# CSC 6580 –Design and Analysis of Algorithms

## Midterm 2

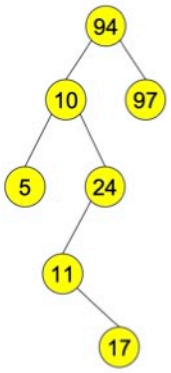
**Time:** 11:30am—12:45pm. Your solution may be hand-written and scanned. But please upload a single PDF file in Canvas. You are given 15 minutes to upload your exam. **You must upload by 1:00pm**. For every minute delay after 1:00pm, your obtained score will be deducted by 1.

Total points: 100

**Question 1 (10+10=20 points)**

(i) Write a modified version of Inorder traversal for a binary search tree that will print the keys in reversed sorted order (that is in non-increasing order).

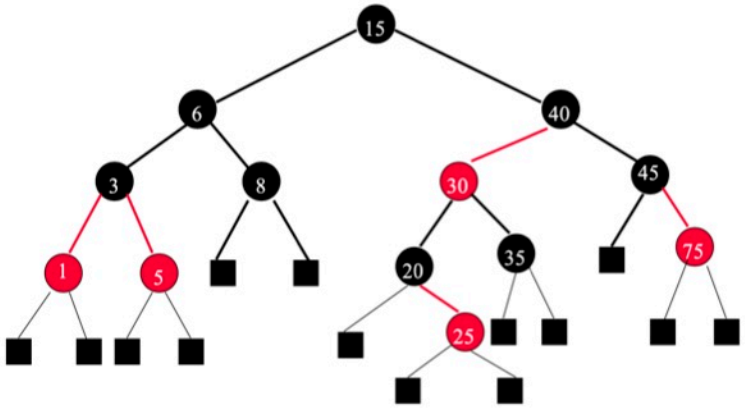
(ii) From the following binary search tree, delete the node with key 10. Show every step.




**Question 2 (10+10=20 points)**

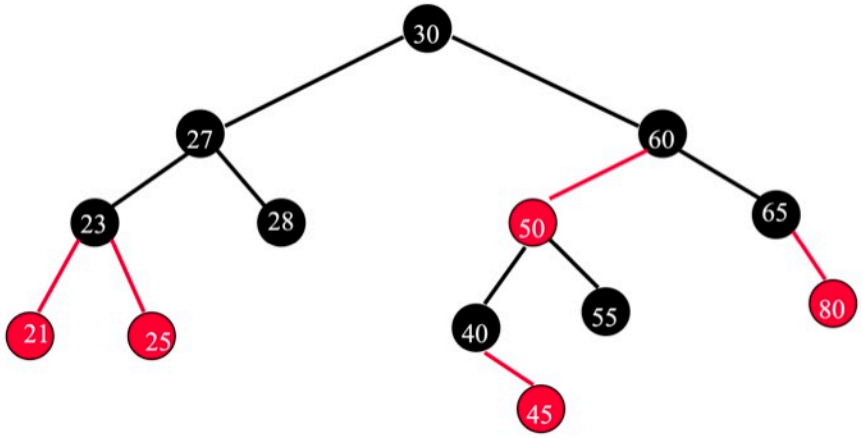
The figures in this problem show red-black trees whose red nodes are colored red while the black nodes are colored **black**. When you draw the figures to answer the following questions you must mark which nodes are black and which ones are red.

(i) Using the following Red-Black tree, show that a red-black is a clever representation 2-3-4 tree (i.e., convert the following red-black tree to a 2-3-4 tree).



# CSC 6580 –Design and Analysis of Algorithms

(ii) From the following red-black tree (whose external nodes are not shown), delete the node with key 65.




**Question 3 (5+25=30 points)**

(i) Using the optimal substructure property of the ***longest common subsequence (LCS)*** problem, write a recursive form of the length of the LCS between two sequences of characters.

(ii) Using dynamic programming, construct the table to determine the length of LCS and to determine the LCS between the following two sequences of characters.

YPVPFYP
PFVYPY

**Question 4 (20+5+5=30 points)**

Consider the five-symbol alphabet {A, B, C, D, _ } with the following occurrence frequencies (normalized within a scale of 1) in a text made up of these symbols:

| symbol | A | B | C | D | _ |
|---|---|---|---|---|---|
| frequency | 0.35 | 0.1 | 0.2 | 0.2 | 0.15 |

(i) Construct a Huffman coding tree (show step by step).

(ii) Show the resulting codewords.

(iii) Determine the percentage of size reduction to store the texts using Huffman code compared to fixed length coding.