

PRÁCTICA EVALUABLE 2 – UT2 – ALGORITMOS RECURSIVOS

1. Crea un programa en Python con una función recursiva para implementar la búsqueda binaria. Crea dos conjuntos de pruebas para ejecutar la búsqueda binaria e incluye el código necesario para ejecutarlas. **Fichero [binaria_tunombre.py]**
2. Crea un programa en Python con una función recursiva para implementar el método “Quicksort” y poder ordenar un array de forma ascendente. Deberás crear un método para realizar la partición del array y a continuación, implementar la función recursiva Quicksort. Crea dos conjuntos de pruebas para ejecutar la ordenación. **Fichero [quicksort_tunombre.py]**
3. Dada una matriz de 8x8 que representa un tablero para el popular juego “buscaminas”, donde las posiciones con valor -1 representan que hay una mina alrededor y las posiciones con valor ≥ 0 representan el número de minas que hay en los vecinos de cada casilla. **Fichero [buscaminas.py]**
 - a. Realiza una función en Python para generar una matriz de 8x8 con 10 minas ubicadas de forma aleatoria en el tablero:

-1 -> la casilla tiene una mina
 0 -> la casilla NO tiene mina

-1	0	0	-1	0	0	0	
0	0	0	0	-1	0	-1	
0	0	0	0	0	-1	0	
0	-1	0	0	0	-1	0	
0	0	-1	-1	0	-1	0	
0	0	0	0	0	0	0	
0	0	0	0	0	0	0	
0	0	0	0	0	0	0	

- b. Implementa una función para calcular cuantas minas tiene cada casilla a su alrededor:

-1	1	1	-1	2	2	1	
1	1	1	2	-1	3	-1	
1	1	1	1	2	-1	3	
1	-1	3	2	4	-1	3	
1	2	-1	-1	2	-1	2	
0	1	2	2	2	1	1	
0	0	0	0	0	0	0	
0	0	0	0	0	0	0	

- c. Programa una función recursiva llamada “descubre” que reciba tres parámetros:
 - Una matriz de 8x8 casillas con una representación del juego “buscaminas” generada a través de los apartados a y b
 - La coordenada de fila
 - La coordenada de columna

La función deberá modificar el valor de la casilla con el número -2 si la casilla tiene un valor mayor que 0 (no tiene una mina) y, si tiene valor cero (no tiene minas alrededor), deberá descubrir sus casillas adyacentes de forma recursiva.