

METU Department of Computer Engineering
CENG 707 Data Structures and Algorithms
Homework 3 – Fall 2020
Deadline: 04.01.2020
Yusuf Sahillioğlu

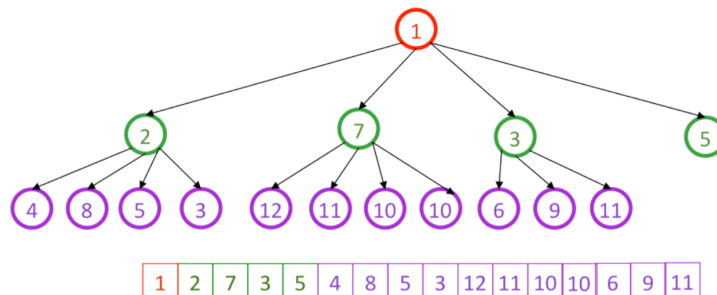
1) Binary Heap:

- a) [10 points] Implement a MinMaxHeap class that supports services we discussed in class (most of the implementation is in our lecture slides; you merely need to collect them in a CPP file). Note that in this class, you additionally have to allow the user to decide if s/he has a min-heap or a max-heap.
- b) [15 points] Implement the solution to the k largest element problem that runs in $O(k + (n-k)\log k)$ worst-case time (see lecture slides).
- c) [15 points] Repeat part b for the k smallest element problem.
- d) [10 points] Insert n random integers to an empty min heap. Use the same numbers with the buildHeap function. Write a function that decides whether 2 heaps are the same and use it to check your resulting heaps (insert vs. buildHeap). The function should belong to the MinMaxHeap class and expect another heap as parameter for testing.
- e) [10 points] Delete m integers from your binary heap (min or max) where $m < n$. Write an indentedPrint function that allows the current heap to be printed nicely to the screen, i.e., set indentation based on the level and position of the nodes. Here are 2 different min heap outputs:



2) Four Min Heap:

- a) [40 points] Implement a FourMinHeap class where you have the min heap property but each node can have up to 4 children. Support all the min heap functionalities and test them one by one: insert, deleteMin, buildHeap. Show your outputs after each action using the modified indentedPrint function. If the printouts are not easy to follow, you will get 0 points from this part. Note that, unlike binary heap, index computation is easier if you start with index 0, not 1. Here is a Four Min Heap:



Submission: Email to ys@ceng.metu.edu.tr a link to the folder of your source code, executables and screenshots.