CSE 373 Summer 2016 Homework 2

Due Thursday, 7/7 at 11PM

Name: ________________________

UWID (not your student number): ________________________

1) Big-O

For each of the following, show that $f \in$ O(g). That is, you will need to find values for c and n0 such that the definition of big-O holds true as we did with the examples in lecture. It may help to look at the handwritten example from lecture.

a) $f(n) = 12n^2 + 2500$ $\quad$ $g(n) = n^3$

b) $f(n) = 3(\log(n) + 1)$ $\quad$ $g(n) = 2n$

c) $f(n) = 7n$ $\quad$ $g(n) = \frac{n}{15}$

2) Runtime Analysis

For each of the following program fragments, determine the asymptotic runtime in terms of n.

a)

```
public void mysteryOne(int n) {
     int y = 1;
     for (int j = 0; j < ((n * n - 5) / 4); j++) {
          for (int i = 0; i < n; i ++) {
               y *= y;
          }
     }
}
```

b)

```
public void mysteryTwo(int n) {
     int x = 0;
     for (int i = n + 3; i >= 0; i - - ) {
          if (i % 4 == 0) {
               break;
          } else {
               for (int j = 1; j < n; j *= 2) {
                    x++;
               }
          }
     }
}
```

c)

```
public void mysteryThree(int n) {
     for (int i = 0; i < n; i++) {
          helper(i);
     }
}

private void helper(int x) {
     if (x > 0) {
          helper(x - 2);
     }
}
```

3) More Asymptotic Analysis

For each of the following, determine if $f \in$ O(g), $f \in \Omega$(g), $f \in \Theta$(g), **several of these, or none of these.**

a) $f(n) = 75n^3 + 2$ $g(n) = n^3 + 6n + n^2$

b) $f(n) = 3^n$ $g(n) = n^3$

c) $f(n) = \log(n)$ $g(n) = \log(\log(n))$

4) Pseudocode and Recurrence Relations

a) Write pseudocode for a function that calculates the largest difference between any two numbers in an array of positive integers with a runtime in $\Theta(n^2)$.

For example, the largest difference between any two numbers in the following array would be 19.

a = [4, 6, 3, 9, 2, 1, 20]

b) Can this function be written with a runtime in $\Theta(n)$? If yes, write pseudocode below. If no, why? What would have to be different about the input in order to do so?

c) Can this function be written with a runtime in $\Theta(1)$?. If yes, write pseudocode below. If no, why? What would have to be different about the input in order to do so?

5) Recurrence Relations

Note: For both of these problems, the base case can be *$T(c) = d$, where both c and d are constants.*

We are asking for the tightest Big-Oh bound in a) and b).
For example, the **tightest** big-oh bound for $f(n) = 5n$ is $O(n)$, not $O(n^2)$

a) **Find the tightest Big-Oh bound for the following recurrence relation** $T(n) = n + T(n/2)$. **Justify your answer.**

b) **Find the tightest Big-Oh bound for the following recurrence relation** $T(n) = n + 2T(n/2)$**. Justify your answer.**

6) Growth Rates

Order the following functions from slowest to fastest in terms of asymptotic runtime. Be sure to show whether two functions have the same asymptotic runtime.

Ex: $5n < 3n^2 = n^2$

- n^{72}
- $n^2 \log(n)$
- $2^{(n/2)}$
- $\log(n)$
- $n \log(n^2)$
- n^6
- $n \log(\log(n))$
- $n \log^2(n)$
- n
- n^2
- $n \log(n)$
- 2^n
- $\log^{2(}n)$
- $2/n$
- $2^{(1/2)}$

7) Induction

Use induction to prove that $\sum_{i=1}^{n} i^2 = \frac{1}{6}n(n+1)(2n+1)$ **for** *n >= 1*

Be sure to clearly mark each step, and where you use the inductive hypothesis.