The first exam will cover all material that we covered in class through February 24. This includes everything in the lecture notes that I post online up to and including the AVL tree rotations. It does not include the worst case analysis of AVL trees.

The list of topics is:

series summations, modular arithmetic, proof techniques, all concepts of C++ described in Chapter 1, rates of growth of functions and order notation, runtime analysis of programs, trees, including general and binary trees, binary search trees and the associated algorithms and properties, and AVL trees and theeir algorithms and properties.

The format of the exam includes true/false questions, short answer questions, questions in which you will write algorithms and code, and questions that ask you to analyse algorithm performance or carry out algorithms on examples. Some sample questions of various types are below.

1. Write a closed form expression for the number of times that the line "I'm here" will be displayed on the screen when the following code fragment has run.

```
for (int i = 0; i < N; i++)
    for (int j = 0; j < N; j++)
        for (int k = j; k < N; k++)
            cout << "I'm here.\n";
```

2. Write a post order traversal of a general tree.

3. Define the internal path length of a tree.

4. If a user defines a one-parameter constructor for a class `C` but no other constructors, what does this code create:

```
C objects[100];
```

5. Write a function in C++ that removes a node from a binary search tree, given a pointer to its root and the element to be deleted.

6. What is the worst case time to insert an element into a BST with n nodes.

7. True or False? If a function's running time is $O(n^3)$ it can never run in strictly linear time.

8. True or False? Let c be a constant > 1. If $f(n)$ is defined as c raised to the n^{th} power and$g(n)$ is defined as c raised to the $(n+1)^{st}$ power, then $f(n) = o(g(n))$.

9. Draw a binary search tree of height no greater than 3 containing the keys 1 2 3 4 5 6 7 8 9 10 11 12.

10. Write a closed form expression of the running time of the function below as a function of N:

```
int foo (int N)
{
    if (N <= 0)
        return 1;
    else
        return foo(N/2) + foo(N/2) + N;
}
```

11. Given an AVL tree in a diagram, insert a specific element into the tree and rotate as necessary to rebalance the tree. You can make up examples.

This list is just a sample!