

CMSC 123: Data Structures

1st Semester AY 2020-2021

Prepared by: CCS Templado & KBP Pelaez

Exercise 07: Heap ADT (Heapsort)

In heaps, deletion removes and returns the maximum or minimum value in the heap. It involves the strategy called *percolate down* which repeatedly compares and swaps values from a node with one of its children. The value is not actually deleted but stored at the end of the heap before the heap size is decreased.

If deletion is performed n times, then the array will contain the sorted values of the heap. For this exercise, you will implement heapsort by creating a copy of the heap and performing heapsort on that heap to preserve the values in the original heap.

Tasks

Implement and test the function below (also listed in `heap.h`):

`int* heapSort(HEAP *H);` - a function that returns an array of sorted values in the heap. If the heap is a min heap, this must return an array in descending order. Otherwise, this must return an array in ascending order.

This is a continuation of the previous exercise. Please make sure that you have a complete header file `heap.h` and a fully working implementation file `heap.c`.

Make sure to test your program using a shell file. Format is as follows:

1. Line 1 should contain either 0 or 1. 0 for MINHEAP and 1 for MAXHEAP.
2. Succeeding lines must contain one of the following commands:
 - `+` `i` - inserts `i` to the heap
 - `-` - deletes the root node
 - `~` - prints the sorted version of the heap
 - `p` - prints the heap
 - `E` - checks if the heap is empty
 - `F` - checks if the heap is full
 - `C` - clears the contents of the current heap
3. The last line in the file must contain the `Q` command for the program to terminate.

Submission

Submit your `heap.c` to Google Classroom.

Questions?

If you have any questions, approach your lab instructor.