

Computer Science Foundation Exam

August 10, 2007

Computer Science

Section 1A

Name: Solution and Grading Criteria

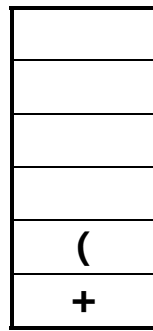
SSN: _____

	Max Pts	Passing Threshold	Type	Student Score
Q1	10	6	KNW	
Q2	10	6	CMP	
Q3	14	10	KNW	
Q4	10	7	KNW	
Q5	6	3	CMP	
Total	50	32		

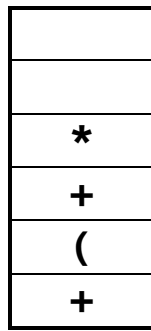
You must do all 5 problems in this section of the exam. Partial credit cannot be given unless all work is shown and is readable. Be complete, yet concise, and above all be neat.

1. [10 pts] Convert the following infix expression into its equivalent postfix expression using a stack. Show the contents of the stack, AND the partial postfix expressions at the indicated points in the infix expression (points 1, 2 and 3). The stack boxes should just show the appropriate characters and *NOTHING ELSE*. You may draw another stack alongside for your work.

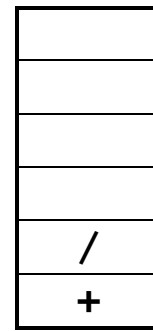
$$B * C + (\overset{1}{A / G + E * \overset{2}{D}}) / \overset{3}{F}$$



1



2



3

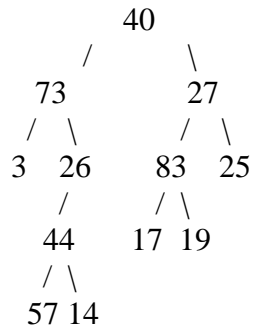
Indicate the partial postfix expression as obtained at various points indicated in the infix expression. The form of your answer is illustrated below for a hypothetical case:

Paritil Expression upto 1	Paritil Expression upto 2	Paritil Expression upto 3	Remaining part
M R+	B P / + D - *	T * S L+	N -

Paritil Expression upto 1	Paritil Expression upto 2	Paritil Expression upto 3	Remaining part
B C *	A G / E	D * +	F / +

Grading: 1 pt for each stack, 2 pts for each partial expression 1-3, 1 pt for remaining part

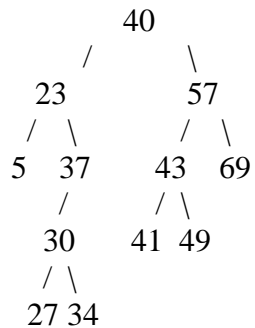
2. a) [5pts] Write the pre-order traversal of the following binary tree.



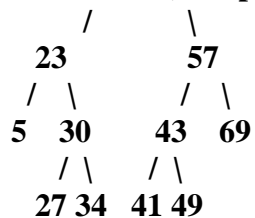
Answer : 40, 73, 3, 26, 44, 57, 14, 27, 83, 17, 19, 25

Grading: 3 pts for either inorder or postorder, otherwise, 1 pt for 2 correct, 2 pts for 5 correct, 3 pts for 7 correct, 4 pts for 10 correct

b) [5 pts] It is desired to delete node 40 from the following BST. Make corrections on this tree to show the new values.



37 (One possibility – the other is placing 41 at the root)



Grading: Deleting a different physical node – 2 pts, Picking 37 or 41 – 2 pts, 1 pt for rest of the structure

3. a) [4 pts] Which of the following is closest to the maximum number of comparisons an Insertion Sort (that is efficiently written) will make when sorting an array of 1000 elements? Justify your choice.

- a) 1000 b) 10000 c) **500000** d) 10000000 e) 100000000

The maximum number of comparisons is equal to $\sum_{i=1}^{999} i = \frac{999 \times 1000}{2} = 499500$, since for the first inserted element, at most one comparison can be made, for the second one 2, etc. until you are inserting the 999th element. The closest choice is C.

Grading: 0 pts if the answer is incorrect, 2 pts for the answer, 2 pts for the justification.

b) [4pts + 6pts] Solve for each summation in terms of n.

$$\sum_{j=10}^n 2j = \sum_{j=1}^n 2j - \sum_{j=1}^9 2j = n(n+1) - 90 = n^2 + n - 90$$

Grading: 1 pt split sum, 2 pts for sum from 1 to n, 1 pt for 1 to 9

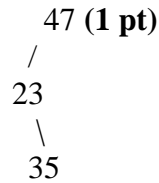
$$\begin{aligned} \sum_{j=1}^{2n} \sum_{m=12}^{3j+5} 2 &= \sum_{j=1}^{2n} 2(3j+5-12+1) = \sum_{j=1}^{2n} (6j-12) \\ &= \sum_{j=1}^{2n} 6j - \sum_{j=1}^{2n} 12 = 6n(2n+1) - 24n = 12n^2 - 18n \end{aligned}$$

Grading: 2 pts inner sum, 1 pt split, 2 pts sum of 6j, 1 pt sum of 12

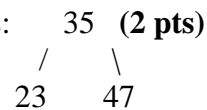
4. [10 pts] Draw an AVL tree as the following integers are inserted in the order they appear in the sequence. If the tree becomes unbalanced, redraw the tree alongside after balancing.

47, 23, 35, 16, 5, 20

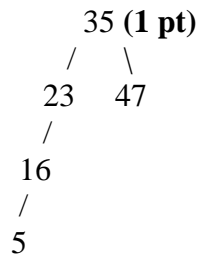
First three inserts lead to:



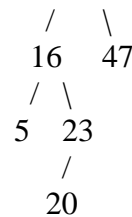
This causes an imbalance at 47 and gets restructured as:



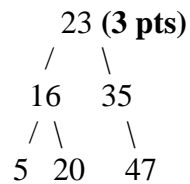
The next two inserts create:



This causes an imbalance at 23, so we restructure, **35 (3 pts – 2 for rebal, 1 pt 20)**
 Then inserting 20 causes:



This causes an imbalance at 35, which gets restructured as follows:



5. [6 pts] The nodes of a linked list p have the following structure

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

p contains the elements 66, 9, 14, 52, 87, 14 and 17, in that order. Consider running the following line of code:

```
p = question5(p);
```

where question5 is the function defined below. Show the contents of p *after* the function call.

```
struct node* question5(struct node *list) {

    struct node* a = list;
    struct node* b = list;
    struct node* c;

    if (a == NULL) return NULL;

    while ( a->next != NULL)
        a = a ->next;
    a->next = b;
    c = b->next;
    b->next = NULL;
    return c;
}
```

p->9->14->52->87->14->17->66

Grading: 1 pt for keeping all of the original elements in the list, 2 pts for not having 66 at the front, 2 pts for keeping 9, 14, 52, 87, 14 and 17 consecutive, 1 pt for attaching 66 at the end