1. Give formal definitions for $O$, Omega, Theta, $P$, and $NP$.

2. What is the “expected” Big-O for Quicksort? What is the worst case behavior? How can that happen?

3. What is the Big-O complexity for the following function?
   
   ```c
   int f(int n)
   {
       if (n <= 1) return 1;
       return 1 + f(n - 1) + f(n - 1);
   }
   ```

4. What is the Big-O complexity for the following function?
   
   ```c
   int f(int n)
   {
       int i, j, k;
       k = 0;
       for (i = 0; i < 10000*n; i = i + 1)
           for (j = 0; j < i; j = j + 1)
               k = k + i * j;
       return k;
   }
   ```

5. The Fibonacci numbers are defined by $F(0) = F(1) = 1$, and $F(n) = F(n - 1) + F(n - 2)$ for values of $n$ greater than 1. Sketch EFFICIENT recursive code to compute $F(n)$; you should use memoization/dynamic programming.

6. Describe the Boolean satisfiability problem; give a short definition of what this is. For a given satisfiability problem, what would you use as a “certificate”? 