1) (5 pts) What is the sum from $i = 0$ to $n$ of $2^i$ for $n = 5$?
   Answer: $1 + 2 + 4 + 8 + 16 + 32 = 63$

2) (5pts) If you are given: $\log_c a$ and $\log_c b$, how would you compute $\log_b a$?
   Answer: $\log_b a = \frac{\log_c a}{\log_c b}$

3) (5pts) Simplify $\sum_{i=0}^{k} c \cdot 2^i$ if $c$ is a constant
   Answer: I would take: $\sum_{i=0}^{k} c \cdot 2^i = c \sum_{i=0}^{k} 2^i$

4) Short answer (10 pts each):
   a) Comment on the statement, “I have an algorithm so complicated that no computer could solve it.”
      Answer: I’ll take a lot of comments, but I am looking for something along the lines of:
      Any algorithm can be converted to a program because both can be “solved” by a Turing machine. (or any sensible assertion that programs are realizations of algorithms.)

   b) List two uses for a loop invariant
      Answer: To insure that the loop terminates, to insure that the loop gives correct answers from correct input, ...and so on.

   c) create a “c” assert statement corresponding to the following invariant (hint: Think about how to step through an array):
      $$A[i] < A[i + 1], i = 1, 2, \ldots, j$$
      Answer (or close to this): for i = 1 to j-1; assert($A[i] < A[i + 1]$);
5) Show the following are correct or incorrect:
   a) (5 pts) $42n^2 - 6n = \Theta(n^3)$
      Answer: We must show that it is both $O(n^3)$ and $\Omega(n^3)$
      
      
      
      $42n^2 - 6n \leq cn^3$
      $42n - 6 \leq cn^2$
      
      This works for $n_0 = 1$ and $c = 42$
      
      
      
      $42n^2 - 6n \geq cn^3$
      $42n - 6 \geq cn^2$
      
      This never works, so the premise is false, $42n^2 - 6n \neq \Theta(n^3)$

   b) (5pts) $10n^2 + 9 = \Theta(n)$
      Answer: We must show that it is both $O(n)$ and $\Omega(n)$
      
      $10n^2 + 9 \leq cn$
      $10 - \frac{9}{n} \leq c$
      
      This works for $n_0 = 1$ and $c = 10$
      
      $10n^2 + 9 \geq cn$
      $10 - \frac{9}{n} \geq c$
      
      This works for $n_0 = 1$ and $c = \frac{1}{2}$ so $10n^2 + 9 = \Theta(n)$

6) Given the following formula, $T(n) = c_1n + 3T\left(\frac{n}{3}\right)$, what can you tell me about the related procedure(5pts)? Draw the recursion tree(5pts).
   Answer: You know that this recursion calls itself three times with each sub-problem $\frac{1}{2}$ the original problem. You also know the combine phase is linear with respect to $n$. I can’t easily draw the picture here (my office) and scan it in.
7) The following program purports to find the maximum element in an array.

Algorithm 1 Max(A[], n)

1: procedure Max(A[], n)
2:     i=2; j=1;
3:     while (i <= n) do
4:         if (A[i] >= A[j]) then
5:             j=i;
6:         end if
7:     i=i+1;
8:     end while
9: end procedure

a) Develop a postcondition for Max in terms of A, n, j(10 pts) Answer: Post condition: for
   \( i \leq n, A[i] \leq A[j] \)

b) Formulate a loop invariant for the while loop(hint: Think about how Max explores the
   loop)(10 pts) Answer: I: for \( k = 1 \) to \( i, A[k] \leq a[j] \)

c) Show that your loop invariant is correct with respect to the program post conditions(10 pts)
8) (10pts) The SPAMCO computer corporation claims their new computer will run 100 times faster than that of their competitor, Dogbert, Inc. Given two algorithms A and B with running times as follows:

\[ t_A(n) = 5000n \]
\[ t_B(n) = 10n^2 \]

For what values of \( n \), if any, will the Dogbert computer run algorithm B faster than the SPAMCO computer will run algorithm A?

9) Prove that the solution of following recurrence for \( n \) a power of 2:

\[
T(n) = \begin{cases} 
2T(n/2) + cn & n > 1 \\
c & \text{otherwise}
\end{cases}
\]

is

\[ T(n) = O(n \log_2 n) \]