

Priority Queue

In this task you have to implement a max priority queue using binary heap data structure. You need to implement the following functions:

1. **Insert(x)**: Inserts an element into the priority queue according to the priority of its key. Time complexity: $O(\log n)$.
2. **FindMax()**: Returns the element with largest key but does not remove it. Time complexity: $O(1)$.
3. **ExtractMax()**: Returns the element with largest key and delete the element from the heap. Time complexity: $O(\log n)$.
4. **IncreaseKey(i, newKey)**: Increase the key of the i^{th} element to *newKey*, and relocate it to maintain heap property. Time complexity: $O(\log n)$.
5. **DecreaseKey(i, newKey)**: Decrease the key of the i^{th} element to *newKey*, and relocate it to maintain heap property. Time complexity: $O(\log n)$.
6. **Print()**: Print the heap. Time complexity: $O(n)$.

You should write your program using features of object-oriented programming.

Input:

Create a menu for the six operations. Use 1 for insert, 2 for FindMax, 3 for ExtractMax, 4 for IncreaseKey, 5 for DecreaseKey, 6 for Print, and 7 for quit. Ask user to select an operation until option 7 is selected. Also prompt user for input any value which is required for the corresponding operations, i.e., insert, increaseKey, etc.

Submission Guidelines:

- a. In your local machine, create a new folder; **the name of the folder should be your 7 digit roll number.**
- b. Put all the source code files in the folder created in step (a).
- c. Finally, compress the folder created in (a) to produce a .zip file. **The name of the .zip file should be your 7 digit roll number.**
- d. Submit the .zip file.

Policy:

Copying from internet, classmate, seniors, or from any other source is strongly prohibited. **-100% marks will be 'rewarded' if any such copying is detected.**

Deadline:

Deadline is set at 20 July 2019, 11:00 pm BDT for all subsections.