

# PRÁCTICA 7

Ángela López López UO270318

## ACTIVIDAD 1 . CASOS DE PRUEBA

**PC CASA:** Intel ® Core(TM) i5-8250U CPU @ 1.60Ghz 1.80 GHz, MEMORIA: 8.00 GB

Debido a la falta de tiempo durante estas semanas (ya que perdí demasiado intentando comprender cómo debía funcionar el juego) solo he podido implementar los dos primeros algoritmos.

Sin embargo, voy a hacer un análisis teórico.

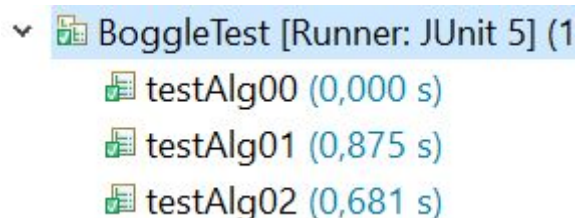
Al ejecutar el juego por backtracking, se usa un recorrido en profundidad (por hijos). Para ello he utilizado una estructura de árbol llamada Trie, en la que cada nodo es un carácter, de forma que se puede utilizar el prefijo a la hora de comprar y realizar el recorrido.

Esto está bien para encontrar todas las soluciones posibles, pero no me parece la solución más acertada para buscar la palabra que mejor puntúa.

Al ejecutar el juego por ramificación y poda, se usa un recorrido en anchura (por niveles). Para ello se utiliza una cola que permita almacenar los niveles. Ambos llegan a la solución ya que el tablero de palabras es un grafo. Sin embargo la implementación de DFS es más sencilla.

Por contra, la solución que más puntúa suele ser aquella que más se repite. Eso ocurre (generalmente) con las palabras más cortas, ya que se puede dar el caso de que sea posible concatenarla con distintas celdas.

En ese caso no tenemos que alejarnos demasiado de la raíz, y la búsqueda por anchura podría resultar más eficiente que la búsqueda por profundidad.



## ACTIVIDAD 2. TIEMPO PARA LAS DIFERENTES EJECUCIONES

JUEGO TODAS LAS SOLUCIONES:

Tamaño: 1 - Tiempo: 1 ms - Puntos: 0

Tamaño: 2 - Tiempo: 6 ms - Puntos: 168

Tamaño: 3 - Tiempo: 50 ms - Puntos: 482

Tamaño: 4 - Tiempo: 5418 ms - Puntos: 14072  
Tamaño: 5 - Tiempo: 9822 ms - Puntos: 10771  
Tamaño: 6 - Tiempo: 14940 ms - Puntos: 6982  
Tamaño: 7 - Tiempo: 182171 ms - Puntos: 20721

#### JUEGO MEJOR SOLUCIÓN:

Tamaño: 1 - Tiempo: 0 ms - Puntos: 0  
Tamaño: 2 - Tiempo: 9 ms - Puntos: 450  
Tamaño: 3 - Tiempo: 47 ms - Puntos: 1382  
Tamaño: 4 - Tiempo: 923 ms - Puntos: 3854  
Tamaño: 5 - Tiempo: 3871 ms - Puntos: 8559  
Tamaño: 6 - Tiempo: 9214 ms - Puntos: 8410  
Tamaño: 7 - Tiempo: 184140 ms - Puntos: 31676

#### **¿Qué piensas de la complejidad del algoritmo?**

En la práctica anterior, la complejidad era  $O(8^n)$  porque solo tenía 8 vecinos directos y la posición inicial podría ser cualquiera del tablero ( $n^2$ ).

Sin embargo, ahora cada nodo tiene tantos vecinos posibles como repeticiones se permitan. Esto aumenta la complejidad a niveles exponenciales que hacen que este problema sea intratable con esta solución.