

# Super Sopa

GRAU A      CURS 2015-2016\*

Departament de Ciències de la Computació  
alg@cs.upc.edu

## Resumen

*Esta práctica tiene como objetivo realizar una validación experimental de la eficiencia y aplicabilidad de diferentes algoritmos para jugar un juego basado en una sopa de letras al que denominaremos SuperSopa.*

*La práctica se realizará en grupos de 2 o 3 personas (excepcionalmente 1, bajo autorización expresa). La **composición de los grupos** se tiene que comunicar a alg@cs.upc.edu antes del **16 de mayo de 2016**.*

*La entrega de la práctica se realizará a través del Racò. La entrega estará abierta hasta las 23:59 horas del día **17 de junio de 2016**. Algunos grupos pueden ser convocados para una entrevista personal el 21 o el 22 de junio con prueba interactiva. Algunos grupos pueden recibir preguntas por e-mail a fin de aclarar dudas relativas a su práctica. **Es obligatorio que en la entrevista estén presentes todos los miembros del grupo.***

## I. EL JUEGO

Se dispone de una colección de números a los que denominaremos *diccionario válido*. Los números están representados mediante una cadena de caracteres (0 a 9) de cualquier longitud. Siguiendo la nomenclatura habitual llamaremos *palabras* a los componentes del diccionario.

Un *tablero* de SuperSopa es un tablero con  $N \times N$  casillas. En cada casilla se almacena un número entre 0 y 9. Dos casillas en el tablero son *adyacentes* si son contiguas horizontal, vertical o diagonalmente.

El objetivo del juego es encontrar el máximo número posible de palabras del diccionario en el tablero.

Una palabra *está en el tablero* si se puede ubicar la palabra en el tablero de manera que letras consecutivas de la palabra aparezcan en celdas adyacentes del tablero.

El objetivo del juego es encontrar palabras del diccionario que están en el tablero y obtener la máxima puntuación dentro de un límite de tiempo. Cada palabra encontrada proporciona tantos puntos como longitud tenga. Si una palabra del diccionario está más de una vez en el tablero sólo puntúa una vez.

## II. OBJETIVOS

El objetivo de esta práctica es analizar el coste y la efectividad de diferentes algoritmos para resolver un tablero de SuperSopa. Para ello os pedimos que diseñéis algoritmos basados en dos criterios de búsqueda:

- Explorando las posibilidad de que un patrón de posiciones contiguas en el tablero contenga una palabra del diccionario.
- Examinando la posibilidad de que una palabra del diccionario esté o no en el tablero.

---

\*La versión más actualizada de este documento, así como cualquier material adicional relacionado se publicará al Racò.

El objetivo final es ver experimentalmente si hay diferencia entre los tiempos de ejecución y la calidad de los resultados en los diferentes algoritmos de búsqueda de palabras que propongais.

Este documento es intencionadamente vago. Por tanto, además de analizar diferentes algoritmos deberá documentar el diseño de experimentos para contrastar y validar sus hipótesis y conclusiones.

### III. PROGRAMAS

Implementar al menos un programa en C++ para cada variante de búsqueda. En la versión más sencilla (suficiente para aprobar el proyecto si se combina con una buena experimentación) se puede diseñar algoritmos que utilicen algunos de los algoritmos de hash más básicos vistos en clase.

Versiones más sofisticadas de la práctica (el nivel de sofisticación y esfuerzo dedicado es opcional y se tendrá en cuenta a la hora de evaluarla) incluirán la implementación de otros algoritmos con coste teórico mejor que los básicos.

Tened en cuenta que deberéis medir el tiempo de los algoritmos. Además, tendréis que hacer un seguimiento de varios contadores que reflejen la cantidad de trabajo que el programa hace. Por ejemplo, el número total de comparaciones necesarias para ordenar los datos del diccionario (si es necesario), número total de comparaciones para todas las búsquedas exitosas / fallidas, etc.

También puede ser interesante medir el coste medio (en comparaciones y llamadas a funciones u otros) necesarios para crear y buscar en la tabla de hash (si el programa la utiliza). Considerad otras medias útiles y documentadlas.

### IV. DATOS

La idea general es que primero se cree un archivo con  $K$  números (cadens de caracteres de entre 0..9) seleccionados al azar. Después, ejecutar los algoritmos con diferentes tableros generados aleatoriamente.

Asegurados de que en cada uno de los ejemplos  $K$  y/o  $N$  son lo suficientemente grandes como para que se puedan obtener buenos resultados experimentales. La comparación de dos programas que se ejecutan en 0,05 segundos y 0,07 segundos y concluyendo que el primero es más rápido no es buena técnica experimental - nada habría podido contribuir a la diferencia! Esto no significa necesariamente que tenga que ser lo más grande posible. valores moderadamente grandes con múltiples ensayos en más de un conjunto de datos puede ser revelador.

Asegurados de mantener dimensiones pequeñas, mientras está probando los programas. Para garantizar la reproductibilidad de los experimentos tenéis que entregar también los algoritmos de generación de los archivos de datos que utiliza.

### V. QUÉ HAY QUE ENTREGAR?

Hay que entregar una carpeta que contenga:

- Una breve descripción de lo que se ha implementado, las pruebas que se ha realizado (y por qué se han diseñado así) y la comparación de los resultados que se han obtenido. También es interesante indicar ideas que se ha probado aunque no hayan dado buenos resultados. El documento en formato pdf debe incluir las referencias adecuadas.
- Subcarpetas con las fuentes necesarias para compilar y ejecutar la práctica. Se incluirán las instrucciones para la compilación y ejecución, tanto de los algoritmos para el problema como para la generación de los ficheros de datos.