

CSE101 Binary Heap Test Questions: Sink or swim, there is no other option

©C. Seshadhri, 2020

- All code must be written in C/C++.
- Please be careful about using built-in libraries or data structures. The instructions will tell you what is acceptable, and what is not. If you have any doubts, please ask the instructors or TAs.

1 Setting

You are already provided with a binary heap, that handles some operations. So far, there is code for “insert”, “findMax”, and “print”. So you already have the code to construct the heap.

The I/O format has been fixed. The input and output files are given as command line arguments. Each line of the input file is an operation on the binary heap. The “print” operation leads to printing the heap (as an array) in the output file.

For this test, you will have to implement one of the functions described below. *You cannot store the heap in some other data structure. You must manipulate the heap directly to perform these functions.*

1.1 The test questions

There are only one operation that you need to study for the test.

- `void extractMax()`. This is just the standard extract maximum function in a binary heap. It would be useful to code up the `heapify` function as well. (You are only provided with the header declaration for `extractMax`.)

1.2 How it works

The header (that you will get in the test) will already have the right function declarations. Your job is to simply code up the function (that you will only get one as a test question) in the file `binheap.cpp`. Note that if you design more functions to help with your code, you may need to declare them as well in `binheap.h`.

The I/O will be taken care of, so there will be specific operations (in the input file) to perform these functions. So like “i <INT>” (for insert) and “em” (for extract max). These will all be explained in a README file. You don’t need to deal with any of this, but it might help you debug (if you run on your own inputs).

You will also get a small test input and test output. To make life easier, I will actually give you access to my grading scripts (that you can run directly through a shell script). If it catches a bug in your code, it will give you a test input where your code file. So you will get to see your score *before* you submit.

The test is completely closed book. You cannot bring any written material, but you can get blank scratch paper. You will get and can only open the specific Codio box for the test. You cannot refer to any other codes, or even open any other Codio box. If you do so, it will be considered cheating and you will get -10 points.

1.3 What should you do

Umm...how about coding these functions in the Codio box I provided?

1.4 Other challenging questions

This is a collection of other questions, that might be good practice for exercising your skills. (And great interview practice.) For these, you don’t need to write code, but you might want to figure out the logic.

- Code up HeapSort.
- Delete the k th largest element from the heap. There is a simple algorithm whose running time is $O(k \log n)$, where n is the number of elements in the heap. There is a more clever algorithm whose running time is $O(k \log k)$. (Hint: this solution uses another heap.)