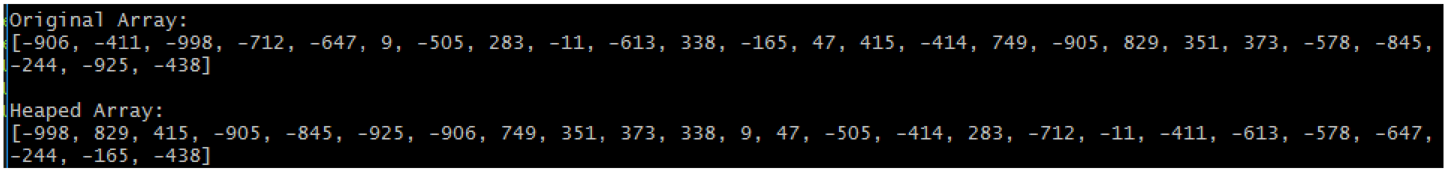
Jay Offerdahl
EECS 560
Lab 8 Report

Lab 8 Report

Testing & Verification:

BuildHeap: Before I had a working solution, I ran into several problems which included levels being organized properly in the sense that max levels had higher values than their children, but not grandchildren. This only meant that trickleDown wasn't working quite right, but after some debugging, I found I was comparing a few wrong values in both trickleDownMin and trickleDownMax. I then tested my buildHeap on 25 integers, with the output below.

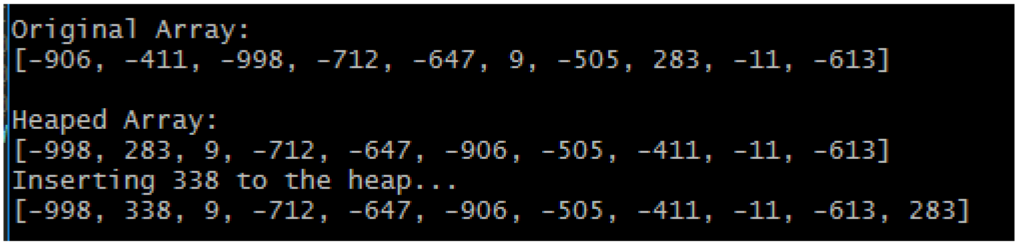


```
Original Array:
[-906, -411, -998, -712, -647, 9, -505, 283, -11, -613, 338, -165, 47, 415, -414, 749, -905, 829, 351, 373, -578, -845,
-244, -925, -438]

Heaped Array:
[-998, 829, 415, -905, -845, -925, -906, 749, 351, 373, 338, 9, 47, -505, -414, 283, -712, -11, -411, -613, -578, -647,
-244, -165, -438]
```

After drawing out the heap, I can confirm that it is in fact a valid min-max heap. We can also see at this point that the lowest value is in the root (-998), and the maximum value is in one of the children of the root (left child, in this case). With this working properly, I felt comfortable moving on to the three other methods required for this lab.

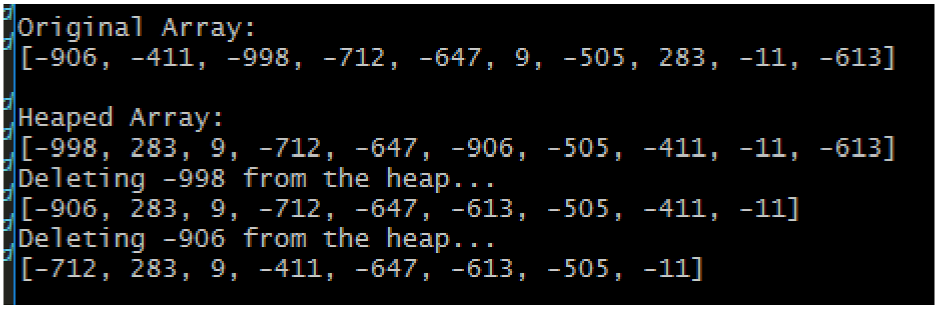
Insert: This function implemented the "bubbleUp" operation, so it took a little bit longer than the two delete methods. In either case, I used a heap of size 10 and inserted a randomly generated value, which was reported to the user. This output makes sense because 338, the newly inserted value should be a child of the root since it's the largest value. Also, it moves 283 to the grandchild of the left child of the parent because that's the next max level on that subtree (also, it has to keep the heap structure).



```
Original Array:
[-906, -411, -998, -712, -647, 9, -505, 283, -11, -613]

Heaped Array:
[-998, 283, 9, -712, -647, -906, -505, -411, -11, -613]
Inserting 338 to the heap...
[-998, 338, 9, -712, -647, -906, -505, -411, -11, -613, 283]
```

DeleteMin: Using a heap of size 10 (for simplicity), I show that this is working correctly as the value in the root is removed, replaced with the last value on the heap, and then that value is trickleDowned. For example, I did this operation twice in the screenshot. (First removes -998, the lowest value, and then -906).

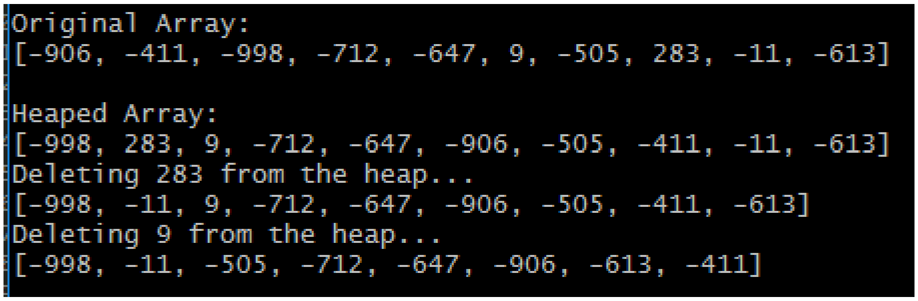


```
Original Array:
[-906, -411, -998, -712, -647, 9, -505, 283, -11, -613]

Heaped Array:
[-998, 283, 9, -712, -647, -906, -505, -411, -11, -613]
Deleting -998 from the heap...
[-906, 283, 9, -712, -647, -613, -505, -411, -11]
Deleting -906 from the heap...
[-712, 283, 9, -411, -647, -613, -505, -11]
```

DeleteMax: Again, I used a heap of size 10. Taking from the previous screen-grab, the maximum value in the original heap is 283, so that should be the first to be deleted. Next, 9 should be deleted because it's the next highest value. Here's what my program put out:

Jay Offerdahl
EECS 560
Lab 8 Report



```
Original Array:
[-906, -411, -998, -712, -647, 9, -505, 283, -11, -613]

Heaped Array:
[-998, 283, 9, -712, -647, -906, -505, -411, -11, -613]
Deleting 283 from the heap...
[-998, -11, 9, -712, -647, -906, -505, -411, -613]
Deleting 9 from the heap...
[-998, -11, -505, -712, -647, -906, -613, -411]
```

Here’s the output from running the requested commands. NOTE: This can also be viewed in “output.lab8.txt”

```
******************************************************************

          Jay Offerdahl - Lab 8

******************************************************************

Original Array:
[3, 79, 51, 25, 2, 49, 84, 64, 39, 89, 12, 40, 96, 98, 92, 68, 30, 29, 83, 50, 23, 82, 7, 78, 95, 56, 71, 46, 10, 37, 38, 57, 42, 1,
26, 97, 59, 90, 13, 67, 53, 18, 99, 34, 27, 36, 62, 24, 45, 8]

Heaped Array:
[1, 99, 98, 3, 2, 8, 10, 68, 97, 89, 82, 95, 96, 84, 92, 42, 25, 29, 13, 50, 18, 12, 7, 24, 40, 56, 71, 46, 51, 37, 38, 57, 64, 30,
26, 39, 59, 90, 83, 67, 53, 23, 79, 34, 27, 36, 62, 78, 45, 49]
Inserting 5 to the heap...
[1, 99, 98, 3, 2, 5, 10, 68, 97, 89, 82, 95, 96, 84, 92, 42, 25, 29, 13, 50, 18, 12, 7, 24, 8, 56, 71, 46, 51, 37, 38, 57, 64, 30,
26, 39, 59, 90, 83, 67, 53, 23, 79, 34, 27, 36, 62, 78, 45, 49, 40]
Inserting 87 to the heap...
[1, 99, 98, 3, 2, 5, 10, 68, 97, 89, 82, 95, 96, 84, 92, 42, 25, 29, 13, 50, 18, 12, 7, 24, 8, 56, 71, 46, 51, 37, 38, 57, 64, 30,
26, 39, 59, 90, 83, 67, 53, 23, 79, 34, 27, 36, 62, 78, 45, 49, 40, 87]
Deleting 99 from the heap...
[1, 97, 98, 3, 2, 5, 10, 68, 90, 89, 82, 95, 96, 84, 92, 42, 25, 29, 13, 50, 18, 12, 7, 24, 8, 56, 71, 46, 51, 37, 38, 57, 64, 30,
26, 39, 59, 87, 83, 67, 53, 23, 79, 34, 27, 36, 62, 78, 45, 49, 40]
Deleting 1 from the heap...
[2, 97, 98, 3, 7, 5, 10, 68, 90, 89, 82, 95, 96, 84, 92, 42, 25, 29, 13, 50, 18, 12, 36, 24, 8, 56, 71, 46, 51, 37, 38, 57, 64, 30,
26, 39, 59, 87, 83, 67, 53, 23, 79, 34, 27, 40, 62, 78, 45, 49]
Inserting 50 to the heap...
[2, 97, 98, 3, 7, 5, 10, 68, 90, 89, 82, 95, 96, 84, 92, 42, 25, 29, 13, 50, 18, 12, 36, 24, 8, 56, 71, 46, 51, 37, 38, 57, 64, 30,
26, 39, 59, 87, 83, 67, 53, 23, 79, 34, 27, 40, 62, 78, 45, 49, 50]
Deleting 98 from the heap...
[2, 97, 96, 3, 7, 5, 10, 68, 90, 89, 82, 95, 71, 84, 92, 42, 25, 29, 13, 50, 18, 12, 36, 24, 8, 56, 50, 46, 51, 37, 38, 57, 64, 30,
26, 39, 59, 87, 83, 67, 53, 23, 79, 34, 27, 40, 62, 78, 45, 49]
Inserting 98 to the heap...
[2, 97, 98, 3, 7, 5, 10, 68, 90, 89, 82, 96, 71, 84, 92, 42, 25, 29, 13, 50, 18, 12, 36, 24, 8, 56, 50, 46, 51, 37, 38, 57, 64, 30,
26, 39, 59, 87, 83, 67, 53, 23, 79, 34, 27, 40, 62, 78, 45, 49, 95]
Inserting 2 to the heap...
[2, 97, 98, 3, 7, 2, 10, 68, 90, 89, 82, 96, 71, 84, 92, 42, 25, 29, 13, 50, 18, 12, 36, 24, 8, 5, 50, 46, 51, 37, 38, 57, 64, 30,
26, 39, 59, 87, 83, 67, 53, 23, 79, 34, 27, 40, 62, 78, 45, 49, 95, 56]
Inserting 1 to the heap...
```

Jay Offerdahl
EECS 560
Lab 8 Report

```
[1, 97, 98, 3, 7, 2, 10, 68, 90, 89, 82, 96, 71, 84, 92, 42, 25, 29, 13, 50, 18, 12, 36, 24, 8, 2, 50, 46, 51, 37, 38, 57, 64, 30,
26, 39, 59, 87, 83, 67, 53, 23, 79, 34, 27, 40, 62, 78, 45, 49, 95, 56, 5]
Inserting 100 to the heap...
[1, 97, 100, 3, 7, 2, 10, 68, 90, 89, 82, 96, 98, 84, 92, 42, 25, 29, 13, 50, 18, 12, 36, 24, 8, 2, 50, 46, 51, 37, 38, 57, 64, 30,
26, 39, 59, 87, 83, 67, 53, 23, 79, 34, 27, 40, 62, 78, 45, 49, 95, 56, 5, 71]
Inserting 99 to the heap...
[1, 97, 100, 3, 7, 2, 10, 68, 90, 89, 82, 96, 99, 84, 92, 42, 25, 29, 13, 50, 18, 12, 36, 24, 8, 2, 50, 46, 51, 37, 38, 57, 64, 30,
26, 39, 59, 87, 83, 67, 53, 23, 79, 34, 27, 40, 62, 78, 45, 49, 95, 56, 5, 71, 98]
Deleting 1 from the heap...
[2, 97, 100, 3, 7, 2, 10, 68, 90, 89, 82, 96, 99, 84, 92, 42, 25, 29, 13, 50, 18, 12, 36, 24, 8, 5, 50, 46, 51, 37, 38, 57, 64, 30,
26, 39, 59, 87, 83, 67, 53, 23, 79, 34, 27, 40, 62, 78, 45, 49, 95, 56, 98, 71]
Deleting 100 from the heap...
[2, 97, 99, 3, 7, 2, 10, 68, 90, 89, 82, 96, 98, 84, 92, 42, 25, 29, 13, 50, 18, 12, 36, 24, 8, 5, 50, 46, 51, 37, 38, 57, 64, 30,
26, 39, 59, 87, 83, 67, 53, 23, 79, 34, 27, 40, 62, 78, 45, 49, 95, 56, 71]
Deleting 2 from the heap...
[2, 97, 99, 3, 7, 5, 10, 68, 90, 89, 82, 96, 98, 84, 92, 42, 25, 29, 13, 50, 18, 12, 36, 24, 8, 56, 50, 46, 51, 37, 38, 57, 64, 30,
26, 39, 59, 87, 83, 67, 53, 23, 79, 34, 27, 40, 62, 78, 45, 49, 95, 71]
Deleting 99 from the heap...
[2, 97, 98, 3, 7, 5, 10, 68, 90, 89, 82, 96, 71, 84, 92, 42, 25, 29, 13, 50, 18, 12, 36, 24, 8, 56, 50, 46, 51, 37, 38, 57, 64, 30,
26, 39, 59, 87, 83, 67, 53, 23, 79, 34, 27, 40, 62, 78, 45, 49, 95]
Deleting 98 from the heap...
[2, 97, 96, 3, 7, 5, 10, 68, 90, 89, 82, 95, 71, 84, 92, 42, 25, 29, 13, 50, 18, 12, 36, 24, 8, 56, 50, 46, 51, 37, 38, 57, 64, 30,
26, 39, 59, 87, 83, 67, 53, 23, 79, 34, 27, 40, 62, 78, 45, 49]
Deleting 97 from the heap...
[2, 90, 96, 3, 7, 5, 10, 68, 87, 89, 82, 95, 71, 84, 92, 42, 25, 29, 13, 50, 18, 12, 36, 24, 8, 56, 50, 46, 51, 37, 38, 57, 64, 30,
26, 39, 59, 49, 83, 67, 53, 23, 79, 34, 27, 40, 62, 78, 45]
```