Computer Science Foundation Exam

August 14, 2009

Computer Science

Section 1A

Name: __________________________

PID: ____________________________

<table>
<thead>
<tr>
<th>Max Pts</th>
<th>Type</th>
<th>Passing Threshold</th>
<th>Student Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1 11</td>
<td>DSN</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>Q2 10</td>
<td>ANL</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>Q3 10</td>
<td>ALG</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>Q4 10</td>
<td>ALG</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>Q5 9</td>
<td>ALG</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>Total 50</td>
<td></td>
<td>35</td>
<td></td>
</tr>
</tbody>
</table>

You must do all 5 problems in this section of the exam.

Partial credit cannot be given unless all work is shown and is readable. Be complete, yet concise, and above all be neat. Do your rough work on the last page.
1) (11 points) **Recursion** Write a recursive function that prints the contents of a linked list in reverse order. Make use of the list node struct and function header below.

```c
struct listnode {
    int data;
    struct listnode* next;
};

void print_reverse(struct listnode* head) {
    Solution:
    if(head == NULL)
        return;

    print_reverse(head->next);
    printf("%d ", head->data);
}
```

**Grading Criteria:**
There are many ways to approach this problem. Be reasonable when grading.
Base case – 3 points
Printing the node in the proper place in the function – 4 points
Making a proper recursive call – 4 points
2) (10 points) **Summations**

a) Consider the following code fragment:

```c
prod = 0;
for(i = 0; i <= n + 7; i++) {
    prod = prod * 5 * i;
    for(j = i - 5; j <= i + 5; j++) {
        prod = prod * n * j * 3;
    }
}
```

Write, but don't solve, a summation to describe the number of multiplications performed by that code fragment in terms of the variable \( n \).

b) Obtain a simplified closed form solution for the following summation:

\[
\sum_{i=1}^{n} \left( 3 \sum_{j=1}^{i} 2ij \right)
\]

Solution:

a) 
\[
\sum_{i=0}^{n-7} \left( 2 + \sum_{j=i-5}^{i+5} 3 \right)
\]

b) 
\[
\sum_{i=1}^{n} \left( 3 \sum_{j=1}^{i} 2ij \right) = \sum_{i=1}^{n} \left( 6i \sum_{j=1}^{i} j \right) = \sum_{i=1}^{n} \left( 6i \frac{n(n+1)}{2} \right) = 6 \frac{n(n+1)}{2} \sum_{i=1}^{n} i = 6 \frac{n(n+1)}{2} \frac{n(n+1)}{2} = \frac{3n^2(n+1)^2}{2}
\]

**Grading Criteria:**

a) 
Correct bounds on the outer summation – 2 points
Correct bounds on the inner summation – 2 points
Answer is otherwise correct – 1 point

b) 
Properly dealing with the inner summation – 2 points
Properly dealing with the outer summation – 2 points
Simplifying the resulting closed form – 1 point
3) (10 points) **Stack Applications** Transform the following infix expression into its equivalent postfix expression using a stack. Show the contents of the stack at the indicated points 1, 2 and 3 in the infix expressions.

\[
( ( A + B ) / C ) - D * ( E / F - G )
\]

**Solution:**

- Point 1:
  - Stack: \[
  \]
  - Contents: \[
  \]
- Point 2:
  - Stack: \[
  \]
  - Contents: \[
  \]
- Point 3:
  - Stack: \[
  \]
  - Contents: \[
  \]

**Resulting postfix expression:**

\[
A \ B + \ C / \ D \ E \ F / \ G \ - \ * \ -
\]

**Grading Criteria:**
- Each correct stack – 2 points
- Resulting expression – 4 points
4) (10 points) **AVL Trees** Consider the AVL tree below:

![AVL Tree Diagram]

**a)** Show the state of the tree after node containing the value 60 is inserted. Be sure to perform any necessary rotations.

**b)** Show the state of the tree after node containing the value 25 is inserted into the original tree (i.e. ignore part a when answering this part). Be sure to perform any necessary rotations.

---

**Solution:**

**a)**

![Inserted Node 60 Diagram]

**b)**

![Inserted Node 25 Diagram]

**Grading Criteria:**

5 points per part:

- Preserving BST order property – 2 points
- Maintaining AVL balance properties – 1 point
- Rotating correctly – 2 points

Note: Showing the pre-rotation tree is not required for a correct answer
5) (9 points) **Binary Tree Traversals**

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

**Solution:**

**Preorder:**
63  47  86  32  95  16  9  53  64

**Inorder:**
86  47  95  32  63  16  53  9  64

**Postorder:**
86  95  32  47  53  64  9  16  63

**Grading Criteria:**
3 points per traversal