CSCI 2002 Module 6 Assignment

This assignment will be submitted via D2L. There will be a document on D2L that gives details on submitting written assignments via D2L.

For this assignment, you will be adding and removing items from a max heap of integers in 6 phases and must draw the resulting heap after each phase as both a binary tree and the corresponding array. You may show the result after each individual addition or show the result after each phase. However, showing all the intermediate results will make it easier to provide partial credit if there are any mistakes.

Don't forget that the heap is reconstructed after every addition and deletion. You can't add or remove several items and then adjust the heap.

Please make sure that after each phase your heap is valid. It will be VERY difficult to allow partial credit for one phase if the starting heap is invalid.

Each phase starts with the heap resulting from the previous phase.

Phase 1: Add 10 items to an empty max heap.

Starting with an empty max heap, add these numbers:

92 48 94 37 32 76 14 84 50 79

Draw the resulting heap, along with the order of the numbers in the array that represents the heap.

Phase 2: Remove 4 items from the heap.

Starting with the heap resulting from phase 1, remove the 4 largest items from the heap. Indicate the numbers that were removed, and in what order. Draw the resulting heap, along with the order of the numbers in the array that represents the heap.

Phase 3: Add 3 items to the heap.

Starting with the heap resulting from phase 1, add these numbers:

23 49 29

Draw the resulting heap, along with the order of the numbers in the array that represents the heap.

Phase 4: Remove 4 items from the heap.

Starting with the heap resulting from phase 3, remove the 4 largest items from the heap. Indicate the numbers that were removed, and in what order. Draw the resulting heap, along with the order of the numbers in the array that represents the heap.

Phase 5: Add 3 items to the heap.

Starting with the heap resulting from phase 1, add these numbers:

```
74   31   40
```

Draw the resulting heap, along with the order of the numbers in the array that represents the heap.

Phase 6: Remove 2 items from the heap.

Starting with the heap resulting from phase 5, remove the 2 largest items from the heap. Indicate the numbers that were removed, and in what order. Draw the resulting heap, along with the order of the numbers in the array that represents the heap.

Example:

The following pages show the result, in the format required, of following the same steps as above with the following numbers for phases 1, 3, and 5.

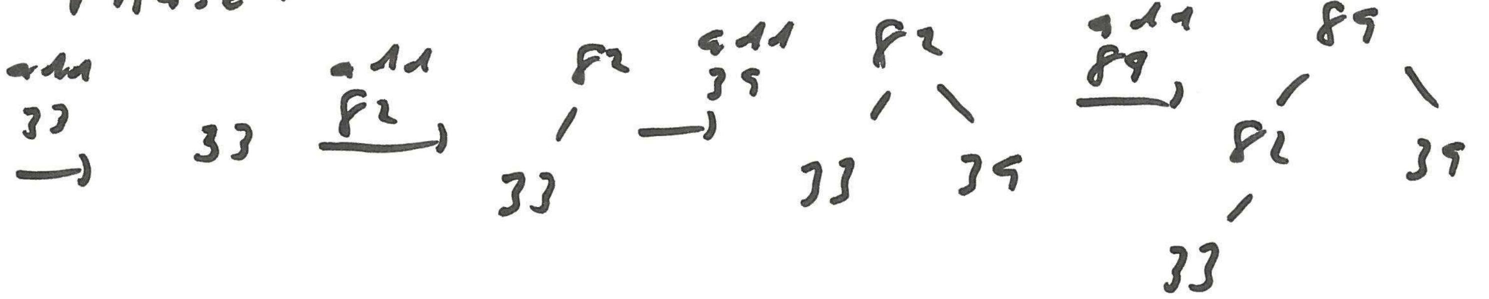
```
33   82   39   89   95   27   24   68   44   25

96   98   56

47   57   42
```

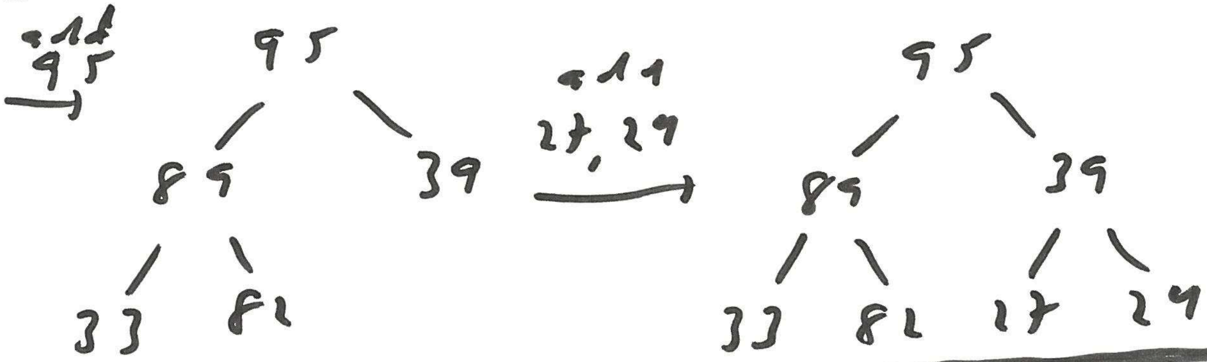
Note that I skipped a few steps in cases where adding a new item didn't require it to swim up. For example, in phase 1, 27 and 24 were both smaller than their parent, so I combined those steps. But adding 68 did require some heap adjustment, so I showed that step along with adding 44 and 25, which didn't require any swimming up.

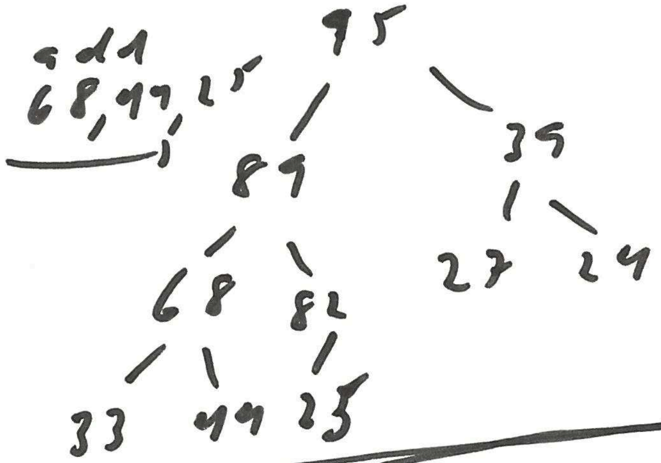
I also showed each operation above the arrow connecting it to the resulting drawing. I suggest you do something similar.

The reason I listed arrays at the end rather than interspersed with the other operations is that, well, I forgot about them until the end. I'd prefer that you put the array contents after each phase.

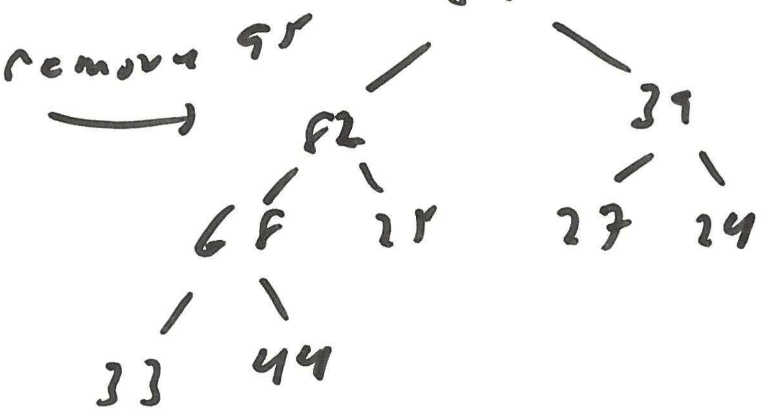
Phase 1

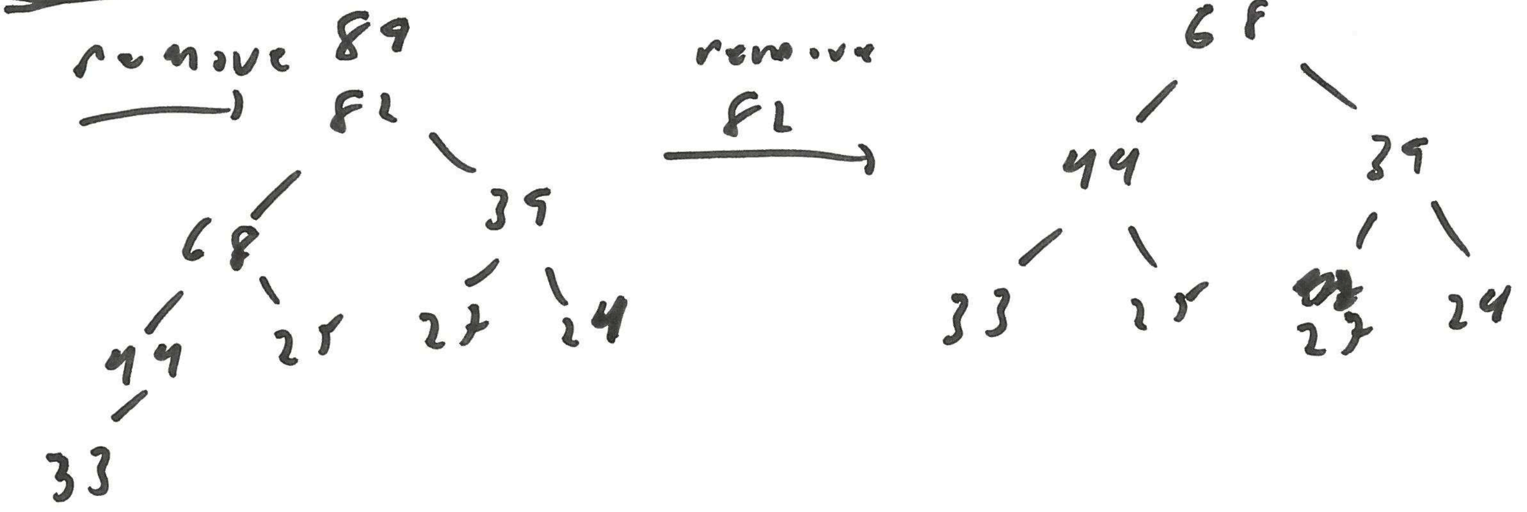






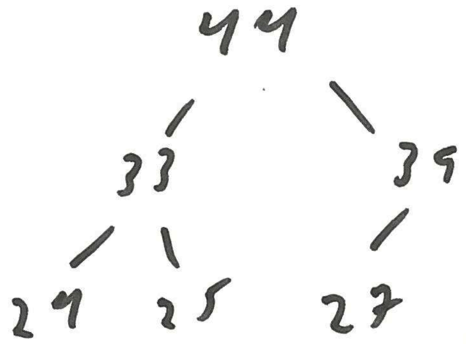
Phase 2





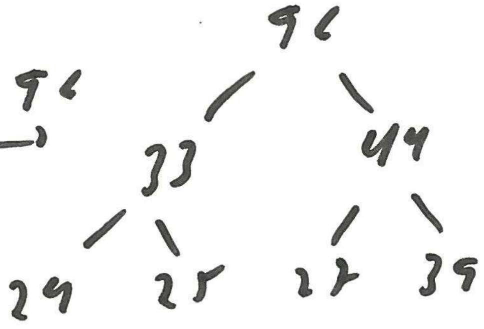
Phase 2

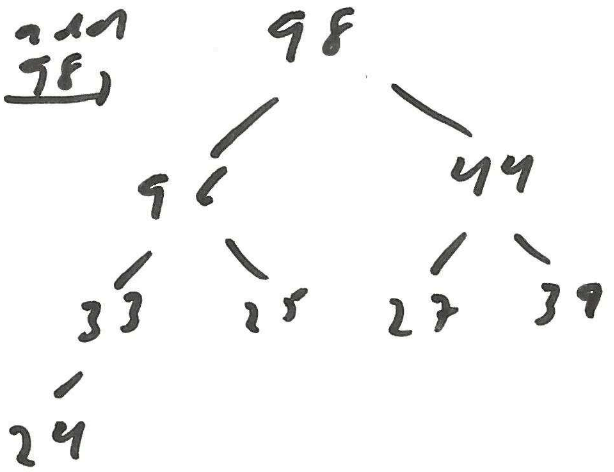
remove 68

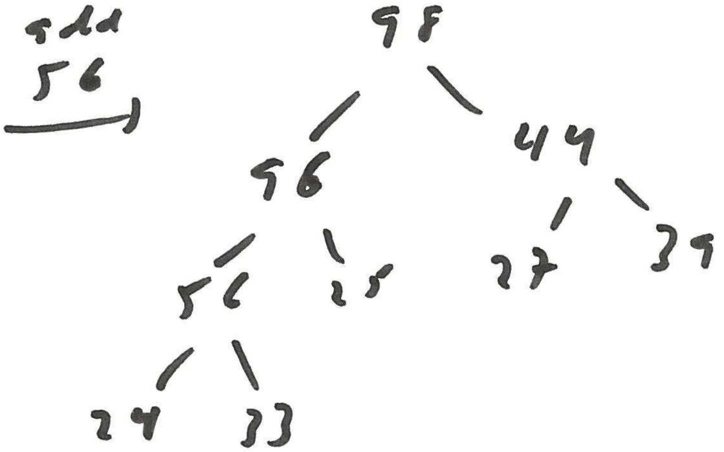


Phase 3

add 96

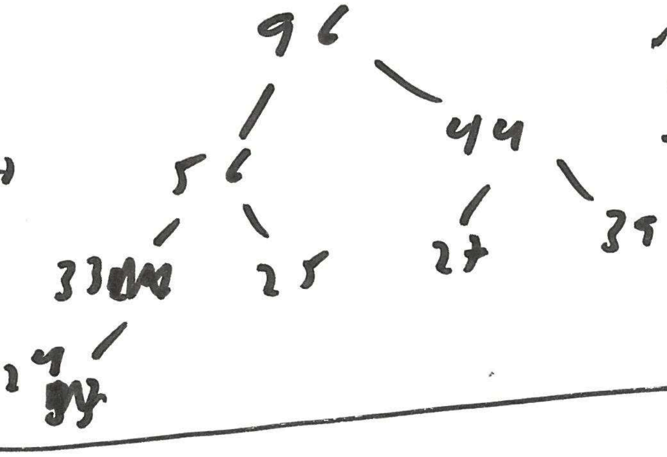




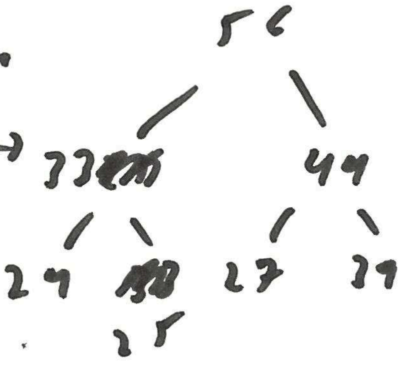


Phase 4

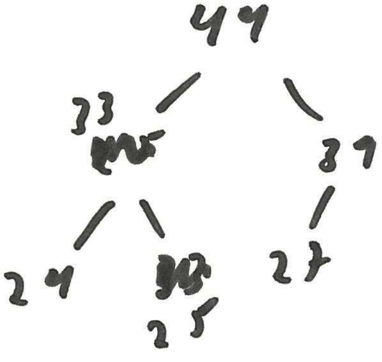
remove 98



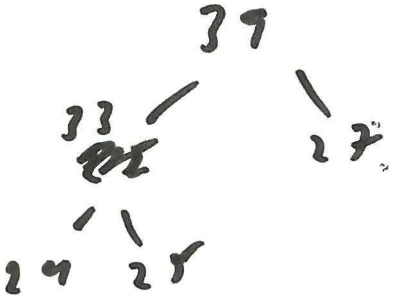
remove 96



remove 56

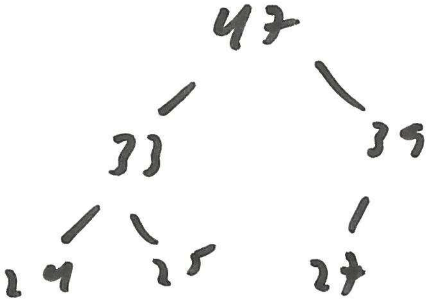


remove 44

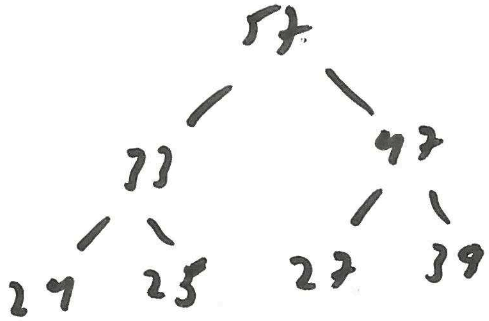


Phase 5

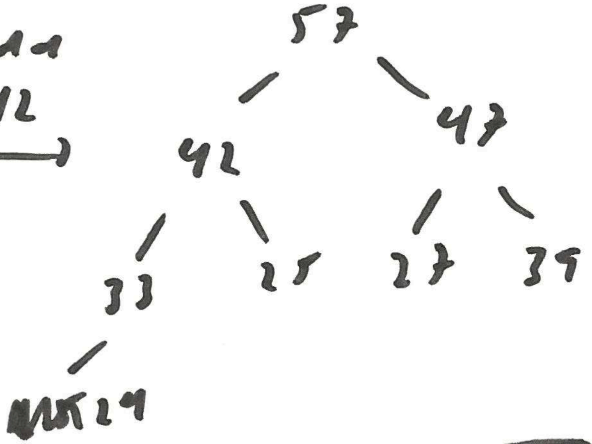
add 47 →



add 57 →

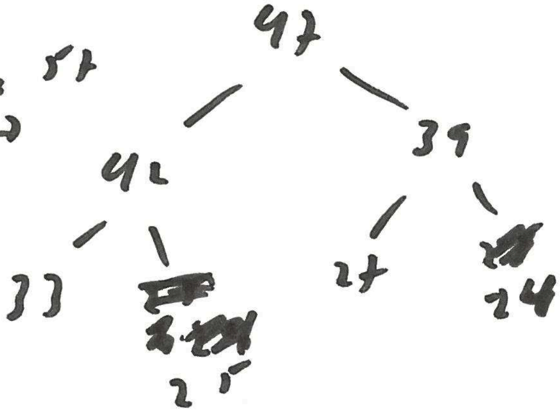


add 42 →

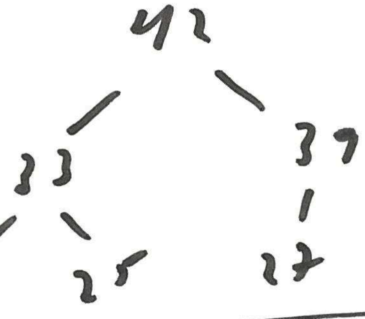


Phase 6

remove 57 →



remove 47 →



Arrays

Phase 1: 55 85 39 61 82 27 24 33 44 25

Phase 2: 44 33 35 24 25 27

Phase 3: 98 96 44 56 25 27 39 24 33

Phase 4: 39 33 27 24 25

Phase 5: 57 42 47 33 25 27 39 24

Phase 6: 42 33 39 24 25 27