Department of Computer Science

CS 201 – Data Structures

Mid Term II (Fall 2013)

Instructors: Adnan Waheed & Rabia Maqsood

November 04, 2013

Total Marks: 40	**Time Allowed: 70 minutes**

Instructions:

(1) Understanding the question is part of exam. NO QUERIES WILL BE ENTERTAINED.
(2) Provide answers in the given space.
(3) Use answer sheet for rough work only.
(4) Write neat & clean.
(5) Use permanent ink pens only.
(6) Poor programming approaches will decrease your marks.
(7) Think about the boundary conditions.

Roll No. _______________Name:_______________________________________Section: _________

Question No.	**1**	**2**	**3**	**4**	**5**	**6**	**Total**
Marks	***05***	***09***	***06***	***09***	***09***	***02***	***40***

GOOD LUCK ☺

Question 1:	**Marks 05**

Write a function **Siblings** which will display the sibling of any given node in a BST, root of the tree is also passed to this function. **You cannot use any other helper function or data structure for this problem.**

Question 2:	**Marks 09**

NOTE: Specify all helper functions/algorithms which you may use.

Part – I (For Sections A & B ONLY)

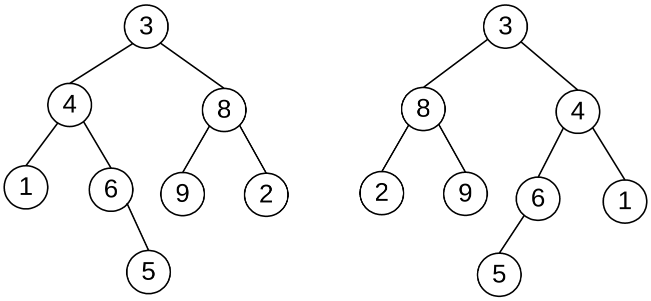
Write an algorithm to traverse a tree (starting from given node) in inorder. Do not use recursive approach.

Part – II (For Sections C & D ONLY)

Write an algorithm to make an insertion in AVL tree.

Question 3:	**Marks 06**

Given two binary trees, you have to determine if one is a mirror image of the other. Write a **recursive** function **bool mirror (Node* first, Node* second)** that returns true if the two trees pointed to by **first** and **second** are mirror images of each other and returns false otherwise. For example, following two binary trees are mirror images of each other. **You cannot use any other helper function or data structure for this problem.**



Question 4:	**Marks 09**

Part – I (For Sections A & B ONLY) **(2+1+1+1+1+1+2)**

Postorder traversal of a tree is as follows.

11 23 30 29 22 49 47 61 64 62 59 69 56

Use this information to answer the questions below. No marks will be given for partial answer.

a. Draw the tree *(bring clarity in your diagram; no overlapping should be there in different nodes and/or levels).*

b. What is the maximum level number of the tree?

c. What nodes are on Level 3?

d. Which levels have the maximum number of nodes that they could contain?

e. Is the tree a binary search tree (BST)? Justify your answer.

f. Trace the shortest path of the tree.

g. What is the maximum height of a binary search tree containing these nodes? Draw such a tree.

Part – II (For Sections C & D ONLY) **(2+1+1+1.5+1.5+2)**

We want to store the following data in a hash table with 11 size, in the order of their appearance.

25 96 42 223 112 12 84 102 153

Use division method for hash function and quadratic probing technique to resolve collision.

a. Draw and fill the hash table with the given data by using the above information.

b. How many comparisons are necessary to locate the record whose key value is 112?

c. How many comparisons are necessary to locate the record whose key value is 102?

d. How many comparisons are necessary to determine that the record whose key value is 14 is not in the table?

e. What happens if you remove the record whose key value is 223 from the table by just setting the field back to empty?

f. Draw and fill the hash table with the given data; by using rehashing to resolve collision. Use the last digit of your Roll# as a constant value in rehash function.

Question 5:	**Marks 09**

Part – I (For Sections A & B ONLY) **(4+5)**

a. Implement ***equalsBST*** boolean method which compares the given two BSTs. Two BSTs are equal if they contain the same elements at the same positions in the tree.

b. Convert the following expression to postfix notation. Show the state of stack and expression after each step.

NOTE: *if any step is incorrect, rest of the steps will not be checked.*

((2 * (6 + 3)) / (4 - 6 + 4)) / 5

Part – II (For Sections C & D ONLY) **(3 each)**

a. One hundred integer elements are chosen at random and inserted into a sorted linked list and a binary search tree. Describe the efficiency of searching for an element in each data structure, in terms of Big-O notation.

b. One hundred integer elements are inserted in order, from smallest to largest, into a sorted linked list and a binary search tree. Describe the efficiency of searching for an element in each data structure, in terms of Big-O notation.

c. Write a recursive boolean function ***isBST*** that determines whether a binary tree is a binary search tree.

Question 6:	**Marks 02 (1+1)**

a. Based on your competency level in each particular course, what points do you assign to yourself? Tick the appropriate box.

Course Title	**Points *(1 is minimum and 10 is maximum value)***									
CS 101- Introduction to Computing	1	2	3	4	5	6	7	8	9	10
CS 104- Computer Programming	1	2	3	4	5	6	7	8	9	10
CS 201- Data Structures	1	2	3	4	5	6	7	8	9	10

b. What factors may contribute to improve your learning in "CS 201 – Data Structures" course?