Examen de Algorítmica y Complejidad (Plan 2014) — **14 de noviembre de 2016**

Nº matrícula:_____ **Nombre:** _____

Apellidos: _____

Problema. Dado un array de números ordenados en los que todos ellos aparecen dos veces salvo uno, se desea buscar el único elemento que aparece sólo una vez.

Ejemplo:

1	1	**4**	5	5	7	7	8	8	9	9

Para este vector, se devolvería el valor 4.

a) Implementar un algoritmo en Java, basado en Divide y Vencerás, que solucione el problema expuesto, con **complejidad O(log N)** en el caso peor[1] (donde N es el tamaño del vector).

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

int elementoSolitarioAux(int[] vector, int i0, int iN){
    if (i0 == iN)
        return vector[i0];
    else {
       int k = (i0 + iN) / 2;
       if (vector[k-1] == vector[k])
         //Se encuentra en [i0...k-2] o bien en [k+1...iN]
         if ((k-2-i0+1)%2==0)
            return elementoSolitarioAux(vector, k + 1, iN);
         else
            return elementoSolitarioAux(vector, i0, k-2);
       else if (vector[k] == vector[k+1])
         //Se encuentra en [i0...k-1] o bien en [k+2...iN]
         if ((k-1-i0+1)%2==0)
            return elementoSolitarioAux(vector, k + 2, iN);
         else
            return elementoSolitarioAux(vector, i0, k-1);
       else
          return vector[k];
   }
 }
```

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

b) Justifica que la complejidad del algoritmo desarrollado en el apartado anterior para el caso peor es O(log N).

El algoritmo implementado obedece a la siguiente ecuación de recurrencia en el tiempo para N>1:

$$T(N) = T(N/2) + O(1)$$

Esta ecuación es del tipo $T(N) = p \cdot T(N/q) + f(N)$, donde $f(N) \in O(N^a)$, con p=1, q=2 y a=0, por lo que podemos aplicar el Teorema Maestro. Dado que $\log_q(p) = \log_2(1)=0$ y a=0 nos encontramos en el caso 2º del Teorema maestro ($a=\log_q(p)$), por lo que la complejidad del algoritmo es: $T(N) \in O\left(N^{\log_q(p)} \cdot \log N\right) = O(N^0 \log N) = O(\log N)$