Computer Science Foundation Exam

August 12, 2011

Section I A

COMPUTER SCIENCE

SOLUTION

**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			

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) (10 points) **Recursion.** Write a **recursive** function that operates on a binary tree of integers. Give two binary trees (as input), your function should check to see if those two trees are structurally identical (they are made of nodes with the same values and arranged in the same way). If the two trees are structurally identical, return 1; otherwise, return 0. Your function should make use of the following `struct tree_node` and function prototype:

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

int areIdentical(struct tree_node *a, struct tree_node *b) {

     // 1. IF both trees are empty, return 1
     // Grading:  2 points

     if (a==NULL && b==NULL)
          return(1);

     // 2. ELSE, if both trees are non-empty, compare them:
     // Grading:  6 points

     else if (a!=NULL && b!=NULL) {
          return(a->data == b->data &&
               areIdentical (a->left, b->left) &&
               areIdentical (a->right, b->right));
     }

     // 3. ELSE, if one tree is null and the other not, return 0
     // Grading:  2 points

     else
          return(0);


}
```

2) (10 points) **Summations.** Determine a **simplified**, closed-form solution for the following summation in terms of n. **You MUST show your work.**

$$\sum_{j=n-10}^{n}\left(4\sum_{i=1}^{3n}2j\right)$$

$$\sum_{j=n-10}^{n}\left(4\sum_{i=1}^{3n}2j\right)=\sum_{j=n-10}^{n}\left(8j\sum_{i=1}^{3n}1\right)=\sum_{j=n-10}^{n}8j*3n=\sum_{j=n-10}^{n}24jn$$

Grading: 4 points for solving inner summation

$$\sum_{j=n-10}^{n}24j=24\sum_{j=n-10}^{n}j=24\left(\sum_{j=1}^{n}j-\sum_{j=1}^{n-11}j\right)$$

Grading: 2 points for changing limits

$$\begin{aligned}
\mathbf{24n}\left(\sum_{j=1}^{n}j-\sum_{j=1}^{n-11}j\right)&=\mathbf{24n}\left(\frac{n(n+1)}{2}-\frac{(n-11)(n-10)}{2}\right)\\
&=\mathbf{12n(n^2+n-(n^2-21n+110))}\\
&=\mathbf{12n(n^2+n-n^2+21n-110)}\\
&=\mathbf{12n(22n-110)}\ \textit{\textbf{(accepted final form)}}\\
&=\mathbf{264n(n-5)}\ \textit{\textbf{(accepted final form)}}\\
&=\mathbf{264n^2-1320n}\ \textit{\textbf{(accepted final form)}}
\end{aligned}$$

Grading: 2 points for applying the sum formulas, 2 points for algebraic simplification into one of the given forms.

Note: Students may avoid using the specific summation formula (sum of i from i=1 to n) by using the arithmetic series sum formula. Please award full credit if they solve the problem this way, or any valid method except brute force.

3) (10 points) **Queues 'n Stacks.** Let `q` be a queue and `s` be a stack. The functions `dequeue` and `pop` obey the convention that they return whatever they remove. Assume that `q` and `s` are initially empty and that `i` has been declared as an `int`. <u>What would be **printed** by the following code fragment?</u> (put answer in the box)

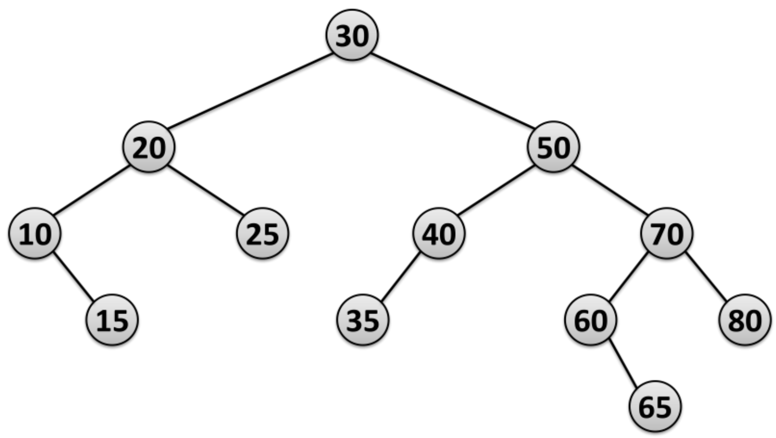
```
enqueue(q, 8);
enqueue(q, 3);
push(s, 7);
push(s, 9)
for(i = 0; i < 5; i++){
     printf("%d  ", dequeue(q));
     printf("%d  ", pop(s));
     enqueue(q, i);
     push(s, i+5);
}
```

8 9 3 5 0 6 1 7 2 8

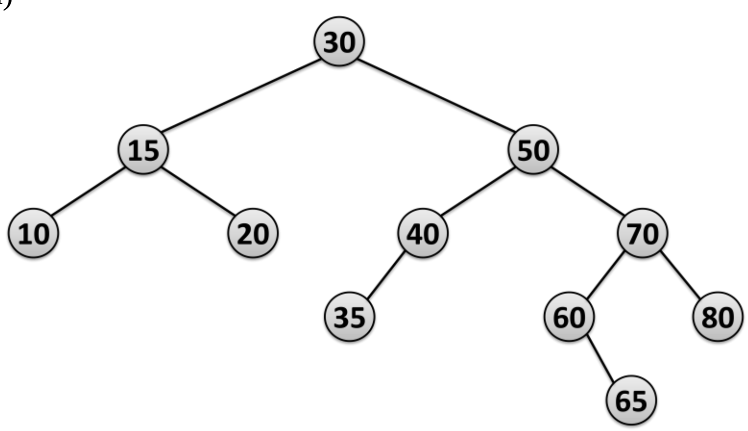
Grading: 1 pt for each correct number in the output

4) (10 points) **AVL Trees.** The tree shown below is a valid AVL tree. You must **delete** node 25 (the node that has 25 as a data value). Then you must rebalance the AVL tree as needed, which will require two restructures.

(a) Show the state of the AVL tree after the first rebalancing.
(b) Show the state of the AVL tree after the second rebalancing.

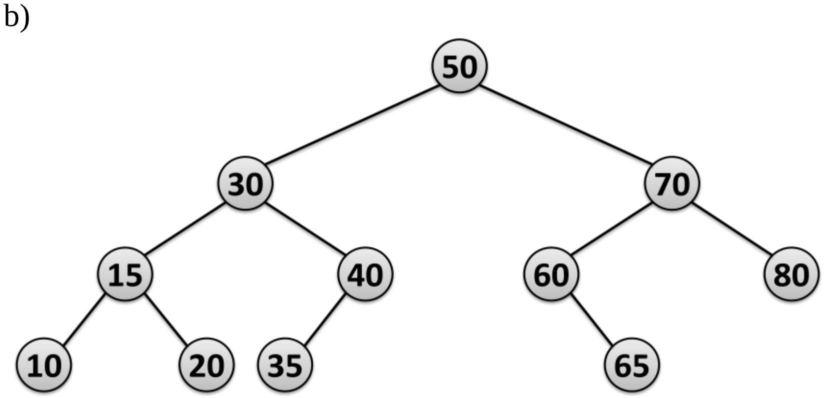


a)



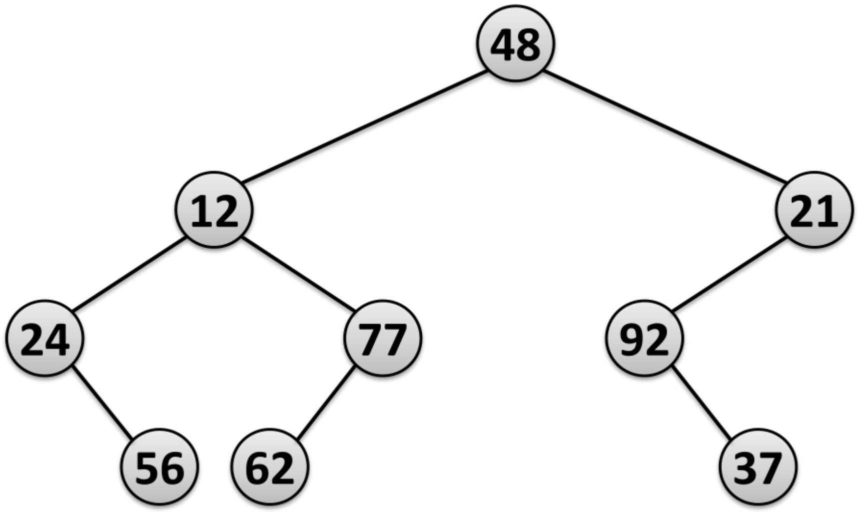
Grading: 1 point for deleting node 25. 2 points for correct rotation of nodes 10, 15, and 20.

b)



Grading: 1 pt for 50 at the root, 1 pt for 30, 1 pt for 70, 1 pt for each sub tree.

5) (10 points) **Binary Tree Traversals**



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

Preorder:

48, 12, 24, 56, 77, 62, 21, 92, 37 (3 points)

Inorder:

24, 56, 12, 62, 77, 48, 92, 37, 21 (4 points)

Postorder:

56, 24, 62, 77, 12, 37, 92, 21, 48 (3 points)

Grading note: If the traversals are correct but the names switched in any way, just take off 3 points. Decide partial credit based on the number of "changes" you have to make to their answer to get to the correct one by inserting one portion of the list into a different portion of the list.