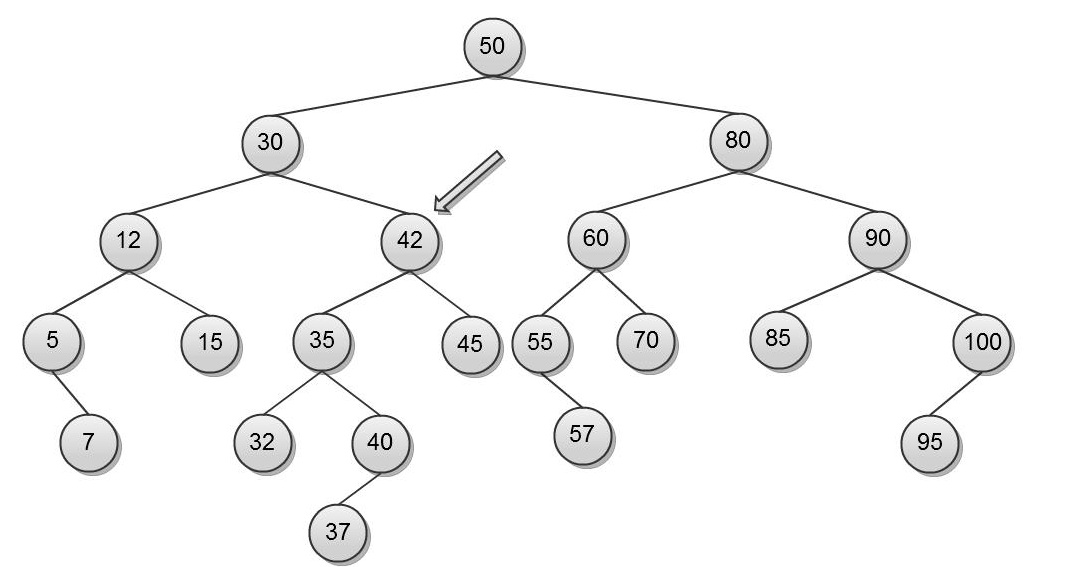
**Final Examination**

Data Structures (Summer 20011)

Time: **3 hours**

**Q#1** (20 marks) Take an empty 2-3 Tree and perform the following insertions in the given order: 80, 40, 60, 90, 70, 85, 100, 20, 110, 30, 95, 15. Now that you have a filled tree, perform the following deletions in it through the given sequence: 40, 80, 85. You must redraw the tree after every insertion/deletion operation. You may not indicate individual rotations and recombinations.

**Q#2** (10 marks) Write the code to balance the following AVL-Tree using standard AVL-Tree balancing rotations. You are given the pointer to the node whose balance factor has been disturbed. 

**Q#3** (10 marks) Given the Min-Heap Data-structure, write the code for the update function and calculate its complexity. The update function takes the address of a node in the heap and an integer value ‘x’ as input. It changes the value of that node to ‘x’ and re-adjusts the heap accordingly.

**Q#4** (15 marks) Given two character arrays, both of size ‘n’, find their intersection, i.e., return an array containing only the elements which exist in both the original arrays. In order to get full marks on this, your algorithm should not take more than O(n) time. However, an algorithm that takes O(n log n) would get 2/3 marks. Whereas, an algorithm taking O(n2) or more time will not be accepted. For this problem, you can use any data structure without giving its implementation.

**Q#5** (20 marks) Give short and comprehensive answers to the following questions. You must give reason for your answer. Without it, the answer carries no credit.

* Calculating the Height of any binary search tree takes how much time?
* Calculating the exact Height of any almost balanced binary search tree takes how much time?
* Searching in a Balanced Binary Tree takes how much time in the worst case?
* Searching in a Binary Search Tree takes how much time in the worst case scenario?
* Calculating the height of any binary heap takes how much time?

**Q#6** (15 marks)Given a graph (of neighboring countries in form of an Adjacency Matrix), Write an algorithm that, given two nodes, finds whether there exists any path between the two nodes. Give the time complexity of your algorithm.

**Q#7** (10 marks) Write the Program that takes the root of a Binary Search Tree and returns (root of) another Binary Search Tree which should be an exact copy of the original tree.

**Q#8** (20 marks) Write a function that takes two Binary Search Tree roots as input parameters and returns a sorted singly link list head which contains all the elements of both the original Binary Search Trees in sorted order. You can assume that all the elements are distinct. Also, your algorithm should take no more than O(n) time to finish.