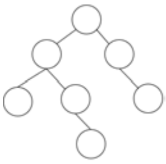
**CS203 DSAA Spring 2019**

**Student ID:** ______________________

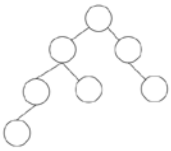
**Student Name:** ___________________

(1) Which of the following is a complete binary tree (  ) [5] C

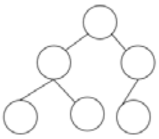
(A)



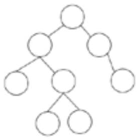
(B)



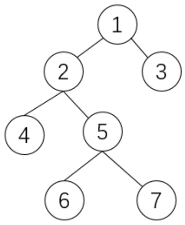
(C)



(D)



(2) Which of the following is the post order traversal of the given binary tree (  ) [5] B



(A) 4567231

(B) 4675231

(C) 1231567

(D) 4265713

(3) Suppose that a binary's pre-order traversal is 1 2 3 4 and post-order traversal is 4 3 2 1, its in-order traversal would not be ( ) [5] C

(A) 1 2 3 4

(B) 2 3 4 1

(C) 3 2 4 1

(D) 4 3 2 1

(4) if the following array represents a complete binary tree, which is the left child of 32 ( ) [5] B

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 23 | 41 | 32 | 17 | 26 | 38 | 71 | 29 | 32 | 21 |

(A) 26

(B) 38

(C) 71

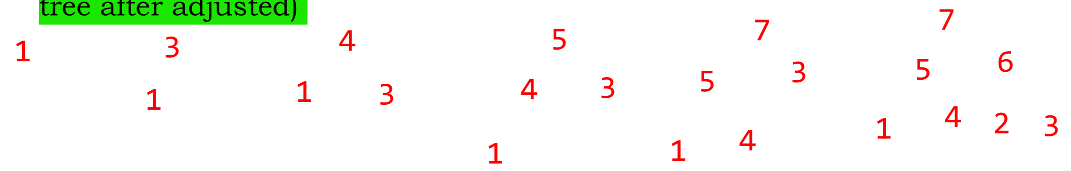
(D) 29

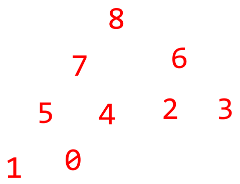
**CS203 DSAA Spring 2019**

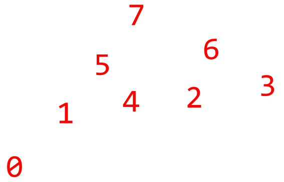
**Problem 1 [Heap, 20]**

(1) Create a max root heap of the given sequence 1,3,4,5,7,2,6,8,0 . Use insertion method please.(plot the tree of heap)

(2) Delete the max element of the heap and adjust the structure (plot the tree after adjusted)



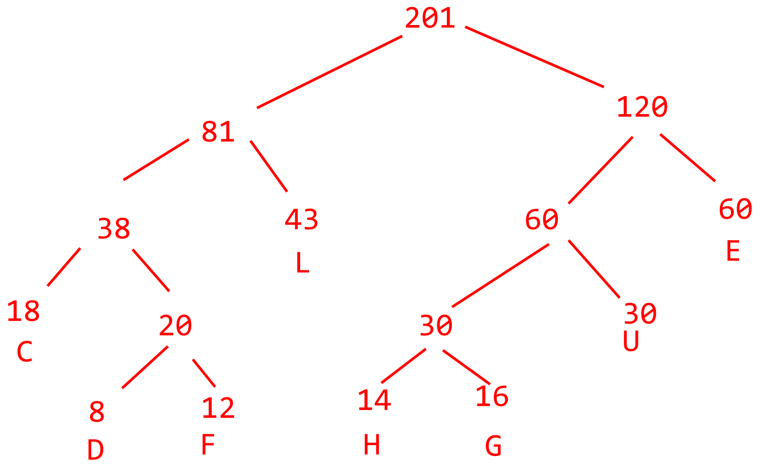




**Problem 2 [Huffman Encoding, 30]**

| F | G | D | H | U | C | L | E |
|---|---|---|---|---|---|---|---|
| 12 | 16 | 8 | 14 | 30 | 18 | 43 | 60 |

(1) Given the table recording the frequency of each character as above, please lot its corresponding Huffman tree.



(2) Please Write down the corresponding encodings as given by the Huffman tree in (1) by filling the following table

| F | G | D | H | U | C | L | E |
|---|---|---|---|---|---|---|---|
| 0011 | 1001 | 0010 | 1000 | 101 | 000 | 01 | 11 |

(3) Write down the corresponding codes of given string "FGGHC" according to the encodings in (2)

0011 1001 1001 1000 000

**Problem 3 [Algorithm Design, 20]**

Given a non-empty binary tree, how to find the path having the maximum weight sum?

For this problem, a path is defined as any sequence of nodes from some starting node to any node in the tree along the parent-child connections. The path must contain at least one node and does not need to go through the root. The weight sum of any path is the sum of the weights of the nodes along it.

Please describe your basic idea briefly and provide the pseudo code.

```
/**
 * Definition for a binary tree node.
 * public class TreeNode {
 *     int val;
 *     TreeNode left;
 *     TreeNode right;
 *     TreeNode(int x) { val = x; }
 * }
 */
```

Basic idea:

```
root's subtree's maxweight =
max{left's subtree's maxweight,right's subtree's maxweight,
left subtree's maxweight from leftsubtree's root to leaf +root.weight+right subtree's maxweight from rightsubtree's
root to leaf  }
root’s subtree's maxweight from leftsubtree's root to leaf=
max{left subtree's maxweight from leftsubtree's root to leaf,
    right subtree's maxweight from rightsubtree's root to leaf}
```

```
class Solution {
   public int maxPathSum(TreeNode root) {



   }
}
```

**Problem 4 [Remedy Imbalance Node, 30]**

Let us define a binary tree node as following:

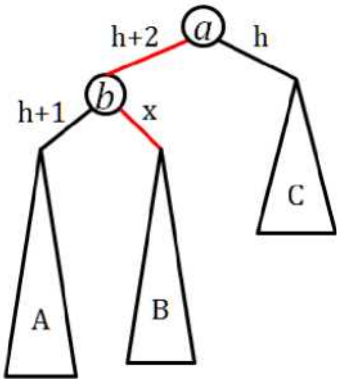


Figure 1 Left-left case

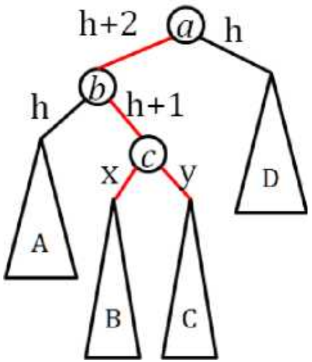


Figure 2 Left-right case

1) Given imbalance node a as Figure 1, after remedied the imbalance node a,

a->leftchild = B , a->rightchild = C

b->leftchild = A , b->rightchild = a

2) Given imbalance node a as Figure 2, after remedied the imbalance node a,

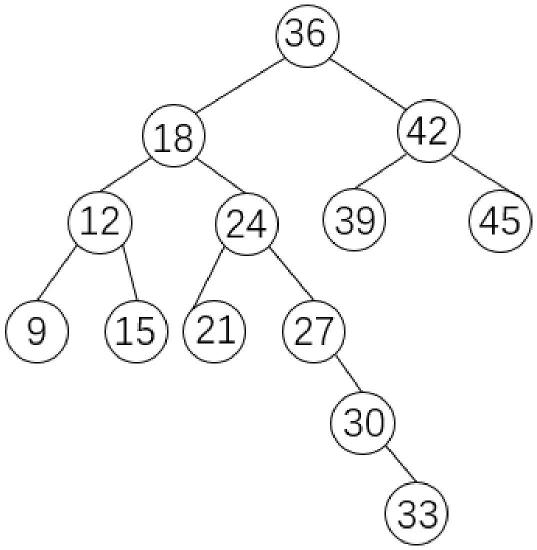
a->leftchild = C , a->rightchild = D

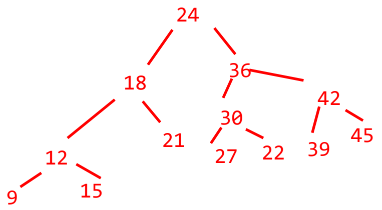
b->leftchild = A , b->rightchild = B

c->leftchild = b , c->rightchild = a

3) Draw the corresponding balanced tree of Figure 1 and Figure 2, respectively.

4) Given the following imbalance tree, please draw the balanced tree after remedy


36
18
42
12
24
39
45
9
15
21
27
30
33


24
18
36
42
12
21
30
9
15
27
22
39
45