Quiz 3

Key

TRUE/FALSE: Write "True" in each of the following cases if the statement is *always* true and briefly justify your answer. Otherwise, write "False" and provide brief reasoning.

1. [2] (HW4 §3.2 #2) The function $5,000,000n^2$ is $O(2^n)$. computations (+1)

True (+.5) Note (+.5) $\lim_{n\to\infty} \frac{5,000,000n^2}{2^n} \overset{L'H}{=} \lim_{n\to\infty} \frac{10,000,000n}{2^n(\ln 2)} \overset{L'H}{=} \lim_{n\to\infty} \frac{10,000,000}{2^n(\ln 2)(\ln 2)} = 0$

or

Knows big O (+.5)

Consider the graphs (+.5) of $5,000,000n^2$ and 2^n as $n\to\infty$. The graph of 2^n is much larger as $n\to\infty$ so 2^n is an upper bound.

(+1)

2. [2] (HW4 §3.2 #2) The function 2^n is $O(5,000,000n^2)$.

False (+.5) The same work from # 1 applies here (+.5)

$\lim_{n\to\infty} \frac{2^n}{5,000,000n^2} = \lim_{n\to\infty} \frac{2^n(\ln 2)(\ln 2)}{10,000,000} = \infty$ not a constant!

or

Knows big O (+.5)

The graphs above indicate as $n\to\infty$, $5,000,000n^2$ is overtaken by 2^n. (+.5)

Free Response: Show your work! No credit is given without supporting work.

3. Define the map $f : \mathbb{R} \to \mathbb{Z}$ by $f(x) = 3\lfloor x+1 \rfloor$.

(a) [2] (functions wks #2) Find $f(\frac{3}{2})$

$3\lfloor \frac{3}{2}+1 \rfloor = 3\lfloor \frac{5}{2} \rfloor = 3\lfloor 2.5 \rfloor = 3\cdot 2 = 6$

(+.5) (+.5) got it (+.5)

notation (+.5)

(b) [2] (functions wks #3) Find $f(f(1.5))$.

$f(f(1.5)) = f(3\lfloor 1.5+1 \rfloor)$

composition (+.5)
algebra (+.5)

$= f(3\lfloor 2.5 \rfloor)$

$= f(3\cdot 2) = f(6)$ (+.5)

$= 3\lfloor 6+1 \rfloor = 3\cdot 7 = 21$ (+.5)

4. [3] (§3.3 #3) Give a big-O estimate for the number of additions and multiplication used in the below segment of an algorithm. Justify your answer.

```
t := 0
for i = 1 to 3 do
    for j = 1 to 4 do
        t := t + ij
    end
end
return t
```

partial - 2 for loops +.5
nested +.5
index +.5

Algorithm 1: Adding Multiples

+1

O(1) Note we will perform one addition + one mult each time we go thru the inner for loop. For each i=1, 2, and 3 we'll let j vary from 1 to 4 for a total call of 3·4 or 12 calls. Thus we'll perform 2·12 operations or 24

+1 explain code

n explain # calls

5. [5] Write an algorithm that is big-O of n^3.

```
t = 0
for i = 0 to n do
    for j = 0 to n do
        for k = 0 to n do
            t = t + i + j + k
        end
    end
end
```

start +.5
nested +1
index +1.5
algorithm form +1
notation / sense +1

6. Recall that Binary Search is $O(\log n)$ and that Bubble Sort is $O(n^2)$.

(a) [2] (HW4 §3.3 #4) How does the number of comparisons change in Bubble Sort when the inputs change from n to n^2.

Start +.5
+.5 notation/sense

Bubble sort is $O(n^2)$ so if the list length changes from n to n^2 the # of comparisons changes from n^2 to $(n^2)^2 = n^4$

+1

(b) [2] (§3.3 #21) How does the number of comparisons change in Binary Search when the inputs change from n to n^2.

Start +.5
+.5 notation/sense

if the list length changes from n to n^2 the # of comparisons changes from $\log n$ to $\log n^2 = 2\log n$

(the time doubles which is way better than n2)

+1