Q5.
(a) Suppose that a tree T is both a min-heap and a binary search tree, and all nodes contain distinct values. At most how many nodes can T have?
(b) Show how to implement a queue with two stacks. What is the complexity of you insert and delete operations.

Q6.
Consider the following:

```c
void P(int n) {
    for(int i=1;i<=n;i++)
        for(int j=1; j<=n; j++)
            cout << i*j << endl;
}
void Q(int n) {
    if (n > 0) {
        P(n); Q(n-1); cout << n << endl;
    }
}
```

Construct the time recurrence relation for the function `void Q(int n)`, solve it to find T(n) exactly then determine the upper bound (big-O). What is the space complexity of `Q`?

Q7. Programs A and B are analyzed and found to have worst-case running times no greater than $150N \log N$ and $N^2$, respectively. Answer the following questions if possible:
   a) Which program has the better guarantee on the running time, for large values of $N$ ($N>10,000$)?
   b) Which program has the better guarantee on the running time, for small values of $N$ ($N<100$)?
   c) Which program will run faster on average for $N = 1,000$?

Q8. Devise an algorithm to count the number of nodes in an m-ary tree (i.e. a tree where each node has at most m-childe-nodes. What is the time complexity of your solution? What if you know that the m-ary tree is a full m-ary tree, can the efficiency of the algorithm be improved? Explain why and provide big-O.

Possible node representation for an n-ary tree:

```c
struct node {
    int info;
    struct node* child[n];
}
```

Q9. Given an array on n integers devise a $\Theta(n \log n)$ algorithm that counts the number of repeated integer values in the array. For example, the array [2,0,12,6,5] would result in a count of 0; the array [1,5,24,6,5,24,5] would result in a count of 2.
Q10. Let S be a sorted array of n integers. Devise an O(n) algorithm to determine whether there exists two integers a and b in S, such that a + b = 0. Modify your algorithm to list all the pairs of values in S whose sum equals 0. What is the complexity of the new algorithm?

Q11. (full grade for a T(n)=O(log_2(n)) algorithm) Given a full binary tree and two nodes in the tree, you are asked to devise an algorithm to find the nearest common ancestor of the two nodes.

Q12. Given a list of n point in a plane (in the form of the x and y coordinates of each), you are asked to:
   a) Devise an O(n^3) algorithm to determine if any three of them are collinear (i.e., lie on the same line).
   b) Devise an O(n^2 log_2(n)) algorithm to determine if any three of them are collinear.
   c) Devise an algorithm to count the number of collinear points; what is its complexity?

Q13. Consider a MAX heap containing n integers. You are given a positive integer k < n and a number x. Devise an algorithm to determine whether the kth largest element of the heap is greater than x or not. Your algorithm must take O(k) time. You may use O(k) extra storage.