Computer Science Foundation Exam

May 3, 2013

Section I A SOLUTION

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	10	DSN	7	
2	10	ANL	7	
3	10	ALG	7	
4	10	ALG	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) DSN (Recursion)

Write a **recursive** function that takes in a linked list and returns a pointer to the node where the data is divisible by three. You are guaranteed that at most one node will have data that is divisible by three. If no nodes are divisible by three, return NULL. Head, representing the head of the list, is a parameter 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 * divThree(struct node * head) {

     if (head == NULL)
          return NULL;

     if (head->data % 3 == 0)
          return head;

     return divThree(head->next);
}
```

Grading: **1 points checking for null**
1 point returning null when appropriate
2 points checking for divisibility
2 points returning the correct node
3 points correct recursive call
1 point for final return

2) (10 pts) ANL (Summations)

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

$$\sum_{k=3}^{n+1} (4k - 2)$$

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

$$= \left[\sum_{k=1}^{n+1} 4k - 2\right] - \left[4\sum_{k=1}^{2} k - \sum_{k=1}^{2} 2\right]$$

$$= \left[\sum_{k=1}^{n+1} 4k - 2\right] - \left[4\sum_{k=1}^{2} k - 4\right]$$

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

$= [\sum_{k=1}^{n+1} 4k - 2] - 8$ //3 points for reducing this half to 8

$= [4\sum_{k=1}^{n+1} k - \sum_{k=1}^{n+1} 2] - 8$ //2 points for separating the constants

$= [4\sum_{k=1}^{n+1} k] - [2(n+1)] - 8$ //1 point for multiplying by n+1

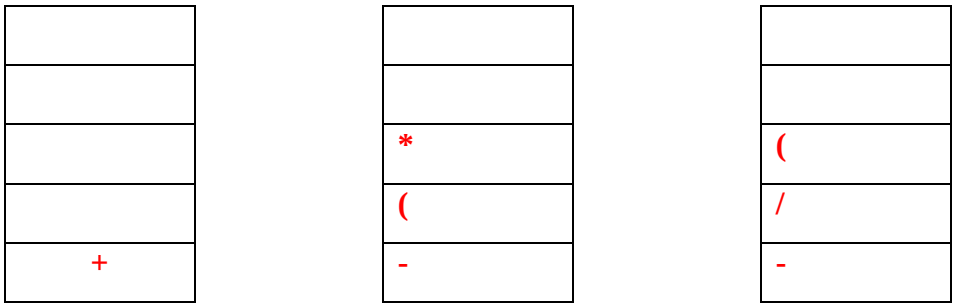
$= 4\left[\frac{(n+1)(n+2)}{2}\right] - [2(n+1)] - 8$ //1 point for substituting (n)(n+1) / 2

= 2(n+1)(n+2) – 2(n+1) – 8 //1 point for algebraic simplification
= $2n^2$ + 4n + 2n + 4 – 2n – 2 – 8
= $2n^2$ + 4n – 6

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.

(A * B - C) * D + 1 **E – (F *** 2 **G) / (H** 3 **+ I)**



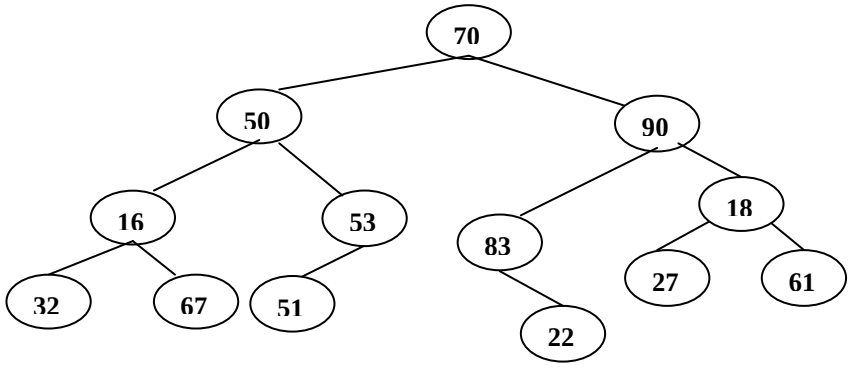
1 2 3

Resulting postfix expression:

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

Grading: 2 points for each stack, 4 points for the whole expression (partial credit allowed.)

4) (10 pts) ALG (Binary Trees)



How many leaf nodes does the tree above have? **6 (1 pt)**

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

Preorder:

70, 50, 16, 32, 67, 53, 51, 90, 83, 22, 18, 27, 61

Inorder:

32, 16, 67, 50, 51, 53, 70, 83, 22, 90, 27, 18, 61

Postorder

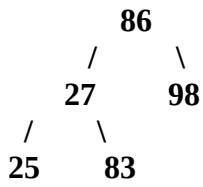
32, 67, 16, 51, 53, 50, 22, 83, 27, 61, 18, 90, 70

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)

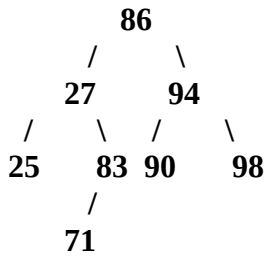
Draw the resulting AVL tree after inserting the following items (in this order) into an initially empty AVL tree: 86, 25, 98, 83, 27, 90, 71, 94. Show the tree after each step that requires a rebalance. (There are 2 of these steps) Show the final tree after all items have been added.

After first rebalance



1 pt each (5 total)

After second rebalance



1 pt root
2 pts left subtree
2 pts right subtree