1. (10) Let $f$ and $g$ be two functions from integers to integers. State the definition of "$f(n)$ is $O(g(n))$" and then prove that $(2n+3)^2$ is $O(n^2)$ by giving the constants $n_0$ and $c$ in the definition of $O$-notation.

2. (10) Show that the harmonic series $S(n) = \sum_{i=1}^{n} \frac{1}{i}$ is $\Theta(\log n)$.

3. (10) The Fibonacci numbers are defined as:

$$F_0 = 0,\ F_1 = 1,\quad F_n = F_{n-1} + F_{n-2} \text{ for } n > 1.$$

   Show that $F_n > 2^{n/2}$ for $n > 5$.

4. (10) Show that $\log(n!) = \Theta(n \log n)$.
   (Hint: To show an upper bound, compare $n!$ with $n^n$. To show a lower bound, compare it with $(n/2)^{n/2}$.)

5. (15) Suppose that a computer can only do addition (+) and arithmetic shift (<< or >>). Write C-like code to compute the following statements efficiently.

   (a) $y = 10x$.
   (b) $y = 15x$.
   (c) Suppose that the computer can also do subtraction (−), in addition to addition and shift. Show how to compute $y = 15x$ more efficiently.

6. (15) A method to solve a recurrence equation is to expand out the recurrence a few times, until a pattern emerges. For each of the following recurrence equation,

   (a) $T(n) = 3T(n/2) + n$,
   (b) $T(n) = T(n-1) + 1$.

   answer the following questions.

   (a) What is the general $k$-th term?
   (b) What value of $k$ should be plugged in to get the answer?
   (c) What is the solution to the recurrence equation?

7. (20) Let $a_i$ and $b_i$, $1 \le i \le n$, be integers. Design a linear time algorithm for computing $s = \sum_{i=1}^{n}\sum_{j=1}^{i} a_i b_j$. Estimate the number of multiplications and the number of additions needed to compute $s$.

8. (10) Consider an infinite array in which the first $n$ cells store a sequence of $n$ sorted integers $x_1 \le x_2 \le \ldots \le x_n$ and the rest cells are filled with $\infty$. Note that $n$ is not given as input to the algorithm. Design an algorithm that takes an integer $y$ as input and finds a position in the array containing $y$ in $O(\log n)$ time. If $y$ is not in the array, then output 0.

9. (10) After a test, the scores of $n$ students are stored in an array $A[1..n]$. Assume that all scores are positive integers. Give a linear time algorithm to rearrange the $n$ scores stored in the array so that all the scores greater than or equal to 60 appears before the scores less than 60.

10. (10) Suppose that $n$ numbers are to be sorted, each of which is an integer in the following interval. Design a linear time algorithm for this problem, or show that this is impossible.

    (a) $[0, n-1]$
    (b) $[0, n^2-1]$
    (c) $[0, n^3-1]$