

UT9_PD1

▼ Property	COMPLETED
📅 Date	@July 7, 2022
☰ BLOCKED	

Ejercicio 1:

1.

```
public int[] generarDatosAscendentes() {  
    int [] copiaAscendente = new int[TAMANIO_MAX];  
    for (int i = 0; i < TAMANIO_MAX; i++) {  
        copiaAscendente[i] = i;  
    }  
    return copiaAscendente;  
}
```

Hace un for entre el valor 0 hasta el TAMANIO_MAX y va agregando cada valor a la copiaAscendente

2.

```
for (int i = 0; i < TAMANIO_MAX; i++) {  
    copiaDescendente[i] = TAMANIO_MAX - i;  
}
```

Realiza el mismo bucle donde agrega el TAMANIO_MAX restando el valor de i y agregandolo al vector copiaDescendete

3.

```
public int[] generarDatosAleatorios() {  
    Random rnd = new Random();  
    int[] datosGenerados = new int[TAMANIO_MAX];  
    boolean[] datosUtilizados = new boolean[TAMANIO_MAX];  
    for (int i = 0; i < datosGenerados.length; i++) {  
        int j = rnd.nextInt(TAMANIO_MAX);  
        while(datosUtilizados[j]){  
            j = (j + 1) % TAMANIO_MAX;  
        }  
        datosGenerados[j] = i;  
    }  
}
```

```

        datosUtilizados[j] = true;
    }
    return datosGenerados;
}

```

Realiza el mismo bucle, generando un entero aleatorio j mediante la función `rnd.nextInt`. El valor de j esta entre 0 y `TAMANIO_MAX`. En el segundo bucle chequea que el vector `datosUtilizados` sea true, sino lo es guarda el valor de i en la posición j , caso contrario va aumentando hasta encontrar una posición libre.

En otras palabras se generan valores de i a `TAMANIO_MAX` y se guardan en posiciones aleatorias j , si j ya esta ocupada va aumentando en 1 y dividiendo entre el largo del arreglo (para no salirse del índice) e insertado cuando encuentre un espacio vacío

No va a tener valores repetidos pues el dato que se guarda es el i del primer bucle que constantemente va incrementando.

$$T(TAMANIO) = O(TAMANIO^2)$$

4. ¿Cuántos elementos contiene el vector de datos generado? → `TAMANIO_MAX` = 32000

Se podría pasar por parametro en el metodo mismo, o generar un constructor de la clase que reciba como parametro el `TAMANIO_MAX`

5. asiendo magia

```

for (int i = 0; i < TAMANIO_MAX-1; i++) {
    if (datosGenerados[i]>datosGenerados[i+1]){
        return false
    }
}
return true

```

Ejercicio 2:

→ INCERCIÓN:

3. El resultado del array es el siguiente: 5,4,3,2,1. Es decir ordena pero al revez.
- 4.

```

while ((j >= 0) && (datosParaClasificar[j+1] > datosParaClasificar[j]))
//CORRECCION:

```

```
while ((j >= 0) && (datosParaClasificar[j+1] < datosParaClasificar[j]))
/* evalua si los datos para clasificar en la posicion actual es mayor
a la siguiente, y no al revez
*/
```

5.

```
public static boolean estaOrdenado(int[] v) {
    for (int i = 0; i < v.length - 1; i++) {
        if (v[i] > v[i + 1]) {
            return false;
        }
    }
    return true;
}
```

7.

$$T(n) = O(n^2)$$

donde n = largo del vector a clasificar

→ **SHELLSORT:**

3. No anda el metodo, nunca sale de este while, ejecuta eternamente

```
while (j >= 0) {
    if (datosParaClasificar[j] > datosParaClasificar[j + inc]) {
        intercambiar(datosParaClasificar, j, j + inc);
        j = j--;
    }
}
```

4.

```
...
if (datosParaClasificar[j] > datosParaClasificar[j + inc]) {
    intercambiar(datosParaClasificar, j, j + inc);
    j = j--;
}
...
// CODIGO SIN ERROR:
...
while (j >= 0) {
    if (datosParaClasificar[j] > datosParaClasificar[j + inc]) {
        intercambiar(datosParaClasificar, j, j + inc);
        j = j - inc;
    }
}
```

```
        else{
            j = -1;
        }
    }
    ...
    // EL ERROR ESTABA EN:
```

$$T(n) = O(n^{1.26})$$

→ **BURBUJAS:**

3. Da un nullpointer exception 😞
4. El error estaba en que seteaba como nulo el parametro

```
datosParaClasificar = null;
```

$$T(n) = O(n^2)$$