

# Problem Set 10

## Heap

---

1. Given the following sequence of integers:

12, 19, 10, 4, 23, 7, 45, 8, 15

1. Build a heap by inserting the above set, one integer at a time, in the given sequence. Show the heap after every insertion. How many comparisons in all did it take to build the heap?
  2. Perform successive *delete* operations on the heap constructed in the previous step, until the heap is empty. Show the heap after every deletion. How many comparisons in all did it take to perform these deletions?
- 

2. Suppose we have a (**max**) heap that stores integers. (By contrast, in a "min" heap the key at any node is *less than or equal to* the key at its children, so the *smallest* valued key is at the top of the heap.) Then, given an integer  $k$ , we would like to print all the values in this heap that are *greater than*  $k$ . Implement the following method to do this.

```
public void printGreater(int[] H, int n, int k) {  
    /* your code here */  
}
```

$H$  is the array storage for the max heap, and  $n$  is the number of entries in the heap.

*Note: The challenge is to do this efficiently. Use the heap order to reduce the number of entries of the heap to be examined.*

---

3. Consider a max heap that only supports the operations **insert**, **deleteMax**, **size**, and **isEmpty**. A client of the heap wants to update the priority of an entry in the heap. Since there is no search operation, the only way to accomplish the update is this:

- Perform successive **deleteMax** operations until the entry is extracted
- Update the entry's priority
- **Insert** the entry, as well as all the other deleted entries back into the heap

What would be the worst case running time (big  $O$ ) of this update process on a heap with  $n$  entries?

---

4. Suppose you are given two heaps, stored in arrays. Write a method to merge them into a single heap, and return this heap. The original heaps are not modified:

```
public static <T extends Comparable<T>> T[] merge(T[] heap1, T[] heap2) {  
    // COMPLETE THIS METHOD  
}
```

---