Comenzado en	Monday, 13 de December de 2021, 18:20
Estado	Terminados
Finalizado en	Monday, 13 de December de 2021, 19:00
Tiempo empleado	40 mins
Puntos	2.00/2.00
Calificación	**10.00** de un total de 10.00 (**100**%)

Pregunta **1**

Completada

Puntúa 1.00 sobre 1.00

a) Escribir una función que inserta elementos en una lista de manera de mantenerla ordenada de menor a mayor. De esta forma cada operación **Head** sobre la lista devuelve el elemento más chico almacenado en ella.
Inserta:: (Ord a)=> a->[a]->[a]

b) Escriba una función **Qsort::(Ord a)=>[a]->[a]**. **Sin utilizar listas por comprensión.**

*Observacion: Escriba una funcion **particion** que reciba como argumento, un valor de referencia o pivot y a una lista de valores del mismo tipo que el pivot. Esta funcion da como resultado una tupla con dos listas (l1 , l2) ... de modo que en l1 estan todos los valores de la lista original que son menores o iguales que el pivot y en l2 todos los mayores que el pivot.*

particion :: Ord a => a->[a]->([a],[a])

qsort.hs

inserta.hs

Comentario:

Pregunta **2**

Completada

Puntúa 1.00 sobre 1.00

Recordemos que la función de biblioteca **ZIP**, recibe como argumento dos listas (x:xs) e (y:ys) y produce una lista de tuplas (i,j) donde los i provienen de la primera lista y los j de la segunda. Cuando una lista es mas larga que la otra, el resultado contempla solo los pares hasta donde pudieron formarse....

Ej zip [1,2,3] [10..] = [(1,10), (2,11), (3,12)]

a)Escriba una versión personal de la función zip, llamada **miZip**

miZip :: [a]->[b]->[(a,b)]

b) Utilizando **miZIP y listas por comprension**. Escriba una función que realice el producto escalar de dos listas. Donde producto escalar estaría definido como la suma de los productos uno a uno, componente a componente de cada lista. Si una lista tuviera más elementos que la otra, al agotarse uno de los operandos se detiene la suma.

a1*b1 + a2*b2+...+ai*bi+...+ an*bn.

mizip-producto.hs

Comentario:

◄ Avisos

Ir a...

TADs ►