100	1,036	nVeces = 1000
200	2,386	
400	6,988	
800	25,534	
1600	123,775	
3200	635,217	
Complejidad:		$O(n^2)$



Teóricamente la complejidad de la programación dinámica es polinómica. Sí se corresponde en las medidas del ejercicio.