



Nº matrícula: _____ Nombre: _____

Apellidos: _____

Sesión 1.3. Divide y Vencerás

Dado un *array*, **vector**, de números enteros **ordenados** y **diferentes entre sí**. Encontrar un índice i tal que **vector**[i]== i .

Ejemplo:

0	1	2	3	4	5	6	7	8	9	10
-10	-2	0	3	7	9	19	28	30	42	55

Para $i=3$, se tiene que **vector**[3]==3

Pregunta 1. Implementa en Java un algoritmo basado divide y vencerás **con complejidad $O(\log N)$** en el caso peor que resuelva el problema.

```
int elementoEspecial(int[] vector){  
}
```

Pregunta 2 (3 puntos). Calcula la complejidad del algoritmo implementado en el caso peor (justifica la respuesta)

Sesión 1.2. Solución.

Pregunta 1. Implementa en Java un algoritmo basado divide y vencerás con complejidad $O(\log N)$ en el caso peor que resuelva el problema.

```
int elementoEspecialAux(int[] vector, int i0, int iN){
    if (i0 == iN)
        if (vector[i0] == i0) return i0; else return -1;
    else {
        int k = (i0 + iN) / 2;
        if (vector[k] > k)
            return elementoEspecialAux(vector, i0, k);
        else if (vector[k] == k)
            return k;
        else
            return elementoEspecialAux(vector, k + 1, iN);
    }
}

int elementoEspecial(int[] vector) {
    return elementoEspecialAux(vector, 0, vector.length - 1);
}
```

Pregunta 2 (3 puntos). Calcula la complejidad del algoritmo implementado en el caso peor (justifica la respuesta)

El algoritmo implementado obedece a la siguiente ecuación de recurrencia en el tiempo para el caso peor:

$$T(N) = T(N/2) + \Theta(1)$$

Aplicando el teorema maestro para el caso 2 (ya que $\log_2 1 = 0$), la complejidad del algoritmo es:

$$\Theta(\log N)$$