Part 1 - Bounds and Runtime Analysis [18 Points]

1. For the following summation determine the unqualified ***O***, **Ω**, and **Θ** bounds, if they exist. For any bounds that do not exist, give a reason why. [4 points]

$$f(n) = \sum_{i=1}^{n^4} \sum_{j=1}^{n} 42i$$

***O*(*n*^{9}), Ω(*n*^{9}), Θ(*n*^{9})**

2. For the following function determine the unqualified ***O***, **Ω**, and **Θ** bounds, if they exist. For any bounds that do not exist, give a reason why. [4 points]

$$f(n) = 4log(2^{n^3}) + n^2 - 16$$

***O*(*n*^{3}), Ω(*n*^{3}), Θ(*n*^{3})**

3. For the following function determine the unqualified ***O***, **Ω**, and **Θ** bounds, if they exist. For any bounds that do not exist, give a reason why. [4 points]

$$f(n) = \begin{cases} log(n) & \textbf{if } n \textbf{ is prime} \\ 16n & \textbf{if } n > 10 \textbf{ and is even} \\ 19nlog(n) & \textbf{otherwise} \end{cases}$$

***O*(*n log*(*n*)), Ω(*log*(*n*)), Θ bound does not exist, because tight *O* and tight Ω are not the same.**

4. For the following functions, prove that $f(n) \in O(g(n))$ by finding appropriate values for the constants *c* and n_0. Show all work. [6 points]

$$f(n) = 6n^3 + 14n - 2$$

$$g(n) = 2n^3$$

Consider the following inequalities:

$6n^3 \leq c_1 * 2n^3$ ← This is true if $c_1 = 3$ and $n_0 = 0$

$14n \leq c_2 * 2n^3$ ← This is true if $c_2 = 7$ and $n_0 = 0$

$-2 \leq c_3 * 2n^3$ ← This is true if $c_3 = 1$ and $n_0 = 0$

So if we set c to 11 and n_0 to 0, we have:
f(n) ≤ c * g(n) for all n > n_0, therefore f(n) is in *O*(g(n))

[1 point] for valid c and n_0 or for showing the limit test
[2 points] for correct setup (showing what it means to be in O)
[3 points] for valid work

Note there are many constants that satisfy this inequality

Part 4 - Arrays and Linked Lists [20 Points]

For questions in this part, consider the following code, in which `array` is an `ArrayList<String>`, and `list` is a `LinkedList<String>`, and x is a valid index:

```
array.add(x, "foo")
array.add("bar")
list.add("baz")
```

5. Assume we don't know anything about the value of x. What is the unqualified worst-case runtime for the call inserting `"foo"` in the above code if we assume that the underlying array in our `ArrayList` is not full? How does your answer change if we cannot assume that the underlying array is not full? [5 points]

[2 points] *O*(n), [3 points] without the assumption it is still *O*(n)

6. What is the unqualified worst-case runtime for the call inserting `"bar"` in the above code if we assume that the underlying array in our `ArrayList` is not full? How does your answer change if we cannot assume that the underlying array is not full? [5 points]

   [2 points] *O*(1), [3 points] without the assumption it becomes *O*(n)

7. What is the unqualified worst-case runtime for the call inserting `"baz"` in the above code if list is a singly linked list (assume no reference to tail)? [5 points]

   [5 points] *O*(n)

8. What is the unqualified worst-case runtime for the call inserting `"baz"` in the above code if list is a doubly linked list (assume reference to tail exists)? [5 points]

   [5 points] *O*(1)

Part 5 - Short Answer [10 Points]

9. In one or two sentences, describe the difference between an ADT and a data structure. Then name 2 ADTs and 2 data structures we have described in class. [5 points]

   [1 point] ADTs just describe what you can do with the data, the data structure is the actual implementation of those capabilities.
   [2 points] ADTS: Sequence, List
   [2 points] Data Structures: Array, ArrayList, LinkedList

10. Consider the following runtime function for a recursive algorithm: [5 points]

$$T(n) = \begin{cases} \Theta(1) & \textbf{if } n \leq 1 \\ 2 \cdot T(\frac{n}{2}) + \Theta(1) + \Theta(n) & \textbf{otherwise} \end{cases}$$

If we hypothesize that the runtime of this recursive algorithm is ***O*(*n* log(*n*))** state inequality we must prove for the base case, and state the inequality we must use as the assumption for the inductive case.

[2 points] Base: T(1) ≤ c * 1 * log(1)

[3 points] Assumption: T(n/2) ≤ c * n/2 * log(n/2)

Part 7 - Extra Credit [5 Points]

11. Is it possible to have a function, $f(n)$, that is in both ***O*(*n*²)** and **Ω(log *n*)**? If so, give an example.

[1 point] Yes. [4 points] Many possible examples: n^2, log(n), n, piece-wise functions, etc.