

EXAMEN RECUPERACIÓN 2º EVALUACIÓN – DAW PRESENCIAL

SE VALORARÁ POSITIVAMENTE:
1. Los algoritmos y las estructuras de datos.
SE PENALIZARÁ:
1. La no justificación de las decisiones. Descuenta 1 punto.
2. Errores en los requisitos y los algoritmos. Un error leve descuenta 1 punto y un error grave 2 puntos.

Nombre y apellidos:

17/05/2024

Firma:

1. Vamos a escribir una versión mejorada del método de ordenación de la burbuja (bubble sort) en Java. **(5 p)**.

- El método tiene que permitir ordenar un array de números enteros de dos dimensiones (es decir una matriz).
- El array lo tiene que recibir el array por parámetro.
- El método tiene que poder realizar la ordenación de forma ascendente o descendente. El sentido de la ordenación (ascendente o descendente) debe ser recibida mediante parámetro que tiene que ser booleano, por ejemplo si ese booleano vale true ordenamos ascendientemente y descendientemente en caso contrario.

Por ejemplo, dada la siguiente matriz:

```
[[5, 2, 8 ],
 [9, 3, 1 ],
 [7, 6, 4 ]]
```

Ordenada de manera ascendente daría:

```
[[1, 2, 3 ],
 [4 5 6 ],
 [7 8 9 ]]
```

Y de manera descendente daría:

```
[[ 9 8 7 ],
 [6 5 4 ],
 [3 2 1 ]]
```

PROGRAMACIÓN - 2

2. Vamos a implementar un método de ordenación que organice los elementos de una matriz de dos dimensiones en en espiral. El método deberá recibir dos parámetros: la matriz de números y un valor booleano. Este valor booleano determinará el sentido de la ordenación de los elementos; si el booleano es true, la ordenación deberá ser en el sentido de las agujas del reloj, y si es false, deberá ser en sentido contrario a las agujas del reloj. **(5 p)**.

Por ejemplo, dada la siguiente matriz:

```
[[ 5, 2, 9 ],  
[ 4, 6, 8 ],  
[ 3, 7, 1]]
```

Así es ordenada en espiral (sentido de las agujas del reloj):

```
[[ 1, 2, 3 ]  
[ 8, 9, 8 ]  
[ 7, 6, 5 ]]
```

Matriz ordenada en espiral (sentido contrario a las agujas del reloj):

```
8 8 9  
7 1 2  
6 5 3
```