**Resolución de Sudoku con Metodologoía de Búsqueda con Retroceso (Backtracking)**

**Autores:** Pablo Lario & Diego Miguel López

**Fecha:** 19/05/2022

**Asignatura:** Diseño De Algoritmos

**Representación del problema**

Como se menciona en el enunciado de la práctica, el problema se va a leer a partir de un fichero .txt. Esto lo podemos hacer con la función open() de python. Para facilitar la resolución del problema, hemos decidido que el problema va a estar representado en una matriz 9x9, en el que los 0s indican un elemento vacío que está por resolver, y los números del 1 al 9 indican el número que hay en esa fila y columna.

Para poder tener el problema en una matriz 9x9, necesitamos leer cada linea del archivo .txt, y para cada linea leer cada carácter. Si este carácter es un espacio, entonces guardamos en la matriz un 0, y si es un número, guardamos ese número en la matriz. Asumimos que los valores en el archivo que leemos están bien y no contienen letras ni símbolos, solo números del 0 al 9.

**Imprimir el tablero**

Una vez tenemos el problema guardado en una matriz 9x9, hemos decidido crear una funcion específica que imprimar el tablero pasado por parámetro por pantalla. Esta función simplemente contiene dos bucles, uno para las filas y otro para las columnas. Además, la función imprime lineas para separar claramente cada caja 3x3 del sudoku, ya que si recordamos como funciona el sudoku tenemos una matriz de 9x9, en la que dos elementos no pueden ser iguales en tu misma fila, en tu misma columna y en tu misma caja, siendo la caja una matriz 3x3. Así, hay 9 cajas.

**Factiblidad y validez de un número**

Hemos decidido crear una funcion cuyo nombre es valido(tablero, num, pos), que toma 3 parámetros. Esta función se va a encargar de comprobar si el número que queremos meter se puede meter, es decir, si no hay ningún número igual a ese que esté en la misma fila, en la misma columna o en la misma caja. Para esto usamos el parámetro pos, que es una lista con dos valores (x,y). Esta función devolverá False si hay un número igual en la columna, en la fila o en la caja, y True si no hay ningun número igual en ninguna de las tres.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

**Búsqueda con retroceso (Backtracking) y resolución del problema.**

El algoritmo que resuelve el tablero por búsqueda con retroceso no es muy complicado. Primero, se busca un cero en el tablero (es decir, se busca un elemento vacío que completar), y una vez encontrado se intentar encontrar un valor factible que poner en la posición de ese elemento. Si no se encuentra ningún 0 es porque el tablero está completo y ya se ha encontrado una solución, por lo que en este caso terminaría el algoritmo.

Si se encuentra un cero, se van a probar (en un bucle) todos los posibles valores de 1 a 9, y si se encuentra que alguno de estos es válido, se introduce en el tablero y se sigue resolviendo el tablero mediante recursión. Si no se encuentra una solución, el algoritmo vuelve hacia atrás hasta que encuentra un nuevo camino por el que seguir.

Así, se repite este proceso hasta que se llega a una solución o hasta que se han probado todas las posibles combinaciones y no se ha encontrado una solución. En este punto terminaría el algoritmo, habiendo encontrado o no una solución.

Captura de pantalla de computadora

Descripción generada automáticamente