ITU Computer Engineering Department
BLG 223E Data Structures, Summer 2021
Homework #7
Due August 10, 2021 23:59

Definition

You are asked to implement a program that implements a generic integer heap data structure that can act like a minheap or maxheap by the anonymous function provided at the construction time. Please pay attention to the following details in your implementations:

- Your heap implementation should contain a **single** addition, removal, heapification, heap printing and sorted printing function.
- Your remove function should remove the first occurence of an element in the heap array.
- Your sorted printing function should print the elements in the heap in a sorted way (MaxHeap: Decreasing order, MinHeap:Increasing order)
- Your heapification function should work as bottom-up heapification process.
- Your heap printing function should print the heap in breadth first way.
- Explicit usage of operator overloading is not allowed.
- It is recommended that you start by implementing a basic max or min heap and then modify with anonymous function usage.
- You shall build your solution on the following class definition. You shall add any necessary number of private/public properties and methods as you like.

```
class Heap{
    private:
        int capacity;
        int size;
        int* arr;
        std::function<int(int,int)> minmax;

    public:
        Heap(int,std::function<int(int,int)>);
        ~Heap();
};
```

Input-Output

Your program should accept a filename as a command line parameter. In the accepted file, each line will contain a distinct command and parameters that should be processed by your program. In the input, the following commands may be given:

- Each file starts with an initial line containing the heap type (MAXHEAP or MINHEAP), followed by the heap capacity as a single integer and **optionally** followed by an unknown number of integers that would be used in the heapification process.
- File will continue with unknown number (may be 0) and order of ADD and DEL commands. Each ADD/DEL command is followed by an integer to be added to heap or removed from the heap.
- The last two lines of the file would be PRINT and SORT. These lines are guaranteed to appear at the end of the file in a fixed order. PRINT command prints the contents of the heap following a breadth first traversal to the command line followed by a newline. SORT prints the elements in a sorted way (MaxHeap: Decreasing order, MinHeap:Increasing order).

Input will not contain syntactical errors but it may contain removal of non-existing elements and such cases that you may need to handle. See the provided test cases for examples.

Deliver

Please zip and deliver the directory structure defined below:

- HW7: Topmost folder, that will contain all the folders in your submission. No other files should be present under this folder in your submission.
- HW7/src: Contains all the *.cpp files
- HW7/src/main.cpp: Contains your main function and other code you want to deliver.
- HW7/src/Heap.cpp: Contains the code for your heap.
- HW5/include: Contains all the header files you use
- HW5/bin: An empty directory that will contain objective files when your project is compiled

Please check the calico test file in the homework definition to see how your files will be compiled and tested.

Restrictions and Guidelines

- Compilation environment: Only the code that can be compiled by the environment of the container definition provided in ninova will be accepted.
- Testing of your program will be performed using Calico (https://calico.readthedocs.io/en/latest/). Test cases that will be used to test your homework is provided as an attachment in ninova.
- **STL usage is not allowed. Only exception is using a vector in collecting the numbers from the input file for heapification.**
- **This homework is for individual submissions**. Any kind of code sharing or code adaptation from an external source is strictly forbidden. Submitted code will undergo a plagiarism check process and plagiarsim penalties may be given if the submitted code's similarity is approved by the instuctor.
- Make sure you write your name and number in all of the files of your project, in the following format:
  /* @Author
  Student Name:<studentname>
  Student ID :<studentid>
  Date:<date>*/
- Only electronic submissions through Ninova will be accepted no later than deadline
- Use comments wherever necessary in your code to explain what you did.