

UT9_TA7

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

2. En comparacion a heapsort este es mucho mas rapido, pero consume memoria auxiliar. Para las cuentas
3. Es estable
- 4.

```
CuentaPorDistribución
begin
  for i = u to v hacer Cuenta[i] = 0
  for j = 1 to N incrementar Cuenta[Kj]
  for i = u +1 to v
    Cuenta[i] = Cuenta[i] + Cuenta[i-1]
  for j = N downto 1 hacer
    i = Cuenta[Kj]
    S[i ] = R[j ]
    Cuenta[Kj] = i-1
end
```

$$T(n) = O(n + k)$$

donde n es el largo del vector y k el tamaño del vector auxiliar

- 5.

```
private int[] ordenarPorCuenta(int[] datosParaClasificar, int maximo) {
    int[] cuenta = new int[maximo + 1];
    for (int i = 0; i < datosParaClasificar.length; i++) {
        cuenta[datosParaClasificar[i]]++;
    }
    for (int i = 1; i < maximo + 1; i++) {
        cuenta[i] += cuenta[i - 1];
    }
    int[] salida = new int[datosParaClasificar.length];
    for (int i = datosParaClasificar.length - 1; i >= 0; i--) {
        int j = cuenta[datosParaClasificar[i]] - 1;
        salida[j] = datosParaClasificar[i];
    }
}
```

```
        cuenta[datosParaClasificar[i]]--;  
    }  
    return salida;  
}
```