

Sorting With Style

Due date: May 5, 2015 at 23:59 CET

This is an individual assignment. You may discuss it with others, but your formulations, your code, and all the required material must be written on your own.

Problem 1 — Hiccup Sort (30 points)

Implement an algorithm that sorts an array A of integers *in-place* such that after, the sorting, the elements of A that are in an *odd* positions (1, 3, 5, ...) appear in decreasing order, those in *even* positions (0, 2, 4, ...) appear in increasing order, and the values in even positions are greater than those in odd positions.

Formally, the resulting array *after sorting* satisfies the following requirements:

$$\forall i | i \text{ even} \Rightarrow A[i] \leq A[i+2]$$

$$\forall i | i \text{ odd} \Rightarrow A[i] \geq A[i+2]$$

$$\forall i, j | i \text{ even}, j \text{ odd} \Rightarrow A[i] \leq A[j]$$

Note: consider the array starting from 0, following the Python convention.

Example

The array

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 5 | 6 | 7 | 3 | 4 | 5 | 2 | 6 | 9 |
|---|---|---|---|---|---|---|---|---|---|

would be sorted as:

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 9 | 2 | 7 | 3 | 6 | 4 | 6 | 5 | 5 |
|---|---|---|---|---|---|---|---|---|---|

Problem 2 — Priority Revised (70 points)

The post office has decided to implement a new strategy for serving its customers: regardless of the arrival order, it wants to serve the youngest and the eldest person in the queue. To put this strategy into action, the analysts of the post office decided to adopt a *min/max priority queue*, and hired you to implement it.

Implement a min/max priority queue. The queue must support the following operations:

- **add(x):** adds x to the queue.
- **getmin():** extracts and returns the minimum value in the queue.
- **getmax():** extracts and returns the maximum value in the queue.

Implement the **add**, **getmin** and **getmax** functions, such that the complexity for each operation is $O(\log n)$.

Example

Starting from an empty queue, below is a list of operations and the corresponding results:

| # | Operation | Output | Status of the Queue |
|----|------------------|--------|---------------------|
| 1 | add (4) | | {4} |
| 2 | add (1) | | {1, 4} |
| 3 | getmin () | 1 | {4} |
| 4 | getmax () | 4 | \emptyset |
| 5 | add (7) | | {7} |
| 6 | add (9) | | {7, 9} |
| 7 | add (2) | | {2, 7, 9} |
| 8 | add (4) | | {2, 4, 7, 9} |
| 9 | getmin () | 2 | {4, 7, 9} |
| 10 | add (7) | | {4, 7, 7, 9} |
| 11 | add (5) | | {4, 5, 7, 7, 9} |
| 12 | getmax () | 9 | {4, 5, 7, 7} |
| 13 | getmax () | 7 | {4, 5, 7} |
| 14 | getmax () | 7 | {4, 5} |

Implementation and Submission

Write your program in Python, version 3. You may not use any library function other than to read data from the input. You may not use any pre-defined data structure other than basic arrays or lists.

You must make sure that your program can run directly from the terminal.

In the archive containing this document you can find the following files files:

- **minmaxqueue.py**: a stub of the functions **add**, **getmin**, and **getmax**;
- **main.py**: a python file containing the commands to test your solution.

Use these files to implement your solution. You can add all the functions you need, but you cannot change the name of the **add**, **getmin**, and **getmax** functions.

You must submit a single **tar.gz** or **zip** archive containing only:

1. The file **minmaxqueue.py**, containing the functions you implemented (do not include the **main.py** file).
2. A **README.pdf** or **README.txt**

Do not include any other file or directory. You may use the IDE of your choice, but do not include any project files and folders.

The readme file should contain a brief description of your implementations, possibly a list of limitations or errors you are aware of but that you were not able to fix, and clear references to any and all external material you might have used, including discussions with or help from other students.

Name your archive file following this format: **assignment03g-lastname-firstname.tar.gz** (or .zip). Submit the tar.gz or zip archive through the *iCorsi* platform.