

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.

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

Observación: Escriba una función **particion** que reciba como argumento, un valor de referencia o pivot y a una lista de valores del mismo tipo que el pivot. Esta función da como resultado una tupla con dos listas (l1 , l2) ... de modo que en l1 están 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])

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 sólo la función miZip escriba la función “adyacentes”.

Dada una lista genera otra lista de pares. Cada elemento de la lista argumento esta en tupla con el elemento siguiente.

adyacentes :: [a] -> [(a,a)]

Ejemplo: adyacentes [1,2,3,4,5] => [(1,2), (2,3), (3,4),(4,5)]

Una cola de prioridad es una estructura de datos que almacena elementos “clasificables”. Con la particularidad que, cuando se saca uno de ella siempre se extrae el elemento con menor clave, de ahí su nombre pues clasifica los elementos en función de su prioridad. La prioridad más baja primero.

Las funciones que manipulan a la cola de prioridad, son:

mkqpr: Instancia una nueva cola de prioridad vacía.

addqpr: Agrega un nuevo elemento a la cola de prioridad

nextqpr: Devuelve el elemento con clave más baja de la Cola de prioridad.

popqpr: devuelve una cola de prioridad donde se ha quitado el nextqpr.

Defina el TAD ColaPrioridad, e implemente el mismo utilizando un árbol binario de búsqueda como estructura de almacenamiento.

Escribir todas las funciones necesarias para la manipulación de la estructura subyacente, es decir para manipular el árbol.

Sugerencia: Recordar como extraer el elemento con clave más pequeña de un árbol.