

Computer Science I – Spring 2012
Lab: Heaps (Solutions)

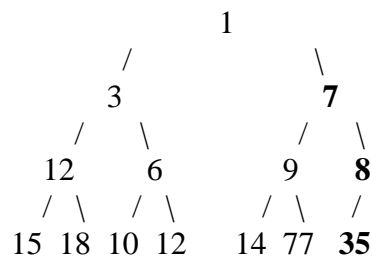
1) In an array-based implementation of a Heap, the left-child of the left-child of the node at index i , if it exists, can be found at what array location?

It will be found at array position $4i$

2) In an array-based implementation of a Heap, the right-child of the right-child of the node at index i , if it exists, can be found at what array location?

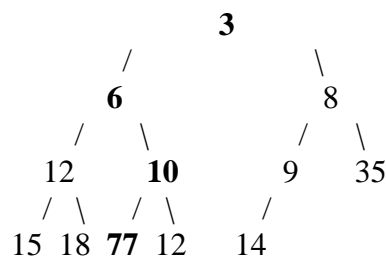
It will be found at array position $4i + 3$

3) Show the result of inserting the item 7 into the heap shown below:



(Changes are marked in bold.)

4) Show the result of removing the minimum element from the original heap in question #2 (without 7) from above.

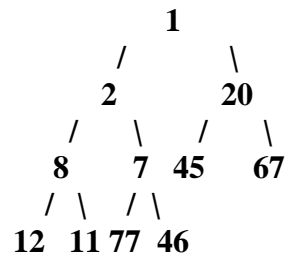
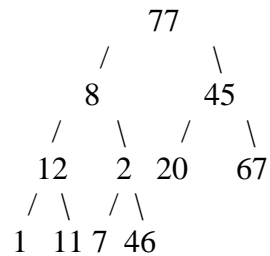


(Changes are marked in bold.)

5) Show the array representation of the original heap from question #2.

| | | | | | | | | | | | | | | |
|-------|---|----------|----------|----------|-----------|----------|----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| Index | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
| value | | 1 | 3 | 8 | 12 | 6 | 9 | 35 | 15 | 18 | 10 | 12 | 14 | 77 |

6) Run the whole Make Heap function on the following random values:



7) Explain each step shown in the code below, for the percolateDown function:

```

void percolateDown(struct heapStruct *h, int index) {
    int min;
    if ((2*index+1) <= h->size) {
        min = minimum(h->heaparray[2*index], 2*index, h->heaparray[2*index+1], 2*index+1);
        if (h->heaparray[index] > h->heaparray[min]) {
            swap(h, index, min);
            percolateDown(h, min);
        }
    }
    else if (h->size == 2*index) {
        if (h->heaparray[index] > h->heaparray[2*index])
            swap(h, index, 2*index);
    }
}

```

(Note: Please reference heap.c *without looking at this function*, if necessary.)

Detailed explanations are given in the comments of heap.c, which is on the website under the sample programs link.