CSE 373 Section Handout #2

Asymptotic Analysis:

1. For each of the following, show that $f \in O(g)$. That is, you will need to find values for c and n_0 such that the definition of big-O holds true.

   a) $f(n) = 3n$ $\quad\quad$ $g = 5n^2$

   b) $f = 17$ $\quad\quad$ $g = 32n + 15n\log n$

   c) $f = 121n^3$ $\quad\quad$ $g = 11n^3$

2. For each of the following pseudocode, determine the asymptotic runtime in terms of n

```
void smiley (int n, int sum) {
      for (int i = 0; i < n * 100; ++i) {
            for (int j = n; j > 0; j--)
                  sum++;
            for (int k = 0; k < i; ++k)
                  sum++;
      }
}

void happy (int n, int sum) {
      int j = n;
      while (j > 2) {
            sum++;
            j = j / 2;
      }
}
```

3. Order these functions from fastest to slowest in terms of asymptotic runtime.

   a) $N(N^2 \log(N) + N)$
   b) N^2
   c) $10{,}0000\ N^3$
   d) $2^N + 3.14159$
   e) $N^{1/2} + N + 128$

Heaps and Priority Queues

For these problems, we will assume the data stored in our internal array representation starts at index 1.

4. Provide two valid binary heaps containing the keys **4, 1, 55, 10, 7, 8, 3, 2**. Show the array representation, as well as the binary tree representation.

5. Draw the resulting binary tree representation from performing the deleteMin operation twice on the following binary heap: **[2, 4, 3, 5, 8, 7, 6, 9]** (array representation provided)

6. Draw the resulting binary tree representation from inserting the following keys **2, 4, 10, 5, 8, 3, 7, 1** in that order into an initially empty binary heap

7. Implement a method *isHeap* to verify whether a given array is a valid binary heap. isHeap should return `true` if the array is a binary heap, `false` otherwise.

   For example, suppose a variable *array* contains the following sequence of values: [2, 4, 6, 10]
   `isHeap(array)` would return true.

8. Implement the *contains* method for the IntPriorityQueue class. The method should return true if and only if the given key is contained within the priority queue.

   What is the runtime of the method? How does this compare to that of other PriorityQueue operations, as discussed in class?

   Note: **heap** is the array representation of a binary heap.

```
public class IntPriorityQueue {
      private int[] heap;
      private int size;

      public boolean contains(int key) {
            // your code here









      }
}
```