Take Home Test 1
CSCI 3394 – Advanced Algorithms

This test consists of several questions that you are to answer and return to me by class time on Thursday. If possible, typed answers are preferred.

1. Given the following function, first fine the recurance relation for its order then find the simplified form of that using the three methods from CLR Ch. 4. For a hint, try to figure out what this is doing and if it looks like any other algorithms you know. (2 points)

```
int valAtPosInArray(int[] a,int pos,int start,int end) {
      int low=start+1,high=end;
      while(low<=high) {
            if(a[low]<a[0]) {
                  low++;
            } else {
                  int tmp=a[low];
                  a[low]=a[high];
                  a[high]=tmp;
                  high--;
            }
      }
      int tmp=a[high];
      a[high]=a[0];
      a[0]=tmp;
      if(high==pos) return a[high];
      if(high<pos) return valAtPosInArray(a,pos-high-1,high+1,end);
      return valAtPosInArray(a,pos,start,high);
}
```

2. Given the following recurance relation find the simplified form of that using the three methods from CLR Ch. 4. (2 points)

$T(n)=2T(n/4)+O(n^2)$

3. Assume the following code is part of a working balanced binary tree. Why would you not want to use this code to get the height of the nodes for balancing operations? (1 point)

```
int height(Node n) {
      if(n==null) return 0;
      return 1+Math.max(height(n.left),height(n.right));
}
```

4. What are sentinals as related to linked data structures? Why would you use them in linked lists? What about in balanced binary trees? (1 point)

5. An alternate method of doing an order statistic tree might be to store the index of each element in the node. Why would you not want to do this? (1 point)

6. There is a data structure that exists that allows you to have a sorted linked list where all operations are $\Theta(\log n)$ just like they would be for a tree. The idea is to have multiple lists that link into the main list where each list is roughly half the length of the one before it. Draw a picture of this data structure. Why does it have the asymptotic behavior given above. Write pseudocode for get, add, and remove. (3 points)