Data Structure and Programming, Fall 2018
Programming Assignment #4

TA: r07943107@ntu.edu.tw

Due 2 p.m., December 17, 2018

1 Problem Statement

Priority queues are widely used in many applications (e.g., Dijkstra's algorithm and best-first search algorithms). Among the variety of data structures, the binary heap is a simple and common one to implement a priority queue.

A binary heap can be stored in an array and we can use their relative positions within that array to represent child-parent relationships. In particular, the parent of the key $array[i]$ is simply $array[\lfloor i/2 \rfloor]$. Besides, the left child and the right child of $array[i]$ are $array[2i]$ and $array[2i+1]$ respectively (if exist). Take the heap in Figure 1(a) for example. The heap can be stored in the array in Figure 1(b). The parent of B ($array[2]$) is A ($array[\lfloor 2/2 \rfloor]$), and the left child and the right of B is C ($array[2 \cdot 2]$) and D ($array[2 \cdot 2+1]$) respectively. Note that $array[0]$ is a dummy node.

This programming assignment asks you to implement a priority queue by a binary heap. The input contains three commands as follows:

- *insert* x: insert the number x into the heap.
- *deleteMin*: delete the minimum key from the heap.
- $findMin$: find the minimum key in the heap.

Note that when $findMin$ is performed, you should store the returned value. After all the commands in the input file are executed, you should return two sequences. The first sequence is $findMin$ sequence which is composed of all the returned values when $findMin$ is performed. The second sequence is the array sequence (without the dummy node). For example, the array sequence of Figure 1(b) is A, B, E, C, D, F. Notably, if the heap is empty and $findMin$ is performed, append "-" to the $findMin$ sequence. Moreover, if the heap is empty and $deleteMin$ is preformed, just do nothing.

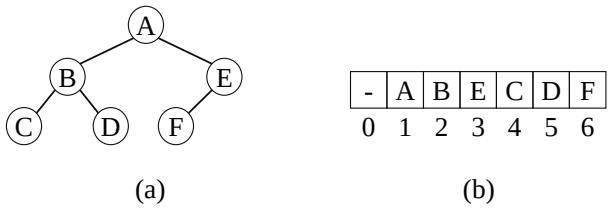

Figure 1: (a) A heap with 6 keys. (b) The array storing the heap in (a).

2 Input/Output Specification

2.1 Input Format

The input consists of several commands separated by a newline. The commands include 1) *insert* x, 2) *deleteMin*, and 3) *findMin*. Figure 2(a) shows an input example.

2.2 Output Format

The output should contains the two sequences separated by a newline. The first sequence is the $findMin$ sequence and the second one is the array sequence. For example, if we execute the commands in Figure 2 sequentially, then the $findMin$ sequence is $1, 2$ and the array sequence is $2, 3, 5, 4$. Figure 2(b) demonstrates the output example of Figure 2(a).

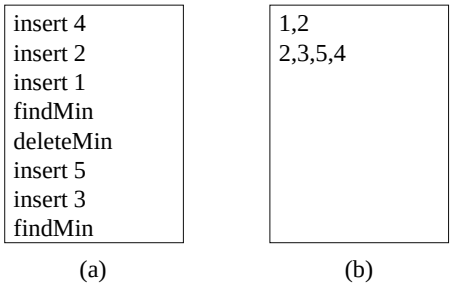

Figure 2: (a) An input example of 8 commands. (b) The output example of (a).

3 Command-line Arguments

Please follow the command-line arguments as below.
Usage:

python3 programming_hw4.py [input file] [output file]

Example:

python3 programming_hw4.py input.txt output.txt

4 Evaluation

To check the correctness of your implementation, 10 public cases are provided for you. In the final evaluation, we will use the 10 public cases and 10 hidden cases (5% for each case) to evaluate your code. The sizes of hidden cases are similar to those of the public cases. The runtime limit for each case is 10 seconds.

5 Submission

- Please compress your programming_hw4.py (.zip or .tar) and upload it to CEIBA.
- Please submit your code before **2 p.m. on December 17, 2018**.