

PREGUNTA 1

a)

Comenzando con sus subrutinas
Left Right Shake recibe un arreglo
de largo finito con sus respectivos
indices. Su unica iteracion es dentro
de un loop for, el cual es finito
trivialmente y siempre termina, \therefore Termina

Right Left Shake es analogo al anterior
solo que esta vez itera inversamente, pero
el for siempre termina, \therefore Termina

Cocktail Shaker Sort recibe un array finito. este entra en un loop while
donde se verifica que aun existan elementos por ordenar.
La brecha entre estos indices se acota cada vez en las lineas 5 y 7
del codigo. Sabemos que este es el caso porque en Left Right Shake
el valor retornado siempre sera mayor o igual a 0, lo mismo ocurre
con Right Left Saker. \therefore termina ya que se acota la diferencia
hasta incumplir la condicion del while.

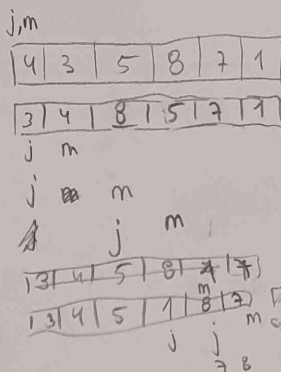
b) Left Right Shake

$P(n)$: Los elementos $A[j, m+1]$ se encuentran ordenados.

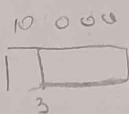
$P(n)$: j se posiciona en el ultimo arreglo de inversion necesario
por lo que todo elemento mayor o igual a el estan
relativamente ordenados.

$P(n)$: Asumimos se cumple

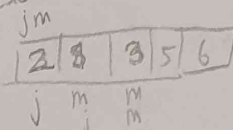
$P(n+1)$ = Tomando nuevos indices se cumple el caso base sobre el
nuevo dominio, que como $P(n)$ es verdadero ya esta
ordenado y $\therefore P(n+1)$ este ordenado.



JORLE DE
COYENECHE
20.538.979-2



Left Right ordena
de izquierda hacia K
Right Left ordena
desde f hacia +



Pⁿ = ... 2

JORGE DE LOYNECHE

a) HI: $P(n)$ se cumple

~~II: $P(n+1)$~~

2 1 4 3
↕ ↕

12 3 4 5 6 8 7

b) ~~Si $P(n)$ se cumple tenemos~~

EI LeftRight Shaker de limite entre j y $m+1$ elementos ordenados, por lo que al no moverse j en las ultimas iteraciones, este lo delimita como el punto de termino del siguiente shaker, el cual hace lo mismo pero para el primer shaker, de esta manera cuando j ya no se mueve, se acabará el loop while y estará ordenado el arreglo.

c) El mejor caso es tener una lista ordenada, ya que mmo aumentará y llegará a ser cercano a 220. m hasta f

d) $O(n \log(n))$

$P(n)$: Los elementos $A[i, t]$ y $A[k, f]$ se encuentran ordenados

$P(1) :=$ Se ejecuta una vez LeftRight y Right Left por lo que estos costados se ordenan ✓

$P(n) =$ Asumimos se cumple

$P(n+1) :=$ Al acotarse los bordes de ordenamiento se ordenan con Left Right y Right Left la siguiente seccion que excluye $A[i, t]$ y $A[k', f]$ que por $P(n)$ sabemos se encuentran ordenados \therefore
 $P(n+1)$ se encuentra ordenado.

1 5 3 7 2

1 5

1 3 5

1 3 5 7

1 3 5 2 7
 ↓
 m

falta por ordenar

7 5 3 2 10

5 7

5 3 7

5 3 2 7 10
 ↑ ↑
 j i