**CS311 DSA Homework 1**

**Your Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Univ. ID: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Instructions**

* Check **due date** and polices on course webpage.
* Write the answer for each question in the space provided below the question.
* Submit your answers to cs\_scu@foxmail.com.
* Submission file name format: CS311\_Hw1\_yourID\_yourLastName.doc (or pdf).

1. (10 points) Use the laws of logical equivalences (do not use truth tables) to show the following equivalence:

*A* ∧ ¬(*A* ∧ *B*) ≡ *A* ∧ ¬*B*

Supply a reason for each step.

***Answer*:**

2. (10 points) If *n* is an integer, then

*n* is even  ∃an integer *k* such that *n* = 2*k*.

*n* is odd  ∃an integer *k* such that *n* = 2*k*+1.

Prove the following proposition using the method of proof by contraposition:

For all integers *n*, if *n*2 is even then *n* is even.

***Answer*:**

3. (15 points) Recursion

1. Recursively define the sequence: 3, 6, 9, 12, 15, ...

***Answer*:**

1. Give f(3), f(4) and f(5) for the following recursively defined sequence

f(1) = 1, f(2) = 3

f(n) = f(n-1)\*f(n-2) + 1 for n > 2

***Answer*:**

1. Give the results of foo(4, 3) for the following program float foo(float x, int n)

{

if (n==0) /\*if (n==0)\*/

return 0

else

return x+ foo(x, n-1)

}

Support the detail of each step.

***Answer*:**

4. (10 points)

1. Order the following functions by growth rate. Indicate the functions that grow at the same rate:

*n*3, *n*2log(*n*), *n*2,2*n*, 64, *n*2 + log(*n*).

***Answer*:**

1. Give the complexity of the following functions in big-O (the bounds should be tight)
2. *f*(*n*) = *n*3 + 2*n* + 5*n*4 + 36 + 1/*n*
3. *f*(*n*) = (3*n*)1000 + *n*3000 + *n*2000(log *n*)1000 + 2*n*

***Answer*:**

5. (10 points)What does the following algorithm do? Analyze its worst-case running time, and express it using “Big-Oh" notation (we want a tight bound).

**Algorithm** Foo (*a*, *n*):

*k* =0

*b* =1

**while** *k* < *n* **do**

*k* = *k* + 1

*b* = *b* \* *a*

**return** *b*

***Answer*:**

6. (15 points) For each of the following loops, give the tightest upper bound using big *O* notation.

a. for ( int *i* = 0; *i* < *n*\**n*; *i*++)

*sum*++;

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

*sum*++;

b. for ( int *i* = 0; *i* < *n*; *i*++)

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

*sum*++;

c. for ( int *i* = 0; *i* < *n*; *i* ++)

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

*sum*++;

**Answer:**

7. (10 points) A collection *T* of numbers is defined recursively by

1. 3 belongs to *T*
2. If *x* belongs to *T*, so does *x*+5 and 3\**x*

Which of the following belong to *T*?

11 12 13 14 19

***Answer*:**

8. (10 points) A sequence *s*0, *s*1, *s*2, *s*3, … is defined by

*s*0=1 and for *n* > 0, *sn*= 2*sn*-1

* 1. Write the first 5 terms in the sequence
  2. Prove the following formula for the sum of the first *n* terms (*n* ≥ 0) using induction:

*s*0+*s*1+*s*2…+*sn*= 2*n*+1 - 1

***Answer*:**

9. (10 points) Suppose the towers of Hanoi puzzle has three poles and *n* rings, write your ideas to solve it (*pseudocodes are optional*).

***Answer*:**