Functional Programming
2020-2021 Spring
Homework 2

This assignment is based on of Programming in Haskell assignment from University of Zagreb.

In this homework, you are asked to implement a min-heap in Haskell by using the following definitions in this document.

We can define a min-heap recursively as follows:

```
data Heap n = Leaf n | Branch (Maybe n, (Maybe Heap n, Maybe Heap n))
```

We can represent an empty heap with Branch (Nothing, (Nothing, Nothing)).

A heap constructed from numbers [5, 1, 2, 4, 3, 6] could look like this:

```
Branch
(Just 1, (
   Branch (Just 3, (Just Leaf 5, Just Leaf 4)),
   Branch (Just 2, (Just Leaf 6, Nothing))
))
```

Each node can be either a branch or a leaf, but not both.

Using such structure, define the following functions:

1- An empty heap value:

```
empty' :: Heap n
```

2- A function to insert an item into the heap:

```
insert' :: Ord n => Heap n => n => Heap n
```

insert function should add element to the end of the heap (next available place to insert an element) first , then move it up recursively until heap condition is satisfied.

3- A function to create heap from a list of items:

```
fromList' :: Ord n => [n] => Heap n
```

fromList function should call insert function for each element in the given list without modifying the order of elements in the list.

4- A function to check whether an element exists in the heap or not.

```
lookup' :: Ord n => n => Heap n => Int
```

lookup function should return 1 if the given element exists, else 0.

5- A function to get maximum element in the heap.

```
maxElement' Ord n => Heap n => Maybe n
```

maxElement function should return Nothing if the heap is empty.

6- A function to delete a given element from the heap.

```
delete' :: Ord n => n => Heap n => Heap n
```

delete function should return the unmodified heap if the given element does not exist in the heap.

7- A function to check whether given heap is a valid min-heap or not.

```
isValidMinHeap' :: Ord n => Heap n => Int
```

isValidMinHeap function should return 1 if the given heap is a valid min-heap (including empty heap), else 0.

Example Heap State with Step-by-Step Insertion

Inserting numbers [5, 1, 2, 4, 3, 6].

Empty Heap:
Branch (Nothing, (Nothing, Nothing))

Insert 5:
Branch (Just 5, (Nothing, Nothing))

Insert 1:
Branch (Just 1, (Just Leaf 5, Nothing))

Insert 2:
Branch (Just 1, (Just Leaf 5, Just Leaf 2))

Insert 4:
Branch (Just 1, (Branch (Just 4, (Just Leaf 5, Nothing)), Just Leaf 2))

Insert 3:
Branch (Just 1, (Branch (Just 3, (Just Leaf 5, Just Leaf 4)), Just Leaf 2))

Insert 6:
Branch (Just 1, (Branch (Just 3, (Just Leaf 5, Just Leaf 4)), Branch (Just 2, (Just Leaf 6, Nothing))))

Notes

You cannot use any external library.
You must explain your code using inline comments.