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.

SOLUTION

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) {

     if (head == NULL)                           //2 points
        return max;

     if (max == NULL)                            //2 points
        max = head;

     if (head->data > max->data)                 //2 points
        max = head;                              //1 point

     max = maxNode(head->next, max);             //3 points

     return max;                                 //1 point

}
```

2) (10 pts) ANL (Summations)

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

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

$= [\sum_{k=1}^{n+3} 3k + 4] - [\sum_{k=1}^{4} 3k + 4]$ //2 points for changing limits

$$= \left[\sum_{k=1}^{n+3} 3k + 4\right] - \left[3\sum_{k=1}^{4} k + \sum_{k=1}^{4} 4\right]$$

$$= \left[\sum_{k=1}^{n+3} 3k + 4\right] - \left[3\sum_{k=1}^{4} k + 16\right]$$

$$= \left[\sum_{k=1}^{n+3} 3k + 4\right] - \left[3\left(\frac{4*5}{2}\right) + 16\right]$$

$= [\sum_{k=1}^{n+3} 3k + 4] - 46$ //3 points for reducing this half to 46

$= [3\sum_{k=1}^{n+3} k + \sum_{k=1}^{n+3} 4] - 46$ //2 points for separating the constants

$= [3\sum_{k=1}^{n+3} k] + [4(n+3)] - 46$ //1 point for multiplying by n+3

$= 3\left[\frac{(n+3)(n+4)}{2}\right] + [4(n+3)] - 46$ //1 point for substituting (n)(n+1) / 2

= (3/2)[(n+3)(n+4)] + 4(n+3) – 46 //1 point for algebraic simplification
= (3/2)(n^2 + 3n + 4n + 12] + 4n + 12 – 46
= (3/2)(n^2 + 7n + 12) + 4n – 34
= (3/2)n^2 + (21/2)n + 18 + 4n – 34
= (3/2)n^2 + (29/2)n – 16

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 (1) **/ ((D + E** (2) **) + F *** (3) **G)**

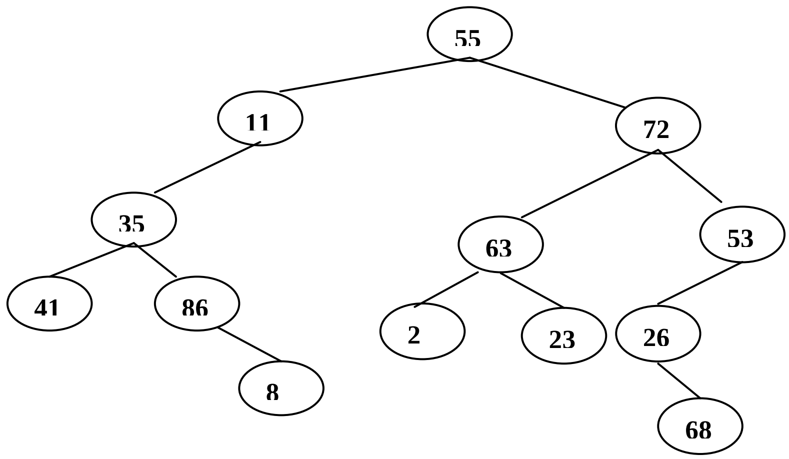
1 (1 point)	2 (3 points)	3 (2 points)
	+	*
	(	+
	(	(
*	/	/
+	+	+

Resulting postfix expression:

A	B	C	*	D	E	+	F	G	*	+	/	+							

Grading: 4 points for the whole expression (partial credit allowed.)
Points for stacks are marked individually.

4) (9 pts) ALG (Binary Trees)



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

Preorder:

55, 11, 35, 41, 86, 8, 72, 63, 2, 23, 53, 26, 68

Inorder:

41, 35, 86, 8, 11, 55, 2, 63, 23, 72, 26, 68, 53

Postorder

41, 8, 86, 35, 11, 2, 23, 63, 68, 26, 53, 72, 55

Grading: 3 points per traversal (partial credit allowed.)
If two traversals are switched (ex: the preorder traversal is labeled inorder) take off 3 points total. If all three are switched, take off 6 points total.

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

After first rebalance (3 pts):

```
    60
    / \
  24    67
```

(1 pt for root, 1 pt for 24, 1 pt for 67)

After second rebalance(4 pts):

```
      60
      / \
    26    67
   / \      \
 24  32      72
```

(1 pt for root, 1 pt for 26, 2 pts for rest)

Final Tree:

```
      60
      / \
    26    67
   / \      \
 24  32      72
       \
       49
```

(1 pt for root, 1 pt for left, subtree, 1 pt for right subtree)