Computer Science Foundation Exam

December 18, 2015

Section I A

COMPUTER SCIENCE

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

Name: ____

UCFID: ____

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 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) DSN (Recursive Functions)

Write a ***recursive*** function hasProperty that takes the root of a ternary tree (a tree where each node has at most three children) and determines whether all the nodes that have a middle child also have both a left child and a right child. If so, return 1. Otherwise, return 0. Note: If the function is given a null input, it should return 1.

The node struct and functional prototype for this question are:

```
typedef struct node
{
   char *data;
   struct node *left, *middle, *right;
} node;

int hasProperty(node *root) {










}
```

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

Give the Big-Oh runtimes for each of the following functions in terms of *n* and/or *k* (where *k* is the length of string *s*), given that *strlen*(*s*) is an O(*k*) function and *toupper*(*c*) is an O(1) function. You may assume that *s* is non-NULL and contains at least one character. No justifications necessary, only answers will be graded.

```
void uppercase(char *s)
{
   int i;
   for (i = 0; i < strlen(s); i++)
      s[i] = toupper(i);
}
```

uppercase run time: ___________

```
void uppercase_remix(char *s)
{
   int i, length = strlen(s);
   for (i = 0; i < length; i++)
      s[i] = toupper(i);
}
```

uppercase_remix run time: __________

```
void uppercase_unreliable(char *s)
{
   int i = 0, j = strlen(s) - 1, m;

   while (i <= j)
   {
      m = i + (j - i) / 2;
      if (rand() % 2 == 0)
      {
         s[i] = toupper(s[i]);
         i = m + 1;
      }
      else
      {
         s[j] = toupper(s[j]);
         j = m - 1;
      }
   }
}
```

uppercase_unreliable run time: __________

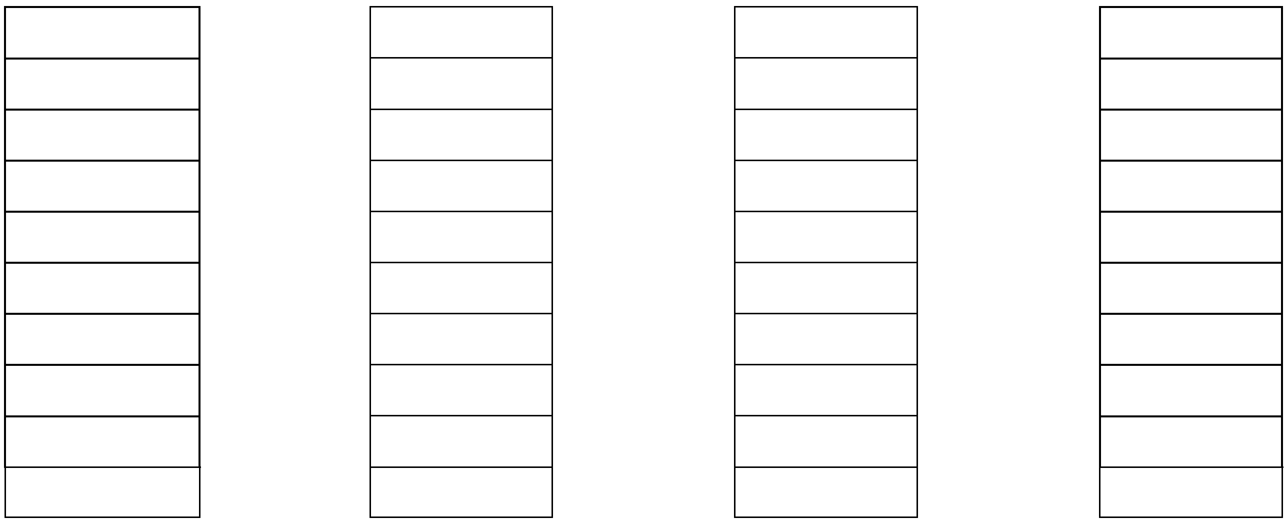
```
void mad_scramble(char *s, int n)
{
   int i;
   for (i = 0; i < n; i++)
      s[strlen(s) - 1] = rand() % 25 + 'a';
}
```

mad_scrable run time: __________

3) (10 pts) ALG (Stacks)

Use a stack to convert the following infix expression to a postfix expression. Please show the state of the stack at the exact point in time when the algorithm reaches the marked locations (A, B, C, and D) while processing the expression.

```
                         A                  B                  C                           D
8   *   3   /   2   +    9   -   6   +    (   3   -   (    8   *   2   )   +   7   )
```



A B C D

Equivalent Postfix Expression:

____

4) (10 pts) ALG (Binary Search Trees and Hash Tables)

a) (8 pts) Draw a **single** binary search tree that gives rise to all three of the following tree traversals:

Inorder: 4 7 8 10 27 30 44 56

Preorder: 4 10 7 8 44 30 27 56

Postorder: 8 7 27 30 56 44 10 4

b) (2 pts) If we insert an element into a hash table using quadratic probing to resolve collisions, what two conditions must be met to ensure that if an open spot exists in our hash table, we will find that spot (rather than getting stuck in an infinite loop)?

1. ____

2. ____

5) (10 pts) ALG (Base Conversion)

Write a function that takes a string *str* and an integer *b* (where $2 \leq b \leq 10$), and returns 1 if *str* represents an integer in base *b* that is a perfect power of *b*. For example:

```
isPower("323", 4);      // Return 0. 323₄ = 59₁₀, which is not a power of 4
isPower("27", 3);       // Return 0. 27 is not a valid base 3 integer.
isPower("plum", 8);     // Return 0. plum is not a valid base 8 integer.
isPower("1000", 10);    // Return 1. 1000₁₀ is a power of 10 (10³)
isPower("000001", 2);   // Return 1. 1₂ = 1₁₀, which is a power of 2 (2⁰)
```

Notes: You may assume *b* is always within the range specified above. Your function must return 0 if *str* is NULL or the empty string. Strings may be padded on the left with any number of zeros.

```
// You must use this function signature. You may write helper functions as
// needed.
int isPower(char *str, int b);
```

Computer Science Foundation Exam

December 18, 2015

Section I B

COMPUTER SCIENCE

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

Name: ____

UCFID: ____

Question #	Max Pts	Category	Passing	Score
1	10	ANL	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 <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) (10pts) ANL (Algorithm Analysis)

Consider the recursive function `sum` shown below:

```
double sum(int* array, int low, int high){
    if (low == high)
        return array[low];
    int mid = (low+high)/2, left = 0, right = 0, i;
    for (i=low; i<=mid; i++) left += array[i];
    for (i=mid+1; i<=high; i++) right += array[i];
    if (left > right) return left + sum(array, low, mid);
    return right + sum(array, mid+1, high);
}
```

(a) (3 pts) Let T(n) represent the run time of the function call `sum(array, 0, n-1)`, where array is an integer array of size n. Write a recurrence relation that T(n) satisfies.

(b) (7 pts) Using the iteration method, determine a closed-form solution (Big-Oh bound) for T(n). Assume T(1) = O(1).

2) (10 pts) ANL (Algorithm Analysis)

(a) (5 pts) A matrix factorization algorithm that is run on a input matrix of size n x n, runs in $O(n^3)$ time. If the algorithm takes 54 seconds to run for an input of size 3000 x 3000, how long will it take to run on an input of size 1000 x 1000?

(b) (5 pts) A string algorithm with inputs of lengths n and m runs in $O(n^2m)$ time. If the algorithm takes 2 seconds to run on an input with n = 1000 and m = 500, how long will the algorithm take to execute on an input with n = 250 and m = 1000?

3) (10 pts) DSN (Linked Lists)

Write a **recursive** function, `aboveThreshold`, that takes in a pointer to the front of a linked list storing integers, and an integer, `limit`, and returns the number of values stored in the linked list that are strictly greater than `limit`. For example, if the function was called on a list storing 3, 8, 8, 6, 7, 5, 7, 9 and `limit` equaled 6, then the function should return 5, since the 2nd, 3rd, 5th, 7th and 8th values in the list are strictly greater than 6. (Notice that we don't count the 4th element.)

Use the struct definition provided below.

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

int aboveThreshold(node* front, int limit) {

}
```

4) (10 pts) DSN (Binary Trees)

We define the ***offcenter value*** for each node in a binary tree as being the absolute value of the difference between the height of its left subtree and the height of its right subtree. For example, for an AVL tree, each node has an offcenter value of 0 or 1. Also, note that we define the offcenter value of a null node to be 0. Write a function, `maxOffCenterValue` that computes the maximum offcenter value of any node in a tree pointed to by root. To make your task easier, assume that the height of each node is stored in the corresponding struct for that node in the component `height`.

Using the struct definition and function max given below, complete the function in the space provided.

```
typedef struct treenode {
    int value;
    int height;
    struct treenode *left;
    struct treenode *right;
} treenode;

int max(int a, int b) {
    if (a > b) return a;
    return b;
}

int maxOffCenterValue(treenode* root) {





}
```

5) (10 pts) ALG (Sorting)

Write the code for any one of the following O(n^2) sorts: Bubble Sort, Insertion Sort, Selection Sort in a single function below. Your code should sort the array from smallest to largest. (Namely, after your code finishes array[i] ≤ array[i+1] for all i, 0 ≤ i < length-1.) Please provide the name of the sort you are choosing to implement and fill in the function prototype below, making sure not to add any additional functions.

Sort implemented below (circle choice): Bubble Sort Insertion Sort Selection Sort

```
void sort(int* array, int length) {









}
```

Computer Science Foundation Exam

December 18, 2015

Section II A

DISCRETE STRUCTURES

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

Name: ____

UCFID: ____

Question	Max Pts	Category	Passing	Score
1	15	PRF (Induction)	10	
2	15	PRF (Logic)	10	
3	10	PRF (Sets)	6	
4	10	NTH (Number Theory)	6	
ALL	50		32	

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 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) (15 pts) PRF (Induction)

Use mathematical induction to prove that, for every positive integer n,

$$5 \mid (8^n - 3^n)$$

2) (15 pts) PRF (Logic)

Validate the following argument using the laws of logic, substitution rules or rules of inference. List the rule used in each step and label the steps used in each derivation.

$$
\begin{array}{l}
(\neg q \lor n) \to r \\
q \\
z \to y \\
p \\
t \to \neg q \\
\hline
\therefore w \to ((p \lor \neg r) \land \neg t)
\end{array}
$$

3) (10 pts) PRF (Sets)

a) Let $C = \{x, y\}$. What is $C \times C$?

b) Let $D = \{p, o, m\}$. What is $\mathcal{P}(D)$ (the power set of D)?

c) Let A and B be finite sets. Then, give the following in terms of $|A|$ and $|B|$:

$|A \times B| =$

$|\mathcal{P}(A)| =$

$|\mathcal{P}(A \times B)| =$

d) Let A and B be finite sets. Give the following in terms of $|A|$, $|B|$, $|A \times B|$, and/or $|A \cap B|$. (Do not use $|A \cup B|$ in your answer.)

$|\mathcal{P}(A \cup B)| =$

e) Under what conditions does $A \times B$ equal to $B \times A$? (To receive full credit, your answer must cover all possible conditions where this occurs.)

4) (10 pts) NTH (Number Theory)

Show that for all integers a and b, if $30 \mid (14a + 8b)$, then $30 \mid (2a - 46b)$.

Computer Science Foundation Exam

December 18, 2015

Section II B

DISCRETE STRUCTURES

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

Name: ____

UCFID: ____

Question	Max Pts	Category	Passing	Score
1	15	CTG (Counting)	10	
2	15	PRB (Probability)	10	
3	10	PRF (Functions)	6	
4	10	PRF (Relations)	6	
ALL	50		32	

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) CTG (Counting)

Please leave your answers in factorials, permutations, combinations and powers. Do not calculate out the actual numerical value for any of the questions. Justify your answers.

Consider a string s of lowercase latin letters of length 10.

(a) (5 pts) Suppose we require s to have no two consecutive letters that are the same. How many such strings s exist?

(b) (5 pts) Suppose we require s to have each consecutive triplet of letters be pairwise distinct. How many such strings s exist?

(c) (5 pts) Suppose we require s to have no two consecutive letters that are the same but additionally require that s be a palindrome. (A string that reads the same forward or backwards.) How many such strings s exist?

2) (15 pts) PRB (Probability)

Suppose we roll a fair 4 sided die with the numbers [1,4] written on them. After the first die roll we roll the die k times where k is the number on the first die roll. The number of points you score is the sum of the face-values on all die rolls (including the first). What is the expected number of points you will score?

3) (10 pts) PRF (Functions)

The function $C(n,k)$ is known as the choose function.
Let $X = \{(n,k) | n \in \mathbb{N} \wedge k \in \mathbb{N} \wedge k \leq n\}$. (Note that ℕ is the non-negative integers.)
For the purposes of this problem we define the choose function, C, as follows:

$$C: X \to \mathbb{Z}^+, C(n,k) = \frac{n!}{k!\,(n-k)!}$$

(a) (5 pts) Is C injective? Prove or disprove this property.

(b) (5 pts) Is C surjective? Prove or disprove this property.

4) (10 pts) PRF (Relations)

Consider the relation $\mathcal{R}$ on $\mathbb{Z}^2$ where $\big((a,b),(c,d)\big) \in \mathcal{R}$ when $ad - bc \geq 0$. Determine (with proof) if $\mathcal{R}$ meets or doesn't meet each of these properties: reflexive, symmetric, anti-symmetric, transitive.