Name:

You must show your work for full credit. Guessing or simply “knowing” the answer will only be worth partial credit. You may make assumptions to remove the floor and ceiling functions. While there are 105 points on the exam, the most credit you can get is 100 points for 100 per cent. Write your name and student number on each sheet you turn in. You are allowed one sheet of notes (front and back). You do not need to turn in your notes sheet.

1) (5 pts) What is the sum from $i = 0$ to $n$ of $2^i$ for $n = 5$?

$$\sum_{i=0}^{5} 2^i = 2^0 + 2^1 + 2^2 + 2^3 + 2^4 + 2^5 = 1 + 2 + 4 + 8 + 16 + 32 = 63$$

2) (5 pts) If you are given: $\log_c a$ and $\log_b a$, how would you compute $\log_a b$?

$$\log_a b = \frac{\log_c a}{\log_c b}$$

3) (5 pts) Simplify $\sum_{i=0}^{k} c \cdot 2^i$ if $c$ is a constant.

$$\sum_{i=0}^{k} c \cdot 2^i = c \sum_{i=0}^{k} 2^i$$

$$= c \left[ \frac{2^{k+1} - 1}{2-1} \right]$$

4) Short answer (10 pts each):

a) Comment on the statement, “I have an algorithm so complicated that no computer could solve it.”

This is not possible. Any algorithm can be changed into a program, but not the other way around.
b) List two uses for a loop invariant
   i) To insure that a loop works correctly (prove).
   ii) To help find off-by-one errors

c) Create a “c” assert statement corresponding to the following invariant (hint: Think about how to step through an array):

\[
A[i] < A[i+1], i = 1, 2, \ldots, j
\]

```
assert (in-order(A, i, j))

bool in-order(A, i, j) {
    while (i <= j) {
        if (A[i] > A[i+1])
            return false;
        i = i + 1;
    }
    return true;
}
```

5) Show the following are correct or incorrect:
   a) (5 pts) \(42n^2 - 6n = \Theta(n^3)\)

For \(O(\cdot)\): \(42n^2 - 6n \leq cn^3\)

\[\frac{42}{n} - \frac{6}{n^2} \leq c\]

As \(n \to \text{large}\) both \(\frac{42}{n}\) and \(\frac{6}{n^2}\) will go towards 0.

\[\therefore \text{we cannot choose a positive } c \text{ small enough to make this true.}\]
b) (5pts) $10n^2 + 9 = O(n^2)$

Proof: $10n^2 + 9 \leq c n^2$

$$10 + \frac{9}{n^2} \leq c$$

as $n \to \infty$, $\frac{9}{n^2}$ will get smaller and smaller.

.: The smallest largest value for the left side will be 9, so $c = 19$ works $\square$

6) Given the following formula, $T(n) = c_1 n + 3T\left(\frac{n}{3}\right)$, what can you tell me about the related procedure (5pts)? Draw the recursion tree (5pts).

- number of branches $b$ — how the problem is split

$$c_1 n$$

$T(\frac{n}{3})$ $T(\frac{n}{3})$ $T(\frac{n}{3})$

$$c_1 \frac{n}{3}$$

$T(\frac{n}{9})$ $T(\frac{n}{9})$ $T(\frac{n}{9})$

$$c_1 \frac{n}{9}$$

$T(\frac{n}{27})$ $T(\frac{n}{27})$ $T(\frac{n}{27})$ $T(\frac{n}{27})$ $T(\frac{n}{27})$

$$c_1 \frac{n}{27}$$

So, it appears this will look like

$O(n \log_3 n) \to O(n \log n)$
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)

\[
(A[j] \geq A[i] \text{ } i=1...n \text{ } i \neq j) \text{ and } \neg (i \leq n)
\]

b) Formulate a loop invariant for the while loop (hint: Think about how Max explores the loop) (10 pts)

\[
I: A[j] \geq A[k] \text{ } k=1...(i-1) \text{ and } j
\]

c) Show that your loop invariant is correct with respect to the program post conditions (10 pts)

\[
I \land \neg (G) \rightarrow \text{post conditions}
\]

\[
A[j] \geq A[k] \text{ } k=1... (i-1) \text{ and } \neg (i \leq n)
\]

\[
\neg (i \leq n) \rightarrow i = n+1
\]

\[
A[j] \geq A[k] \text{ } k=1... (n+1)-1
\]

\[
A[j] \geq A[k] \text{ } k=1... n \text{ as we want it to be}
\]
8) (10pts) The SPAMCO computer corporation claims their new computer will run 100 tiems 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 A faster than the SPAMCO computer will run algorithm B?

\[
\frac{t_A(n)}{100} = \frac{5000n}{10n^2} = \frac{500}{n} \]
\[
10 \cdot 5000 = n^2
\]
\[
100 \cdot 50,000 = n^2
\]
\[
50,000 = n
\]

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

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

is \( T(n) = O(n \log_2 n) \)

By Master Theorem: \( n^{\log_2 2} \rightarrow n \) and \( f(n) = cn \)

Case 2: \( T(n) = O(c \cdot n \log_2 n) \)

Substitution: guess \( cn \log_2 n \)

\[
T(n) = 2T(\frac{n}{2}) + cn \leq c \cdot n \log_2 \frac{n}{2} + cn \leq c \cdot n \log_2 n
\]

Recursion Tree on Back.
So, each level contributes $c_n$ to the work. The question is to find how many levels. Since we know $n = 2^k$ from the problem and $k$ is "roughly" equal to the number of levels (for large $n$), $n = 2^k \Rightarrow \log n = k$ (taking log of both sides). Therefore we have

$$c_n \cdot k \rightarrow c_n \log n$$