

Computer Science Foundation Exam

December 14, 2012

Section I A

COMPUTER SCIENCE

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

Question #	Max Pts	Category	Passing	Score
1	11	DSN	7	
2	10	ANL	7	
3	10	ALG	7	
4	9	ALG	6	
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) (11 pts) DSN (Recursion)

Write a **recursive** function that takes in a linked list and returns a pointer to the node with the highest value. Head, representing the head of the list, and max, representing the current maximal node, are parameters to the function. Your function should make use of the following struct node and function prototype:

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

```
struct node * maxNode(struct node * head, struct node * max) {
```

```
}
```

2) (10 pts) ANL (Summations)

Determine a **simplified**, closed-form solution for the following summation in terms of n . **You MUST show your work.**

$$\sum_{k=5}^{n+3} (3k + 4)$$

3) (10 pts) **Stack Applications.**

Convert the following infix expression to postfix. Show the contents of the stack at the indicated points (1, 2, and 3) in the infix expression.

$$A + B * C \quad \overset{1}{\quad} \quad / \quad (\quad (\quad D + E \quad \overset{2}{\quad} \quad) + F * \quad \overset{3}{\quad} \quad G)$$

1

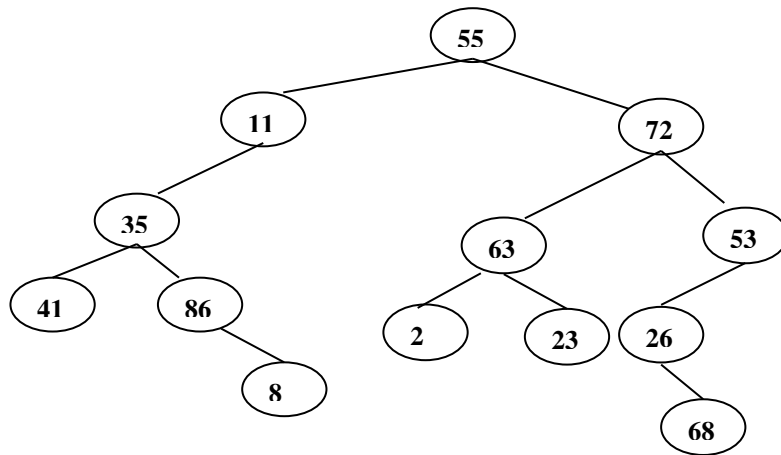
2

3

Resulting postfix expression:

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

4) (9 pts) ALG (Binary Trees)



Give the preorder, inorder, and postorder traversals of the binary tree shown above.

Preorder:

Inorder:

Postorder

5) (10 pts) ALG (AVL Trees)

Draw the resulting AVL tree after inserting the following items (in this order) into an initially empty AVL tree: 67, 24, 60, 72, 32, 26, 49. Show the tree after each step that requires a rebalance. (There are 2 of these steps) Show the final tree after all items have been added.

Computer Science Foundation Exam

December 14, 2012

Section I B

COMPUTER SCIENCE

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

Name: _____

PID: _____

Question #	Max Pts	Category	Passing	Score
1	10	ANL	7	
2	10	DSN	7	
3	10	DSN	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) ANL (Algorithm Analysis)

(a) (6 pts) List the best case, worst case and average case run-times of each of the following algorithms/operations in terms of their input size, n:

(i) Inserting an item into a Linked List of n elements

best case: _____ average case: _____ worst case: _____

(ii) A Quick Sort of n elements

best case: _____ average case: _____ worst case: _____

(iii) Searching for an element in a binary tree

best case: _____ average case: _____ worst case: _____

(b) (4 pts) Consider the recurrence relation $T(n) = T(n-1) + n2^n$, for $n > 1$ and $T(1) = 2$. Rewrite the value of $T(n)$ utilizing a summation so $T(n)$ is expressed without any reference to $T(x)$, for any value x . **Please leave your final answer as a summation and DO NOT ATTEMPT to solve the sum.**

2) (10 pts) DSN (Recursive Algorithms – Binary Trees)

Write a **recursive** function that will return the number of nodes in a binary tree that contain a particular value. Your function will take in a pointer to the root of the binary tree as well as the value for which to search. The prototype for the function and the binary tree struct are given to you below. Complete the function.

```
struct treeNode {  
    int data;  
    struct treeNode *left;  
    struct treeNode *right;  
};
```

```
int numOccurrences(struct treeNode* root, int value) {
```

}

3) (10 pts) DSN (Linked Lists)

Imagine using a linked list to store a large integer. In particular, each node of the linked list stores a single digit with the least significant digit being stored first. For example, the number 1387 would be stored in a linked list of length four storing the digits 7, 8, 3, and 1, respectively. Complete the function below that compares two large integers stored in this fashion. In particular, if the first number is strictly less than the second number, return -1. If they are equal, return 0. Otherwise, return 1. Although it's not a requirement, it's probably easiest to write this function recursively. Use the struct and function prototype provided below.

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

```
int intcmp(struct node *ptrA, struct node* ptrB) {
```

}

4) (10 pts) ALG (Tracing) Consider the following function:

```
int f(int array[], int length, int target) {  
    int i=0, j=0, sum=0, cnt=0;  
  
    while (j < length) {  
        if (sum < target) {  
            sum += array[j];  
            j++;  
        }  
        else if (sum > target) {  
            sum -= array[i];  
            i++;  
        }  
        else {  
            cnt++;  
            sum -= array[i];  
            i++;  
        }  
    }  
  
    if (sum == target) cnt++;  
  
    return cnt;  
}
```

(a) (3 pts) If array stores the elements 2, 3, 3, 2, 5, 4, 1, 3, 6, 8, 2, 3, 4, 4, 2, 2, what would the return value of the function call $f(\text{array}, 16, 8)$ be?

(b) (4 pts) Give a concise description of what f calculates.

(c) (3 pts) Let n be the length of the input array to f . What is the run time of f in terms of n ?

5) (10 pts) ALG (Sorting)

(a) (4 pts) Consider running a Merge Sort on the array below. What would the contents of the array be right BEFORE the last Merge operation?

Index	0	1	2	3	4	5	6	7
Values	13	2	8	7	14	6	19	1

Index	0	1	2	3	4	5	6	7
Values								

(b) (3 pts) Consider running an insertion sort on the array below. How many swaps would be performed total while the algorithm ran?

Index	0	1	2	3	4	5	6	7
Values	13	2	8	7	14	6	19	1

(c) (3 pts) Given the array below, which element, 8 or 83 would be a better pivot element for running the Partition in Quick Sort? Why? (Note: By better pivot element, we mean a choice of pivot that's likely to reduce the run time of the algorithm.)

Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Values	13	2	8	7	14	6	19	1	99	5	57	44	3	31	83