1. (10) Derive a formula for the sum $\sum_{i=1}^{n} i2^i$.

2. (10) Let c be a positive number. Show that $f(n) = 1 + c + c^2 + \cdots + c^n$ is
   (a) $\Theta(1)$ if $c < 1$,
   (b) $\Theta(n)$ if $c = 1$,
   (c) $\Theta(c^n)$ if $c > 1$.

3. (10) Show that in any base $b > 1$, the sum of three single-digit numbers is at most two digits long.

4. (10) Let a and b be two n-bit integers, n is very large.
   (a) Is it possible to design an algorithm for computing a^2 which is asymptotically faster than computing $a \times b$?
   (b) Is it possible to design an algorithm for computing $a \times b$ which is asymptotically faster than computing a^2?

5. (20) Suppose that a computer can only do addition (+) and arithmetic shift (<< or >>). Write C code to compute the following statements efficiently.
   (a) $y = 10x$.
   (b) $y = 15x$.

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

6. (20) Give asymptotically tight upper bounds $T(n)$ for each of the following recurrences. Justify your answers.
   (a) $T(n) = 2T(n/2) + n$
   (b) $T(n) = 9T(n/4) + n^2$
   (c) $T(n) = 3T(n/2) + n$
   (d) $T(n) = T(\sqrt{n}) + 1$

7. (20) Consider the following program for computing the greatest common divisor of two positive integers a and b.

   while $(b > 0)$ $\{r = a\%b;\ a = b;\ b = r;\}$ print(a);

   (a) Show that the program will eventually stop and print a correct answer.
   (b) Assume that $a > b$. Show that the number of iterations for the **while** loop is bounded by $\frac{\log b}{\log \theta} + 1$, where $\theta = (1 + \sqrt{5})/2$ is a solution to the equation $x^2 - x - 1 = 0$.

8. (10) Consider an infinite array in which the first n cells store a sequence of n sorted integers $x_1 \leq x_2 \leq \ldots \leq x_n$ and the rest cells are filled with ∞. 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) Give a linear time algorithm to rearrange n keys stored in an array $A[1..n]$ so that all the negative keys appear before the non-negative keys.

10. (20) Consider the linear time algorithm for finding the median of a set S described in the textbook. The set S is initially partitioned into 5-element subsets. Can the algorithm be still linear if the algorithm uses 3 elements, instead of 5. How about partitioning using 7 elements?