Sorting with a Heap

In this assignment, given a sequence of n distinct numbers $a_1, a_2, \ldots, a_n$, we will sort them in an increasing order to obtain a sequence $b_1, b_2, \ldots, b_n$ such that $b_1 \leq b_2 \leq \ldots \leq b_n$. We will use a heap data structure in order to do this sorting. Recall that in a min-heap, the elements are stored in a binary-tree like structure where the value at a parent node is smaller than the values at the child nodes. Here is how we will use a min-heap to sort a sequence of numbers.

Step 1: Take the number n as well as n distinct numbers as input from the user, and insert these n numbers into a list L.

Step 2: Build a min-heap with the numbers in L. Let us use an array-based implementation for a min-heap. In the process of building the min-heap, we will rearrange elements in the list L. Let us also denote the resulting min-heap by L.

Step 3: Write a function that takes the min-heap L and the number of elements in L as input, and removes and returns the minimum element of the min-heap L. Note that after we remove the minimum element of L (the root of the min-heap), we copy the last element of the array into the location $L[0]$. This may violate the min-heap property. Hence, one needs to ensure that the min-heap property is restored. (You may want to write a function that takes the min-heap L and an index as input, and ensures that the heap property holds at that node. Note that you may have to call this function on all nodes on a path from the root to a leaf.)

Step 4: Once we have the above functions ready, we go as follows. The first time we call the above function, it will return the smallest element in L and the number of elements in L will be reduced to $(n-1)$. The second time this function is called, it will return the second smallest element and so on. You will output a sorted sequence of the numbers in this manner.

Expected input/output behaviour: Given an input sequence of n numbers, print the elements of the resulting heap, starting from $L[0]$ to $L[n-1]$. In the end, print the sorted sequence of n numbers.

Submission: Name your submission as RollNo-Assign5.py and upload on Google classroom.