

10

DATA STRUCTURES

Lab Exercise # 10: Heaps

Download the startercode from Brightspace that contains a **min Heap** class. Your task is to write the following missing methods for the Heap class.

Important: Minimum value of the heap must be stored at index 1 of the internal array i.e., index 0 of the array will remain unused for arithmetic simplicity. The **display** method of the starter code prints elements of the heap starting from index 1.

1. **int Heap::getMin()** (0.5 Points)
This method returns (but doesn't remove) the minimum value from the Heap. This method should throw an exception if the Heap is empty.
2. **int Heap::parent(k)** (0.5 Points)
This method returns the index of the parent of the node at index k.
3. **int Heap::left(int k)** (0.5 Points)
This method returns the index of the left child of the node at index k.
4. **int Heap::right(int k)** (0.5 Points)
This method returns the index of the right child of the node at index k.
5. **void Heap::insert(int key)** (1 Points)
This method inserts an element/Node in the Heap keeping the Heap property intact.
6. **int Heap::removeMin()** (1 Points)
This method removes and returns the minimum value from Heap keeping the Heap property intact. An exception should be thrown if the Heap is empty.
7. **void Heap::bubbleup(int k)** (2 Points)
This method bubbles up the node at index k.
8. **void Heap::bubbledown(int k)** (2 Points)
This method bubbles down the node at index k.

Hints:

1. if the node has no children do nothing.
2. if the node has only a left child which is smaller than the key of the node then swap the node with the left child.
3. if the node has both children, then swap the node with a smaller child.
4. keep repeating steps 1-3 until the node reaches its correct position.

9. **sort()** (1 Points)
This method should sort the Heap. Do not use typical sorting methods e.g., BubbleSort, SelectionSort etc., instead keep extracting the min value from the heap and store it in a dynamically created auxiliary array. Replace the original internal array of the Heap with the new array.

Comments: (1 Points)
Comments are a very important part of any program. You should always write comments in your code, even if not explicitly asked.

Desired Output:

```
> g++ lab10.cpp && ./a.out
```

```
-----
display           : Display the Heap
insert <key>       : Insert a new element in the Heap
getMin            : Get the element with Min. Key in the Heap
removeMin         : Remove the element with Min. Key from Heap
sort              : Sort the Heap
help              : Display the available commands
exit/quit         : Exit the program
-----
```

```
>demo
```

```

      5
    7      19
  24      8      23      22
-----
```

```
>getMin
Min Key = 5
```

```
>insert 3
```

```

      3
    5      19
  7      8      23      22
24
-----
```

```
>removeMin
3 has been removed..!
```

```

      5
    7      19
  24      8      23      22
-----
```

```
>sort
```

```

      5
    7      8
  19      22      23      24
-----
```

CODE OF CONDUCT

All assignments are graded, meaning we expect you to adhere to the academic integrity standards of NYU Abu Dhabi. To avoid any confusion regarding this, we will briefly state what is and isn't allowed when working on an assignment/lab-task.

Any documents and program code that you submit must be fully written by yourself. You can, of course, discuss your ideas with fellow students, as long as these discussions are restricted to general solution techniques. Put differently, these discussions should not be about concrete code you are writing, nor about specific results you wish to submit. When discussing an assignment with others, this should never lead to you possessing the complete or partial solution of others, regardless of whether the solution is in paper or digital form, and independent of who made the solution, meaning you are also not allowed to possess solutions by someone from a different year or course, by someone from another university, or code from the Internet, etc. This also implies that there is never a valid reason to share your code with fellow students, and that there is no valid reason to publish your code online in any form.

Every student is responsible for the work they submit. If there is any doubt during the grading about whether a student created the assignment themselves (e.g. if the solution matches that of others), we reserve the option to let the student explain why this is the case. In case doubts remain, or we decide to directly escalate the issue, the suspected violations will be reported to the academic administration according to the policies of NYU Abu Dhabi.

<https://students.nyuad.nyu.edu/campus-life/community-standards/policies/academic-integrity/>