Computer Science Foundation Exam

August 12, 2016

Section I A

COMPUTER SCIENCE

**NO books, notes, or calculators may be used,
and you must work entirely on your own.**

Name: ______________________________

UCFID: ______________________________

NID: ______________________________

Question #	Max Pts	Category	Passing	Score
1	10	DSN	7	
2	10	ANL	7	
3	10	ALG	7	
4	10	ALG	7	
5	10	ALG	7	
TOTAL	50			

You must do all 5 problems in this section of the exam.

Problems will be graded based on the completeness of the solution steps and <u>not</u> graded based on the answer alone. Credit cannot be given unless all work is shown and is readable. Be complete, yet concise, and above all <u>be neat</u>.

1) (10 pts) DSN (Recursive Functions)

Write a recursive function that will return the binary equivalent of its input parameter, `decimalNo`. You may assume that `decimalNo` is in between 0 and 1023, inclusive, thus the converted binary value will fit into an integer variable. For example, toBinary(46) should return the integer 101110 and toBinary(512) should return 1000000000.

```
int toBinary(int decimalNo) {

}
```

2) (10 pts) ANL (Summations and Algorithm Analysis)

Find the closed form solution in terms of n for the following summation. Be sure to show all your work.

$$\sum_{i=n}^{3n} \sum_{j=1}^{n-2} j$$

3) (10 pts) ALG (Stacks and Queues)

Consider the process of merging two queues, `q1` and `q2`, into one queue. One way to manage this process fairly is to take the first item in `q1`, then the first item from `q2`, and continue alternating from the two queues until one of the queues run out, followed by taking all of the items from the queue that has yet to run out in the original order. For example, if `q1` contains 3(front), 8, 2, 7 and 5, and `q2` contains 6(front), 11, 9, 1, 4 and 10, then merging the two queues would create a queue with the following items in this order: 3(front), 6, 8, 11, 2, 9, 7, 1, 5, 4, and 10. Assume that the following struct definitions and functions with the signatures shown below already exist.

```
typedef struct node {
    int data;
    struct node* next;
} node;

typedef struct queue {
    node* front;
    node* back;
} queue;


// Initializes the queue pointed to by myQ to be an empty queue.
void initialize(queue* myQ);

// Enqueues the node pointed to by item into the queue pointed to by
// myQ.
void enqueue(queue* myQ, node* item);

// Removes and returns the front node stored in the queue pointed to
// by myQ. Returns NULL if myQ is empty.
node* dequeue(queue* myQ);

// Returns the number of items in the queue pointed to by myQ.
int size(queue* myQ);
```

On the following page, write a function that takes in two queues, `q1` and `q2`, merges these into a single queue, by dequeuing all items from `q1` and `q2` using the process described above and enqueuing those items into a new queue, and returns a pointer to the resulting queue.

```
queue* merge(queue* q1, queue* q2) {

}
```

4) (10 pts) ALG (Hash Tables)

Insert the following numbers (in the order that they are shown…..from left to right) into a hash table with an array of size 12, using the hash function, H(*x*) = *x* mod 12.

234, 344, 481, 567, 893, 178, 719, 686, 46, 84

Show the result of the insertions when hash collisions are resolved through

a) Linear Probing

b) Quadratic Probing

c) Separate Chaining Hashing

Index	**a** **Linear**	**b** **Quadratic**	**c** **Separate Chaining**
0			
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			

5) (10 pts) ALG (Base Conversion)

(a) (5 pts) Convert $FAD8_{16}$ to octal.

(b) (5 pts) Convert 2120_{10} to hexadecimal.

Computer Science Foundation Exam

August 12, 2016

Section I B

COMPUTER SCIENCE

NO books, notes, or calculators may be used,
and you must work entirely on your own.

Name: ____

UCFID: ____

NID: ____

Question #	Max Pts	Category	Passing	Score
1	10	ALG	7	
2	10	ANL	7	
3	10	DSN	7	
4	10	DSN	7	
5	10	ALG	7	
TOTAL	50		35	

You must do all 5 problems in this section of the exam.

Problems will be graded based on the completeness of the solution steps and not graded based on the answer alone. Credit cannot be given unless all work is shown and is readable. Be complete, yet concise, and above all be neat.

1) (10 pts) ALG (Analysis and Critical Thinking: AVL Trees, Hash Tables, and Heaps)

a) (1 pt) Using big-oh notation, what is the **best-case** runtime for inserting an integer into an AVL tree that contains *n* integers?

b) (1 pt) Using big-oh notation, what is the **worst-case** runtime for inserting an integer into an AVL tree that contains *n* integers?

c) (2 pts) What is the worst-case runtime for insertion into a hash table with *n* elements, assuming we use quadratic probing to resolve collisions? (You may assume that our hash table satisfies all conditions necessary to ensure that quadratic probing won't get stuck in an infinite loop.)

d) (2 pts) Given the following hash table, suppose we know that no strings have been deleted, but we don't know the order in which these three strings were inserted into the hash table. If we used linear probing to resolve collisions, what are all the possible hash values for the string "of" (assuming those hash values are modded by the table size, so the only valid values are 0 through 6)?

		pied	piper	of		Hamelin
0	1	2	3	4	5	6

e) (2 pts) Using big-oh notation, what is the **worst-case** runtime for deletion from a minheap that contains *n* elements?

f) (2 pts) Draw a minheap that contains 10 elements and which will incur the worst-case runtime if we call deleteMin() on it.

2) (10 pts) ANL (Summations and Algorithm Analysis)

a) (8 pts) Give a summation that represents the value returned by the following function in terms of its input parameter, n, and then derive its closed form:

```
int something_to_ponder_over(int n)
{
   int i, retval = 0, pow = 1;
   for (i = 0; i < n; i++)
   {
      retval += pow;
      pow *= 14;
   }
   return retval;
}
```

b) (2 pts) Using big-oh notation, what is the runtime of the function given in part (a)?

3) (10pts) DSN (Linked Lists)

Write a ***recursive*** function that takes the head of a linked list (possibly NULL) that contains positive integers only. The function must return -1 if the list contains any integer that is equal to the sum of all integers that come after it in the list. If not, the function can return whatever value you feel is appropriate other than -1. (Figuring out what to return is part of the fun for this problem.)

For example, the function should return -1 for the following linked list because 4 is the sum of all the nodes that follow it (1, 2, and 1):

```
20 -> 3 -> 1 -> 4 -> 1 -> 2 -> 1 -> NULL

^
head
```

The function signature and node struct are:

```
typedef struct node {
    int data;
    struct node *next;
} node;

int listylist(node *head) {
```

```
}
```

4) (10 pts) DSN (Binary Trees)

Write a ***recursive*** function that takes the root of a binary tree (possibly NULL) and returns the sum of all the nodes that are left children in the tree. See the example below, which returns 15 + 49 = 64, since the only nodes that are left children anywhere in the tree are 15 and 49.

```
     28
    /  \
  15    22
    \   / \
    1  49  68
```

The node struct and function signature are:

```
typedef struct node {
    int data;
    struct node *left;
    struct node *right;
} node;

int add_all_left_children(node *root) {





}
```

5) (10 pts) ALG (Sorting)

a) (3 pts) The following diagram shows an initial array, followed by what the array looks like after a single pass of some sorting algorithm. Indicate what sorting algorithm is being applied, and give that algorithm's worst-case runtime using big-oh notation, for an array of size n.

22	49	36	22	17	18	4

4	49	36	22	17	18	22

Sorting algorithm being applied: ______________________

Worst-case runtime for algorithm: ______________________

b) (3 pts) For the following array, follow the same instructions from part (a):

84	19	23	66	91	44	42

19	23	66	84	44	42	91

Sorting algorithm being applied: ______________________

Worst-case runtime for algorithm: ______________________

c) (4 pts) Give a recurrence relation that represents the runtime for a Merge Sort of n items. Let T(n) represent the runtime of Merge Sort of n items in setting up your recurrence relation.

Computer Science Foundation Exam

August 12, 2016

Section II A

DISCRETE STRUCTURES

NO books, notes, or calculators may be used,

and you must work entirely on your own.

Name: ____

UCFID: ____

NID: ____

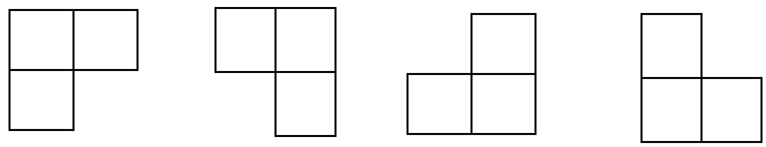
Question	Max Pts	Category	Passing	Score
1	15	PRF (Induction)	10	
2	15	PRF (Logic)	10	
3	10	PRF (Sets)	7	
4	10	NTH (Number Theory)	7	
ALL	50		34	

You must do all 4 problems in this section of the exam.

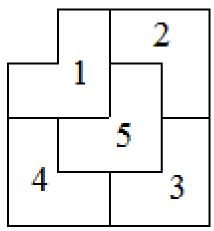
Problems will be graded based on the completeness of the solution steps and <u>not</u> graded based on the answer alone. Credit cannot be given unless all work is shown and is readable. Be complete, yet concise, and above all <u>be neat</u>.

1) (15 pts) PRF (Induction)

A tromino is a tile consisting of three unit squares in an L shape. The following are the four possible orientations a tromino can be placed:



Using induction on n, prove that for all non-negative integers, n, a 2^n x 2^n grid of unit squares with a single unit square removed can be tiled properly with a set of trominos. A proper tiling covers every unit square of the original object with a single unit square of a single tromino. For example, the following is a valid tiling of the 4 x 4 grid with the top left corner missing:



2) (15 pts) PRF (Logic)

(a) (8 pts) Complete the truth table below.

p	q	r	$p \wedge q$	$\bar{p} \vee r$	$\overline{\bar{p} \vee r}$	$(p \wedge q) \vee (\overline{\bar{p} \vee r})$
F	F	F				
F	F	T				
F	T	F				
F	T	T				
T	F	F				
T	F	T				
T	T	F				
T	F	T				

(b) (7 pts) Create a logical expression using the variables p and q and only the logical operators ($\wedge$) and ($\overline{\ }$) which evaluates as described by the truth table below. (Note: There are many correct answers and each variable and operator may appear in the expression you create as many times as necessary.)

p	q	result
F	F	F
F	T	T
T	F	T
T	T	F

3) (10 pts) PRF (Sets)

Let A, B and C be finite sets such that $A \subseteq B$, $B \subseteq A \cup C$, and $C \subseteq B$. Prove or disprove the following assertion: $|A| = |B|$ or $|A| = |C|$ or $|B| = |C|$.

4) (10 pts) NTH (Number Theory)

Find an integer, n, in between 0 and 231, inclusive, such that $105n \equiv 1 \pmod{232}$. (Note: To earn full credit you must use the Extended Euclidean Algorithm.)

Computer Science Foundation Exam

August 12, 2016

Section II B

DISCRETE STRUCTURES

NO books, notes, or calculators may be used,

and you must work entirely on your own.

Name: ______________________________

UCFID: ______________________________

NID: ______________________________

Question	Max Pts	Category	Passing	Score
1	10	CTG (Counting)	7	
2	10	PRB (Probability)	7	
3	15	PRF (Functions)	10	
4	15	PRF (Relations)	10	
ALL	50		34	

You must do all 4 problems in this section of the exam.

Problems will be graded based on the completeness of the solution steps and <u>not</u> graded based on the answer alone. Credit cannot be given unless all work is shown and is readable. Be complete, yet concise, and above all <u>be neat</u>.

1) (10 pts) CTG (Counting)

a) (6 pts) In an election with 142,070,000 eligible voters and only three candidates to choose from for some particular office, how many different distributions are possible for the number of votes each candidate could receive, provided that every eligible voter is forced to vote, and they must vote for one of the three candidates (so, the voters can't abstain from voting or choose some write-in candidate)?

For example, if candidate A receives 100,000,000 votes, candidate B receives 32,070,000 votes, and candidate C receives 10,000,000 votes, that is different from A receiving 32,070,000 votes, B receiving 100,000,000 votes, and C receiving 10,000,000 votes.

Note that votes are cast anonymously, so all that matters is the number of votes each candidate receives, with no consideration for which voters those votes came from.

b) (4 pts) What would be the answer to (a) if instead of voters being forced to vote, they were allowed to sit at home and not vote for any of the candidates? (But still, no write-in candidates are allowed on the ballot. Those who vote are constrained to the three candidates on the ballot.)

2) (10 pts) PRB (Probability)

Suppose six wizards are seated in a row along one side of a long, straight banquet table, in totally random order. Among those wizards are Lily Evans, James Potter, and Severus Snape.

Let P be the event that Lily Evans and James Potter end up sitting next to one another, and S the event that Severus Snape and Lily Evans end up sitting next to one another. Prove or disprove that P and S are independent events. (You may assume there are no magical shenanigans at play that would affect the probabilities of these events.)

3) (15 pts) PRF (Functions)

Let $f: \mathbb{Z} \rightarrow \mathbb{Z}$ and $g: \mathbb{Z} \rightarrow \mathbb{Z}$, where $f(x) = 5x + 10$ and $g(x) = 10x + 5$. Then:

(a) (3 pts) Give $f \circ g$.

(b) (4 pts) Prove or disprove that $f \circ g$ is surjective.

(c) (4 pts) Prove or disprove that $f \circ g$ is injective.

(d) (4 pts) For a function $h: \mathbb{Z} \rightarrow \mathbb{Z}$, use quantifiers to write a statement in symbolic logic that says h is a surjective function.

4) (15 pts) PRF (Relations)

(a) (3 pts) What three properties must a relation satisfy in order to be an equivalence relation?

(b) (6 pts) Is it possible to define an equivalence relation R on $A = \{1, 2, 3, 4, 5, 6, 7\}$ such that $|R|$ is even? If so, give one such equivalence relation. If not, *briefly* explain why not.

(c) (6 pts) Suppose we define a relation R by choosing 9 random ordered pairs (without replacement) from $A \times A$, where $A = \{1, 2, 3, 4, 5, 6, 7\}$. What is the probability that R will be an equivalence relation?