FINAL EXAM
CS 201 DATA STRUCTURES
[Computer Science Department, FASTNU, LAHORE Campus]

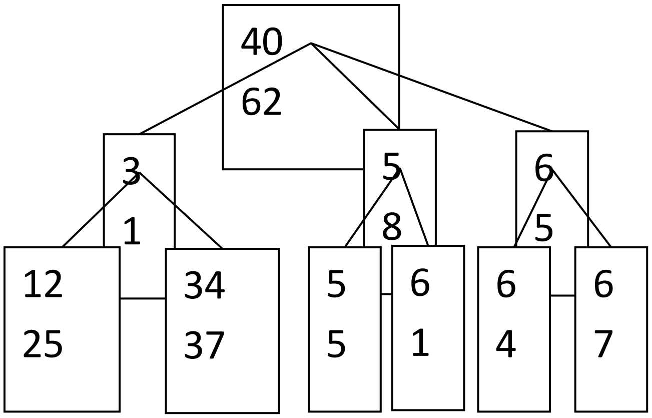
[Wednesday 26th December 2012.] **[Time**: 3 hrs] [**Total Marks**: 80]

Note: You are not allowed to ask any questions during the exam.
Code should be clear and well documented. **Marks will be deducted for poor quality code**

GOOD LUCK

QUESTION 1 **(5+5 MARKS)**

Delete 62 and 55 from the following 2-3 tree using successor (where required)



QUESTION 2 **(10 MARKS)**

Consider a min-heap containing 15 elements. Assume that all elements in the heap are distinct. The 1st smallest element in the heap must appear at the root (i.e. position 1 in the array). Similarly, the 2nd smallest element must appear at either position 2 or position 3 (the two children of the root). In each of the following, give a list of positions where:

a) The 3rd smallest element can appear.
b) The 4th smallest element can appear.
c) The 5th smallest element can appear.
d) The largest element can appear.

QUESTION 3 **(10 MARKS)**

You have the following data structures available to you: single and two dimensional arrays, linked-list, doubly linked-list, queue, stack, binary tree, binary search tree, heap, a balanced search tree (2-3), hash table, and directed and undirected graphs. For each of the scenarios given below, suggest the most appropriate data structure chosen from the above list.

a) to store a set of programs which are to be given access to a hard disk according to their priority.
b) for representing Lahore region telephone network.
c) to store a set of fixed key words which are referenced very frequently.
d) to represent an image in the form of a bit map.
e) to implement "back" functionality in the internet browser.
f) to store dynamically growing data which is accessed very frequently, based upon a key value.
g) to implement printer spooler so that jobs can be printed in the order of their arrival.

h) to record the sequence of all the pages browsed in one session.
i) to implement the undo function in a text editor
j) to store information about the directories and files in a system.

QUESTION 4 (10 MARKS)

Given the following In-Order and Post-Order traversals of a binary tree, construct the tree.

In-Order: ***1 2 3 4 5 6 7 8 9 10***
Post-Order: ***1 3 5 4 2 7 10 9 8 6***

QUESTION 5 (10 MARKS)

Write a C++ code to delete an element at index ***i*** from a max heap.

QUESTION 6 (15 MARKS)

Write a C++ class, **Stack**, to implement an array based stack that can store only ***n*** elements. If more than n elements are pushed, the stack only remembers the last n pushed elements.

For example, if the stack contains, from top to bottom {4, 3, 2, 1} i.e. **4** is on top of stack and **1** is at the bottom. If 5 is pushed, the stack will contain {5, 4, 3, 2}. Similarly after pushing 6, the contents of stack will be {6, 5, 4, 3}

The Time Complexity of Push and Pop operations is O(1). No marks, if the Time Complexities of these operations are not O(1).

Note: You may not write **copy constructor** and **assignment operator**.

QUESTION 7 (15 MARKS)

Write a recursive function to generate the deep copy of the link list.

class Node
{
        Node *next;
        int Data;
        friend class LinkedList;
};
	class LinkedList
	{
	public:
	        LinkedList(); //implemented
	        ~LinkedList(); //implemented
	        LinkedList(const LinkedList &L); // **you have to implement**
	
	        //add any other method you like but implement it too
	        // **You shall not overload assignment operator**
	
	private:
	        Node *head;
	};