

Procesador:	11th Gen Intel(R) Core(TM) i7-1165G7 @ 2.8GHz
Memoria:	16 GB

Cuestiones:

Complejidad del algoritmo usando programación dinámica:

- La función rellenaTabla posee una complejidad cuadrática, $O(n^2)$.
- Si la condición es verdadera, la función encontrarMSC posee complejidad $O(n)$.
- La función máximo posee complejidad constante.

Complejidad del algoritmo usando recursividad:

- La función encontrarMSC posee complejidad exponencial, $O(3^n)$, pues $\{a=3, b=1, k=0\}$, por substracción.
- La función máxima posee complejidad constante.

La complejidad del algoritmo con implementación recursiva es exponencial, cuanto más largas sean las cadenas más repeticiones innecesarias se realizarán en el algoritmo, y como consecuencia, más tiempo tardará en hallar la solución al problema.

Con programación dinámica se acelera este proceso, mejora la eficiencia del algoritmo. Evita calcular dos veces (o más) lo mismo.

Mi algoritmo recursivo no está completado.

Tablas de tiempos:

nTimes = 1000000000

n	progdin
	milisegundos
100	4.0E-9
200	2.71E-7
400	2.31E-7
800	2.33E-7
1600	2.66E-7
3200	2.36E-7
6400	2.47E-7
Java heap space	

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.

Según la implementación de la función encontrarMSC.

Muestro los dos ejemplos del enunciado de la práctica. El ejemplo pedido 'GCGCG' no conseguí encontrarlo.

```
<terminated> MscPrueba [Java Application] C:\Users\mayte\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_15.0.1.v20201011-1920\jre\bin\java.exe
PROGRAMACIÓN DINÁMICA:
String1: *GCCCTAGCG
String2: *GCGCAATG
Iniciando la tabla...
Rellenando la tabla...
Imprimiendo la tabla...
      *      *      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( 0, 0) 1( 1, 1) 1( 2, 1) 1( 3, 1) 1( 4, 1) 1( 5, 1) 1( 6, 0) 1( 7, 1) 1( 8, 0) 1( 8, 0)
C 0( 0, 0) 1( 1, 1) 2( 1, 1) 2( 2, 1) 2( 3, 1) 2( 4, 2) 2( 5, 2) 2( 6, 2) 2( 7, 1) 2( 8, 2) 2( 8, 2)
G 0( 0, 0) 1( 1, 2) 2( 2, 2) 2( 3, 2) 2( 4, 2) 2( 5, 2) 2( 6, 2) 3( 6, 2) 3( 7, 3) 3( 8, 2) 3( 8, 2)
C 0( 0, 0) 1( 1, 3) 2( 2, 3) 3( 2, 3) 3( 3, 3) 3( 4, 4) 3( 5, 4) 3( 7, 3) 4( 7, 3) 4( 8, 4) 4( 8, 4)
A 0( 0, 0) 1( 1, 4) 2( 2, 4) 3( 3, 4) 3( 4, 4) 3( 5, 4) 4( 5, 4) 4( 6, 5) 4( 8, 4) 4( 9, 4) 4( 9, 4)
T 0( 0, 0) 1( 1, 5) 2( 2, 5) 3( 3, 5) 3( 4, 5) 3( 5, 5) 4( 6, 5) 4( 7, 5) 4( 8, 5) 4( 9, 5) 4( 9, 5)
G 0( 0, 0) 1( 1, 6) 2( 2, 6) 3( 3, 6) 3( 4, 6) 4( 4, 6) 4( 6, 6) 4( 7, 6) 4( 8, 6) 4( 9, 6) 4( 9, 6)
Buscando la MSC...
      5( 8, 7)->G
      4( 6, 5)->A
      3( 3, 3)->C
      2( 1, 1)->C
      1( 0, 0)->G
Imprimiendo la MSC...
GCCAG
```

```
<terminated> MscPrueba [Java Application] C:\Users\mayte\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_15.0.1.v20201011-1920\jre\bin\java.exe
PROGRAMACIÓN DINÁMICA:
String1: *GCATGCAT
String2: *GAATTCAG
Iniciando la tabla...
Rellenando la tabla...
Imprimiendo la tabla...
      *      *      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) 1( 0, 0) 1( 1, 1) 1( 2, 1) 1( 3, 1) 1( 4, 1) 1( 5, 1) 1( 6, 1) 1( 7, 1) 1( 7, 1)
A 0( 0, 0) 1( 1, 1) 1( 2, 1) 2( 2, 1) 2( 3, 2) 2( 4, 2) 2( 5, 2) 2( 6, 1) 2( 7, 2) 2( 7, 2)
A 0( 0, 0) 1( 1, 2) 1( 2, 2) 2( 3, 2) 2( 4, 2) 2( 5, 2) 2( 6, 2) 3( 6, 2) 3( 7, 3) 3( 7, 3)
T 0( 0, 0) 1( 1, 3) 1( 2, 3) 2( 3, 3) 3( 3, 3) 3( 4, 4) 3( 5, 4) 3( 7, 3) 4( 7, 3) 4( 7, 3)
T 0( 0, 0) 1( 1, 4) 1( 2, 4) 2( 3, 4) 3( 4, 4) 3( 5, 4) 3( 6, 4) 3( 7, 4) 4( 8, 4) 4( 8, 4)
C 0( 0, 0) 1( 1, 5) 2( 2, 5) 2( 3, 5) 3( 4, 5) 3( 5, 5) 4( 5, 5) 4( 6, 6) 4( 8, 5) 4( 8, 5)
A 0( 0, 0) 1( 1, 6) 2( 2, 6) 3( 2, 6) 3( 4, 6) 3( 5, 6) 4( 6, 6) 5( 6, 6) 5( 7, 7) 5( 7, 7)
G 0( 0, 0) 1( 1, 7) 2( 2, 7) 3( 3, 7) 3( 4, 7) 4( 4, 7) 4( 6, 7) 5( 7, 7) 5( 8, 7) 5( 8, 7)
Buscando la MSC...
      5( 6, 6)->A
      4( 5, 5)->C
      3( 3, 3)->T
      2( 3, 2)->A
      1( 0, 0)->G
Imprimiendo la MSC...
GATCA
```