This document provides the necessary instructions for completing the Week 4 lab exercises.

**Exercise 1: Counting Primitive Operations**

Count the primitive operations for the following algorithms/methods:

//

**int** n = 1000;

System.***out***.println("Hey - your input is: " + n);

System.***out***.println("Hmm.. I'm doing more stuff with: " + n);

System.***out***.println("And more: " + n);

//

**for** (**int** i = 1; i < n; i = i \* 2){

System.***out***.println("Hey - I'm busy looking at: " + i);

}

//

**for** (**int** i = 1; i <= n; i++){

**for**(**int** j = 1; j < n; j = j \* 2) {

System.***out***.println("Hey - I'm busy looking at: " + i + " and " + j);

}

}

**//**

**public** **static** **double**[] prefixAverage1(**double**[] x) {

**int** n = x.length;

**double**[] a = **new** **double**[n]; // filled with zeros by default

**for** (**int** j=0; j < n; j++) {

**double** total = 0; // begin computing x[0] + ... + x[j]

**for** (**int** i=0; i <= j; i++)

total += x[i];

a[j] = total / (j+1); // record the average

}

**return** a;

}

**public** **static** **double**[] prefixAverage2(**double**[] x) {

**int** n = x.length;

**double**[] a = **new** **double**[n]; // filled with zeros by default

**double** total = 0; // compute prefix sum as x[0] + x[1] + ...

**for** (**int** j=0; j < n; j++) {

total += x[j]; // update prefix sum to include x[j]

a[j] = total / (j+1); // compute average based on current sum

}

**return** a;

}

**Exercise 2: Big O descriptions**

**Describe the following Big O notations for algorithm runtimes:**

**O(1) –** the algorithm has a constant execution time

**O(n) -**

**O(logn) -**

**O(nlogn) -**

**O(n^2) –**

**O(n^3) -**

**O(2^n) -**

**O(n!) -**

**Exercise 3: Big O Examples:**

Recall the definition: f(n) <= c\* g(n) for n >=n0

**Let** f(n) = 2\*n + 10.

Calculate **c** and **n0**.

Do the same for the following functions:

1. 15\*n -2
2. 7\*log n – 5
3. 10\* n3 + 15\*n2 -15
4. 2*n*4 +7*n*3- 5*n*2 + 2*n* - 7

**Exercise 4: Asymptotic Analysis and Big O notation**

**What is the growth rate for the following functions? Express it in terms of Big Oh notation:**

1. All the examples of exercise 1
2. 1000\*n + 1
3. n ( n-1000)
4. 100\* (n-1)3 - 10\*(n2 -15) + 3n -10000
5. (n-10)\*(n+15) + 100000
6. 2n+100log n
7. 3logn+2 is O(logn)
8. 20n3 +10nlog n+5 is O(n3)

**Exercise 5: Asymptotic growth rate**

Order the following functions by asymptotic growth rate.

4nlogn+2n, 210 , 2logn . 3n+100logn, 4n, 2n , n2+10n, n3 ,nlogn

Hint: Simplify the expressions, and then use the ordering of the seven important algorithm-analysis functions to order this set.