Week 5 Section Handout

This week’s section handout has practice with Big-O, priority queues and heaps.

1. Big Oh

Give a tight bound of the nearest runtime complexity class for each of the following code fragments in Big-Oh notation, in terms of the variable N.

// a)                           answer:
int sum = 0;
for (int i = 1; i <= N + 2; i++) {
    sum++;
}
for (int j = 1; j <= N * 2; j++) {
    sum += 5;
}
cout << sum << endl;

// b)                           answer:
int sum = 0;
for (int i = 1; i <= N - 5; i++) {
    for (int j = 1; j <= N - 5; j += 2) {
        sum++;
    }
}
cout << sum << endl;

// c)                           answer:
int sum = N;
for (int i = 0; i < 1000000; i++) {
    for (int j = 1; j <= i; j++) {
        sum += N;
    }
    for (int j = 1; j <= i; j++) {
        sum += N;
    }
    for (int j = 1; j <= i; j++) {
        sum += N;
    }
}
cout << sum << endl;

Thanks to CS106B and X instructors and TAs for contributing problems on this handout.
2. Min-Heap
We have implemented the Priority Queue ADT using a binary min-heap. Draw a diagram of the heap’s tree structure that results from inserting the following priority values in the order given: 25, 37, 28, 12, 30, 3

Diagram after inserting 25: 25  
We did this one for you
Diagram after inserting 37:  
Diagram after inserting 28:  
Diagram after inserting 12:  
Diagram after inserting 30:  
Diagram after inserting 3:  

3. Max-Heap
You have a PriorityQueue class which treats higher (larger integer value) priority elements as frontmost. The internal implementation of the class is a binary max-heap stored in an unfilled array. The initial allocation of the array is for capacity 5 and the array capacity is doubled when asked to enqueue to a queue which is full. You are going to trace the operation of enqueuing and dequeuing elements from the priority queue. You can sketch as a tree if you prefer when working it out, but your final answer should be based on an unfilled array that skips use of index 0.

(a) Show the contents of the internal array after these elements are enqueued to an empty priority queue in this order. Each element has a string value and a priority in parenthesis.
Red(8), Blue(33), Green(29), Purple(42), Orange(20), Yellow(22), Indigo(10), Teal(21)

(b) Dequeue is called twice on the priority queue. Which two values are removed?

(c) Show the contents of the internal array after the above two elements have been dequeued

Thanks to CS106B and X instructors and TAs for contributing problems on this handout.