**COSC 240 Introduction to Algorithms
Problem Set 1 Total points: 56
Spring 2019**

**Assigned on 1/22/2019
Due at start of class on 1/29/2019**

**Question 1**. Prove the following statements using the definitions of asymptotic notations learned in class. (Identify constants such as *c*, *n0* so that the statements would hold):

1. (2 points) $2n +8=O\left(n^{3}\right)$
2. (2 points) $\frac{n}{100}=Ω(π) $
3. (4 points) $3n^{2}+11n+6= Θ\left(n^{2}\right)$

(Recall: to show f(n) = Θ(g(n)), you must show both that f(n) = O(g(n)) *and* that f (n) = Ω(g (n)).)

Questions 2, 3 and 4 use the recurrence below.

$$T\left(n\right)= \left\{\begin{matrix} 3T\left(\frac{n}{4}\right)+n if n\geq 4\\ Θ(1) if n<4\end{matrix}\right.$$

**Question 2**. (8 points) Find a big-Θ bound for T(n) using the master method.
 ***Note***: Clearly identify ϵ. If case 3 applies, provide a constant $c$ by which the regularity condition holds.

**Question 3**. (8 points) Find a big-Θ bound for T(n) using the substitution method.

**Question 4**. (8 points) Assuming that n is a power of 4, draw a recursion tree for the above recurrence, and label it with the following: (a) the number of levels; (b) number of nodes per level; and (c) cost of leaf level. *Please draw at least 1 full level of the recursion tree in addition to the root.*

**Question 5**. (8 points) Use a loop invariant to prove that the following subroutine returns the sum of the elements of array A, where n is the length of the array.

**Sum**(A, n)

 i 🡨 n
 sum 🡨 0
 while (i >= 1)
 sum 🡨 sum + A[i]
 i 🡨 i-1
 return sum

**Question 6** (8 points)

Find the big-Θ$ $bound on the expected execution time for the following algorithm called KeepRolling().

$RollSixSidedDie()$ is a Θ(1) function that returns a random integer between 1 and 6. Assume that the function rolls a fair die, i.e., each integer between 1 and 6 (inclusive) has an equal probability to be the outcome of the function.

**KeepRolling()**

 *flag* 🡨 True

 While (*flag* == True)

 *p 🡨 RollSixSidedDie()*

 If (*p* == 6)
 *flag* 🡨 False

**Question 7** (8 points) Assume that in a particular system the array input A for insertion sort is “almost-sorted” in the following sense:
 there exists at most one value of i (1<=i<=n-1) such that A[i]>A[i+1]

where n is the length of the array (the array is assumed to be indexed from 1 to n).

For such inputs, estimate big- Θ bound for worst-case execution time of insertion sort.