1. (2 pts) Which set has more elements: $O(n^2)$ or $\Theta(n^2)$?

2. (4 pts) Prove that $5000n^2 - 1000n$ is in $\Theta(n^2)$. Specify values for $n_0$, $c_1$, and $c_2$.

3. (6 pts) Determine the complexity of the following algorithm and express your answer in terms of big-O. Explain/show your work. NOTE: $j$ is multiplied by two each time through the $j$-loop.

```java
count = 0;
for (i=n/2; i<n; i++) {
  for (j=1; j<n; j*=2) {
    count++;
  }
}
```

4. (2 pts each) State (using $O$) the complexity of each of the following operations:
   (a) insert in an ordered array-based list
   (b) insert in an unordered, linked list-based list
   (c) minimum in an ordered array-based list
   (d) search in an ordered, linked list-based list
   (e) binary search tree delete (balanced tree)
   (f) binary search tree max (unbalanced tree)

5. (10 pts) Consider these growth functions: $n!$, $\log n$, $n \log n$, $2^n$, $10^{100}$, $n^3$, $n$.
   (a) Write them in order from slowest growing to fastest growing.
   (b) Give the name for each type of growth function represented.

6. (4 pts) What is returned by this method call: `fun(3,5)`? Show your work. REMEMBER: An integer divided by an integer results in an integer.

```java
public int fun(int x, int y) {
  if (x == 0)
    return y;
  else
    return fun(x - 1, x + y) + fun(x/2,y);
}
```

7. (4 pts) What binary search tree would be built if an initially empty tree had these elements inserted in the specified order: 56, 34, 23, 78, 32, 12, 33, 90, 80?
8. (4 pts) Suppose a tree has 100 elements. What is the height of the tree assuming the tree is as balanced as possible? What is the height of the tree assuming it is as unbalanced as possible?

9. Suppose a binary search tree is implemented using the following class:

```java
class TreeNode {
    String word;
    int count;
    TreeNode left, right;
};
```

Suppose a MyTree class has been built and is an implementation of a binary search tree that it has been used to track the number of times words in a document have been used. Assume the words are used as the ordering key. Also, assume MyTree has an attribute called root.

(a) (6 pts) Write a method to be added to the MyTree class called findMin that accepts no parameters and will find the alphabetically smallest word in the tree. The method should return a reference to the TreeNode containing the alphabetically smallest word and should return null if the tree is empty.

(b) (8 pts) Write a method to be added to the MyTree class called calcSum that that accepts no parameters and returns the sum of all the counts. It should return 0 if the tree is empty. You will probably want to write a helper method as part of your solution.

10. (6 pts) Briefly outline the steps (3 cases) to delete an element from a binary search tree.

11. (2 pts each) Use the diagram of the binary tree below to answer the following questions:

(a) Is this tree a binary search tree?
(b) Show order of visits using a preorder traversal.
(c) Show order of visits using an inorder traversal.
(d) Show order of visits using a postorder traversal.