Examen de Algorítmica y Complejidad (Plan 2014) — **30 de junio de 2023**

Nº matrícula:_____ **Nombre:** _____

Apellidos: _____

1) **Problema** (**5 puntos**). Dada una cadena de ceros y unos, encontrar la secuencia más larga de unos basado en la estrategia de **Divide y Vencerás** con complejidad en el caso peor[1] de **O(N·logN)** (donde N es el tamaño del vector). Se pide implementar un algoritmo que deberá tener la siguiente cabecera:

```
int oneSubArray (int[] v)
```

donde v es el array que contiene la secuencia de ceros y unos.

Ejemplos:

v1	[1, 1, 1, 1, 1, 0, 1]	**SOL = 5**
v2	[0, 0, 0, 0, 1, 1, 1]	**SOL = 3**
v3	[1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0]	**SOL = 5**
v4	[0, 0, 0, 0, 0]	**SOL = 0**
v5	[1, 0, 1, 0, 1, 0, 1, 0, 1]	**SOL = 1**

Aclaraciones: Se podrán implementar todos los métodos y clases adicionales que se consideren necesarios.

```
int oneSubArray(int[] v){
    return oneSubArrayAux(v, 0, v.length-1);
}

int oneSubArrayAux(int[] v, int i0, int iN){
    if(i0==iN){
        return v[i0];
    } else{
        int k = (i0 + iN) / 2;
        int l1 = oneSubArrayAux(v, i0, k);
        int l2 = oneSubArrayAux(v, k+1, iN);
        int l3 = oneSubarrayCruzada(v, i0, k, iN);
        return Math.max(l1, Math.max(l2, l3));
    }

}
```

[1] Desarrollar un algoritmo que tenga una complejidad diferente a O(N · logN) en el caso peor conllevará una puntuación de 0 en la pregunta.

```
int oneSubarrayCruzada(int[] v, int i0, int k, int iN) {
    int len_i = 0;
    int len_j = 0;
    int i=k;
    int j=k+1;

    while(i >= i0 && v[i] == 1){
        len_i++;
        i--;
    }
    while(j <= iN && v[j] == 1){
        len_j++;
        j++;
    }

    int full_len = len_i + len_j;

    return full_len;
}
```