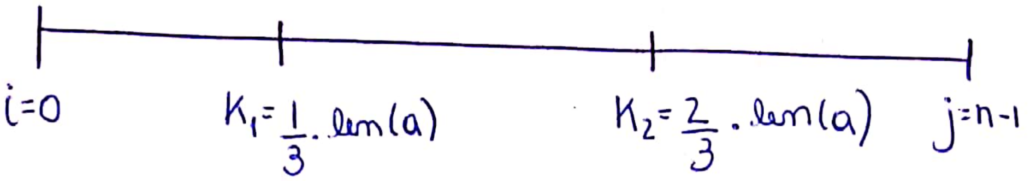
TAREA N°3 ALGORITMOS

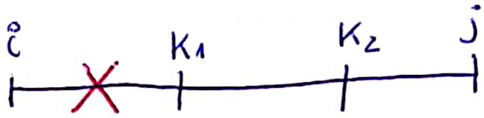


arreglo

$n = len(arreglo)$

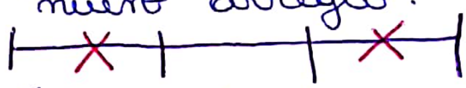
tenemos el arreglo dividido en 3 subdivisiones.
Nos ponemos en 3 casos

CASO I: $a[K_1] < a[K_2]$



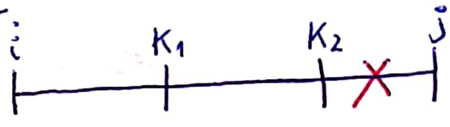
se elimina el primer tercio y se genera un arreglo que va desde K_1 hasta j; $a[K_1:j]$, se calcula K_1, K_2, j, i para este nuevo arreglo.

CASO II: $a[K_1] = a[K_2]$



se elimina el primer y el último tercio y se genera un nuevo arreglo que va desde K_1 hasta K_2: $a[K_1:K_2]$, se calculan entonces K_1, K_2, i, j para este nuevo arreglo:

CASO III: $a[K_1] > a[K_2]$



se elimina el último tercio y se genera un nuevo arreglo que va desde i hasta K_2: $a[i:K_2]$

Además se vio que existían 3 casos bases:

CASO BASE I: largo del arreglo igual a 1
si el largo del arreglo es 1 entonces el uno elemento dentro del arreglo es el maximo

CASO BASE II: largo del arreglo igual a 2 ⇒ comparar los dos elementos del arreglo y ver cual es mayor. El mayor de los 2 será el máximo de arreglo.

CASO BASE III: largo del arreglo igual a 3. Comparar los 3 elementos dentro del arreglo. El mayor de éstos, será el máximo del arreglo,

Arreglo inicial 7

Ej: $\underbrace{62}_{i=0}$ 78 $\underbrace{71}_{K_1}$ 60 $\underbrace{55}_{K_2}$ 21 20 $\underbrace{15}_{j=7}$

len = 8

$K_1 = 8 // 3 = 2$

$K_2 = \frac{2}{3} \cdot 8 = 4$

CASO III $a[K_1] > a[K_2]$

$\Rightarrow$ elimino el último tercio

$\Rightarrow a = \underbrace{62}_{i}\ \underbrace{78}_{K_1}\ \underbrace{71}_{K_2}\ 60\ \underbrace{55}_{j}$

len = 5

$K_1 = 5//3 = 1$

$K_2 = \frac{2}{3} \cdot 5 = 2$

CASO III $a[K_1] > a[K_2]$

$\Rightarrow$ elimino el último tercio

$\Rightarrow a = 62 \quad 78 \quad 71$

CASO BASE III len(a) = 3

el máximo de a es 78

Para calcular $T(n) = aT\left(\frac{n}{b}\right) + f(n)$

n: tamaño del arreglo

a: subproblemas en la recursión, en este caso
$a = 3$

$\frac{n}{b}$: tamaño de cada subproblema, en este caso
$b = \frac{3}{2}$

f(n): costo fuera de la llamada recursiva, en este caso $f(n) = \frac{n}{3}$

Calculando para $a = 3$ y $b = \frac{3}{2}$

$f(n) = \frac{n}{3} = \Theta(n^c \log^k n)$ donde $K = 0$ y $c = 1$

calculando $\log_b a = \log_{\frac{3}{2}} 3 = 2.709 > C$

$\Rightarrow T(n) = \Theta(n^{\log_b a})$

$\Rightarrow \boxed{T(n) = \Theta(n^{2.709})}$ muy ineficiente ya que es un tiempo cercano a n^3

Escaneado con CamScanner