

## Lab Programming Assignment #3

**Due:** Sunday, November 8, 2020 by 11:59 pm PDT

**Points Possible:** 20 + 10 Extra Credit

### Submission Requirements:

- You can work in teams of up to 3. If working in teams, only one person should submit the required file on BeachBoard.
  - 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:
  - **BuildMaxHeap.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 **BuildMaxHeap**.
- 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).
- In addition, all team member names need to be listed on the submission in BeachBoard, 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 test with JDK version 14.0.2 and the expectation is that your code works with that version. Otherwise, you will get points deducted, up to and including zero credit
- I should be able to both pass in arguments and run your program via the command line.
- Ensure the Java keyword "package" doesn't exist in your java file before you submit your code. That will prevent me from properly testing your code, so remove it before submitting your code.

### Program Requirements:

- Implement Build Max Heap using an array. It must output the following:
  1. The output max heap sequence.
  2. Height of the max heap.
- Note that your output will not be a sorted array. It will be a max heap, with the highest value in A[0].
- It must be able to support up to 200 integers.
- Since this is Build Max Heap, you must implement Max-Heapify.
- Use the algorithms from the book. Although it may need some modifications when you code your program as the indices in Java start at 0 instead of 1.
- Your algorithm should also be able to support just one element as your input, as well as duplicates.
- Executing your program should be done via the command line (in the exact format as the below examples). If I cannot both run your program and pass in my test cases through the command line, you will get a lot of points deducted, up to and including receiving zero credit. No exceptions.

After executing your program via the command line, I should see the following (the below input and output array values are just examples):

```
>java BuildMaxHeap 4 5 1 7 3 3 8
Max Heap Array:
8 7 4 5 3 3 1
```

Height: 2

```
>java BuildMaxHeap 2
Max Heap Array:
2
```

Height: 0

```
>java BuildMaxHeap 2 2
Max Heap Array:
2 2
```

Height: 1

```
>java BuildMaxHeap 2 3 2
Max Heap Array:
3 2 2
```

Height: 1

```
>java BuildMaxHeap 4 3 1 2
Max Heap Array:
4 3 1 2
```

Height: 2

```
>java BuildMaxHeap 4 1 3 2
Max Heap Array:
4 2 3 1
```

Height: 2

**Extra Credit (Up to 10 Points Extra Credit):****Extra Credit Submission Requirements:**

- You can work in teams of up to three, but it would need to be the same team that worked on the BuildMaxHeap.java program, or you can do it individually.
  - 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:
  - **BuildMaxHeapWeather.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 **BuildMaxHeapWeather**.
- 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).
- In addition, all team member names need to be listed on the submission in BeachBoard, as a comment.
- Submit the Java file to BeachBoard. No email submissions will be accepted.
  - A separate DropBox will be created on BeachBoard for the extra credit.
- The project must be in the Java programming language.
  - I test with JDK version 14.0.2 and the expectation is that your code works with that version. Otherwise, you will get points deducted, up to and including zero credit
- I should be able to both pass in arguments and run your program via the command line.
- Ensure the Java keyword "package" doesn't exist in your java file before you submit your code. That will prevent me from properly testing your code, so remove it before submitting your code.

**Extra Credit Program Requirements:**

For up to 10 extra credit points (counted towards the Lab category), modify your BuildMaxHeap program per the following requirements:

1. Call the OpenWeather API ([api.openweathermap.org/data/2.5/weather](https://api.openweathermap.org/data/2.5/weather)) to get the current weather (in Fahrenheit) of a city inputted into the program via the command line.
  2. Based on the weather temperature returned, take the floor of that and convert it to an integer. Display that integer (see example below for the output display format).
  3. Randomly generate integers to sort and display that (see example below for the output display format). The integer returned from step 2 will tell you how many random integers there should be to sort. So for example, if step 2 returns 89, generate 89 random integers.
    - a. The random integer range should be between 0 and 200. This includes 0 and 200.
  4. Build a max heap on those randomly generated integers and display that (see example below for the output display format).
- Before doing the above steps, you would need to create a free OpenWeather API account: [https://home.openweathermap.org/users/sign\\_up](https://home.openweathermap.org/users/sign_up). This will then provide you with an appid so that you can call the OpenWeather API.
  - See examples below on how the program will be executed from the command line.

Example:

## Example:

The current weather of Long Beach is 81.46. So there should be 81 randomly generated integers (between 0 and 200).

```
>java BuildMaxHeapWeather "long beach,ca,usa"
81 randomly generated integers:
80 16 159 196 116 176 186 37 107 67 194 144 178 110 149 122 141 177 108 120 162 55
110 154 104 51 118 162 59 132 14 148 25 117 23 109 27 60 36 10 92 128 111 198 57 129
199 23 49 16 16 185 144 27 129 141 178 90 177 124 11 86 132 179 200 90 103 73 108 28
55 119 21 116 68 178 134 42 8 178 56
```

## Max Heap Array:

```
200 199 186 196 198 185 178 179 178 178 194 154 178 177 149 148 141 177 134 120 162
116 129 144 104 176 129 162 159 132 132 122 103 117 55 119 116 108 42 67 92 128 111
55 57 80 110 23 49 16 16 51 144 27 118 141 110 90 59 124 11 86 14 16 37 90 25 73 108
28 23 109 21 27 68 60 107 36 8 10 56
```

Height: 6

The below applies to both the BuildMaxHeap.java assignment and the extra credit BuildMaxHeapWeather.java assignment:

**Failure to follow the above requirements will 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.**

**Be aware that I use plagiarism detection software on the submissions. If plagiarism is detected, at a minimum, you will receive zero credit for this assignment.**

**Grading Guidelines:**

- Does the program meet the requested requirements/criteria?
- Are the submission instructions followed?
- Does your code compile?
- Does your code pass my test cases? Note that I do not share my test cases.
- Am I able to both pass in inputs and execute your program via the command line?