TCSS 342 Winter 2018

Assignment 1
Due Jan 14

1. [20%] **Big Oh**

   A Let $f(n) = 123456n + 251$, prove that $f(n) = O(n)$. [5%]

   B Let $f(n) = n^2$, prove that $f(n)$ is not $O(n)$. [5%]

   C Is $f(n) = n!$ in $\Theta(2^n)$. Prove or disprove it. [5%, Bonus]

   D Is $f(n) = \log n$ in $O(n^{0.1})$. Prove or disprove it. [5%]

2. [15%] Give a pair of functions $f(n), g(n)$ such that $f(n) \notin O(g(n))$ and $g(n) \notin O(f(n))$.

3. [20%] State true or false. Give a 3-line justification for each question.

   A If the worst-case running time of an algorithm A is $O(n)$ and the worst-case running time of an algorithm B is $O(n^2)$, then algorithm A will run faster than algorithm B on all inputs. [10%]

   B Is $n \in O(n^{|\cos(n)|})$, where $|\ |$ denote the absolute value of $\cos(n)$. [10%]

4.[20%] **Tight bound on Time complexity.** For the following pieces of code, give the tightest $\Theta(\ )$ estimate that you can, and Show all the working. Part A, B each worth [5%] and C worth [10%].

A int $total = 0$;

```
for (int i = 0; i < n; i + +){
    for (int j = 0; j < 5; j + +){
        total = total + 1;
    }
}
```

B int $total = 0$;

```
for (int i = n; i < n − 1; i + +){
    for (int j = 0; j < n; j + +){
        total = total + 1;
    }
}
```

C int $total = 0$;

```
for (int i = n; i > 0; i = i/2){
    total = total + 1;
}
```

5. [25%] **Arrays**

A Suppose that you are given an array of integers. Write a Java program which prints out the element or elements that occur least often in the array, and the number of times they occur. For example, for array [1 4 4 3 4 3 5 2 1] your algorithm should print the message:

Elements: 5 2

Number of occurrences: 1

The implementation should be done in the main method of your class. The only data structure you are allowed to use is arrays. No hash map etc. You must submit a java file (.java extension). [20%]

B What is the worst-case time complexity (O( )) of your algorithm? [5%]