Practice Quiz 2

Names ________________________________

Problem 1 (20 points)

In this problem, you will prove that $2n^2 + 3 = O(n^3)$ using the definition of big O. Follow the three steps carefully.

(5 points) Write down the definition of big O.

(10 points) Give a c and a n_0 that can be used to prove that $2n^2 + 3 = O(n^3)$.

(5 points) Explain what it would mean for this c and n_0 to work in a proof that $2n^2 + 3 = O(n^3)$, and *very briefly* explain why they do (write one sentence, draw a graph, etc).

Problem 2 (20 points)

Complete the proof that $\sum_{i=0}^{n} i = \frac{1}{2}n(n+1)$ using mathematical induction by filling in the following. Note that $\sum_{i=0}^{n} i = 0 + 1 + 2 + \cdots + (n-1) + n$. Each underline is worth one point. The proof of the base case is worth two points. The proof of the inductive case is worth eight points.

Proof. We prove that $\sum_{i=0}^{n} i = \frac{1}{2}n(n+1)$.

First, let $P(n)$ be the predicate that ____________________________________.

We prove that __________________ (something to do with P) using mathematical induction over n.

Base case. We show that _________________ (something to do with P).

Inductive case. We show that _____________________________________ (something to do with P).

Assume _________________ (something to do with P). That is, _____________________________________ (translating the previous blank using the formula, aka, the inductive hypothesis, or IH.).

Because we proved both _________________ and ________________________________, by the principle of mathematical induction, $\forall n \geq 0 : P(n)$. $\square$

Problem 3 (40 points)

(a) Give an example of a function that is **not** one-to-one.

(b) Suppose $f : A \to B$ is both onto and one-to-one. How do $|A|$ and $|B|$ compare?

(c) Let $f : \mathbb{Z} \to \mathbb{Z}$ be defined as $f(n) = 2n$. What is the codomain of f? What is the range of f?

(d) Suppose we have a recurrence relation describing the runtime of a recursive algorithm as follows: $T(0) = 5$ and $T(n) = T(n-1) + 3$. What is $T(2)$?

(e) Is $n = O(\log n)$?

(f) Is $\log n = O(n)$?

(g) Fill in the rest of the recursive definition of a linked list.

A linked list is either

(1) An empty list $\langle\rangle$, or

(2) $\langle x, L\rangle$ where x is a data element and L is _____

Problem 4 (20 points)

(a) For the following algorithm give a proposed function representing the number of primitive operations for the algorithm in terms of the input size, addressing each line and/or loop of the algorithm. You do not need to be precise counting constant numbers of primitive operations (e.g., figuring out exactly how many primitive operations a single line does). However, you should try to be precise about how many times a loop runs.

Algorithm 1

```
for i = 1 to 2n do
    j = i;
    while j > 1 do
        j = j/3;
```

Algorithm 1 takes $f(n) =$ primitive operations.

(b) For the $f(n)$ you gave in (a), give the "tightest" (aka asymptotically smallest) $g(n)$ such that $f(n) = O(g(n))$.

$f(n) = O($ $)$