

# PREGUNTA 3

a)

numeros Magicos (A, i, f)

$$m = \lfloor \frac{i+f}{2} \rfloor$$

if  $m = A[m]$ : return m

else if  $m > A[m]$ :

numeros Magicos (A, i, m-1)

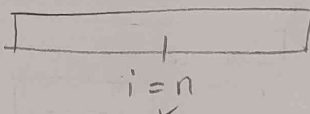
else if  $m < A[m]$ :

numeros Magicos (A, m+1, f)

$O(\log(n))$

JORGE DE LOYNECHE

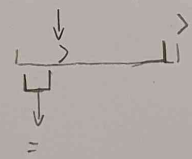
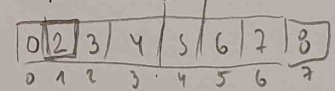
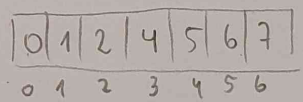
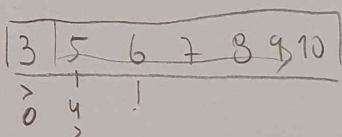
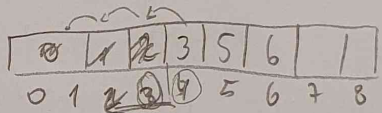
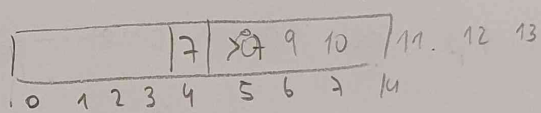
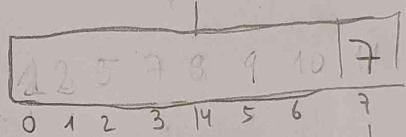
20.538.979-2



$i < n$

2 4 15

1 2 3 4 5



b)

numeros Magicos (A, i, f)

$$m = \lfloor \frac{i+f}{2} \rfloor$$

if  $m = A[m]$ : return m

elif  $m > A[m]$ :

return numeros Magicos (A, i, m-1)

elif  $m < A[m]$ :

return numeros Magicos (A, m+1, f)

else : return null

b) El peor caso de mi algoritmo es cuando no existe ningún número mágico, o este se encuentra en "i" o "f".

Para cada iteración, se chequea si el valor central del array es un número mágico, si no lo es, se divide el array y se vuelve a chequear. Esto es así hasta que quede un solo elemento en el sub-array.

Como se ejecuta dividiendo el tamaño hasta quedar en uno la complejidad del peor caso es de  $O(\ln(n))$

y en comparación su mejor caso es  $O(1)$ , con un número mágico en la mediana del arreglo.