Computer Science Foundation Exam

August 10, 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	8	
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 **<u>recursive</u>** function that removes targeted nodes from a linked list pointed to by head. Head and the target value are parameters to the function. You should return the head of the list after all nodes with the target value have been removed. Your function should make use of the following `struct node` and function prototype:

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

struct node * remove_target(struct node *head, int target) {

    if (head == NULL) return NULL;      // 2 pts

    if (head->data == target) {         // 1 pt
        struct node* rest = head->next;   // 1 pt
        free(head);                       // 1 pts
        return remove_target(rest, target); // 2 pts
    }

    head->next = remove_target(head->next, target); // 3 pts

    return head;                                  // 1 pt









}
```

2) (10 pts) ANL (Summations)

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

$$\sum_{k=4}^{2n+2} 2k - 2$$

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

$$= \left[\sum_{k=1}^{2n+2} 2k - 2\right] - \left[2\sum_{k=1}^{3} k - \sum_{k=1}^{3} 2\right]$$

$$= \left[\sum_{k=1}^{2n+2} 2k - 2\right] - \left[2\sum_{k=1}^{3} k - 6\right]$$

$$= \left[\sum_{k=1}^{2n+2} 2k - 2\right] - \left[2\left(\frac{3 * 4}{2}\right) - 6\right]$$

$= [\sum_{k=1}^{2n+2} 2k - 2] - 6$ **//1 point for reducing this half to 6**

$= [2\sum_{k=1}^{2n+2} k - \sum_{k=1}^{2n+2} 2] - 6$ **//2 points for separating the constants**

$= [2\sum_{k=1}^{2n+2} k] - [2(2n + 2)] - 6$ **//1 point for multiplying by 2n+2**

$= 2\left[\frac{(2n+2)(2n+3)}{2}\right] - [2(2n + 2)] - 6$ **//2 points for substituting (n)(n+1) / 2**

$= (2n+2)(2n+3) – 2(2n+2) – 6$ **//2 points for algebraic simplification**
$= 4n^2 + 4n + 6n + 6 – 4n – 4 – 6$
$= 4n^2 + 6n – 4$

Note: Students may solve this differently and still get credit. One example of such a method is using the formula for an arithmetic series. Award points accordingly. (2 pts for first term, 2 pts for last term, 2 pts for number of terms, 4 pts for algebraic simplification.) Do not give full credit to guess and check methods, or any method that wouldn't work on an arbitrary summation of this form in a reasonable amount of time.

3) (10 pts) ALG (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.

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

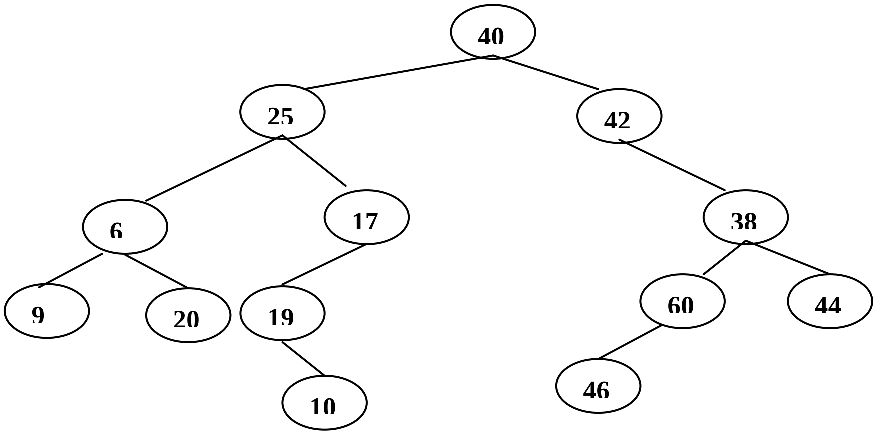
(		
*		
(		
-	+	(
(	(	/
1	2	3

Resulting postfix expression:

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

Grading: 2 points for each stack, 4 points for the whole expression (partial credit allowed.) Take off one point for one distinct error, even if it causes two items to be wrong in this criteria.

4) (9 pts) ALG (Binary Trees)



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

Preorder:

40, 25, 6, 9, 20, 17, 19, 10, 42, 38, 60, 46, 44

Inorder:

9, 6, 20, 25, 19, 10, 17, 40, 42, 46, 60, 38, 44

Postorder

9, 20, 6, 10, 19, 17, 25, 46, 60, 44, 38, 42, 40

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)

(a) (7 pts) Draw the resulting AVL tree after inserting the following items (in this order) into an initially empty AVL tree: 80, 35, 16, 42, and 50. Show the tree after each step that requires a rebalance. (There are 2 of these steps)

```
After first rebalance (3 pts):       35                  (1 pt for root, 1 pt for 16,
                                     / \                  1 pt for 80)
                                   16    80

After second rebalance(4 pts):       35                  (1 pt for root, 1 pt for 16,
                                     / \                  1 pt for 50, 1 pt for rest)
                                   16    50
                                         / \
                                       42    80
```

(b) (3 pts) Delete the node that has 16 as a data value from the AVL tree from part a. Show the state of the AVL tree after any necessary rebalancing.

```
After rebalance:                     50                  (1 pt for root, 1 pt for left
                                     / \                  subtree, 1 pt for right
                                   35    80               subtree)
                                     \
                                     42

After rebalance:                     42                  (1 pt for root, 1 pt for left
                                     / \                  subtree, 1 pt for right
                                   35    50               subtree)
                                           \
                                           80
```

Note: Accept either of the two answers listed above.