

## Lab Programming Assignment #4 – Extra Credit

**Due:** Friday, May 1, by 11:59 pm PDT

**Points possible:** 20

### Submission Requirements:

- You can work in teams up to 3. If working in teams, only one person should submit the required file. However, all team member names need to be listed within the Java file, as a comment at the top.
  - If working in a team, adhere to social distancing and collaborate remotely over the phone and internet.
- You should submit just one file, and that file's name should be:
  - **MaxPriorityQueue.java**
- Submitting a file with any other name will get you points deducted. Do not zip the file either.
- Because the Java file name should be what's listed above, the top level class name within it should also be named **MaxPriorityQueue**
- Within the java file, include your name (at the top as a comment). If working in a team, the full names of all team members must also be within the Java file (at the top as a comment).
- Submit the Java file to BeachBoard. No email submissions will be accepted.
- The project must be in the Java programming language.
- I should be able to run your program via the command line.

### Program Requirements:

1. Implement the Max Priority Queue operations as mentioned in Section 6.5 of the textbook. Your program should use an array.
2. The following operations should be implemented:
  - INSERT( $S, x$ )** inserts the element  $x$  into the set  $S$ , which is equivalent to the operation  $S = S \cup \{x\}$ .
  - MAXIMUM( $S$ )** returns the element of  $S$  with the largest key.
  - EXTRACT-MAX( $S$ )** removes and returns the element of  $S$  with the largest key.
  - INCREASE-KEY( $S, x, k$ )** increases the value of element  $x$ 's key to the new value  $k$ , which is assumed to be at least as large as  $x$ 's current key value.
3. Your program must be able to support up to 10 integers. It would take in those integers (separated by spaces), build a max heap, then based on the operation chosen above, will perform the operation on the max heap and output per the example provided below.
4. For the program inputs, you can assume the index starts at 0.
5. Use the algorithms from the book. Although it may need some modifications as the indices in Java start at 0 instead of 1.
6. The inputs to your program must be done via the command line.
7. Your algorithm should also be able to support just one element as your input, as well as duplicates.

8. The Increase Key operation should output an error if the new key is less than the old key.
9. For the Increase Key operation, when inputting the node number to increase, assume the index starts at 1. See the example section **highlighted in yellow** below to see it in action.
10. Extract Max should output an error if there are no integers to extract (in case of empty heap).
11. Executing your program should be done via the command line. If I cannot run your program through the command line, you will get a lot of points deducted, up to and including receiving zero credit.
12. Example (inputs are bolded)

```
>java MaxPriorityQueue
```

Input an array of integers separated by spaces: **5 10 8 13 2 17 12**

Outputted Max Heap: 17 13 12 10 2 8 5

Choose from the following options:

1. Insert
2. Maximum
3. Extract-Max
4. Increase-Key
5. Exit

**1**

Input the integer you want inserted: **14**

Outputted Max Heap: 17 14 12 13 2 8 5 10

Choose from the following options:

1. Insert
2. Maximum
3. Extract-Max
4. Increase-Key
5. Exit

**2**

Maximum value is: **17**

Choose from the following options:

1. Insert
2. Maximum
3. Extract-Max
4. Increase-Key
5. Exit

**3**

Maximum value is: 17

Outputted Max Heap: 14 13 12 10 2 8 5

Choose from the following options:

1. Insert
2. Maximum
3. Extract-Max
4. Increase-Key
5. Exit

**4**

Input the index of the node you want to increase: **3**

Input the new value: **6**

ERROR: new key is smaller than current key

Choose from the following options:

1. Insert
2. Maximum
3. Extract-Max
4. Increase-Key
5. Exit

**4**

Input the index of the node you want to increase: **3**

Input the new value: **18**

Outputted Max Heap: 18 13 14 10 2 8 5

Choose from the following options:

1. Insert
2. Maximum
3. Extract-Max
4. Increase-Key
5. Exit

**5**

Comment your code enough so that in case it doesn't work, I can see what you're trying to do and possibly give you partial credit.

**Failure to follow the above requirements may result in substantial point deduction, up to and including receiving zero credit. I recommend you read the instructions more than once to ensure your program meets the requirements.**

**Grading Guidelines:**

- Does the program meet the requested requirements/criteria?
- Are the submission instructions followed?
- Is your code properly commented?
- Does your code compile?
- Does your code pass my test cases?