Student ID (SID):
Student's Name:

CSCE-4263/5183 Advanced Data Structures
Spring 2022

Midterm Exam (Sample)

Date: Wednesday, Mar. 30, 2022

Time: 8:35 a.m – 9:25 a.m

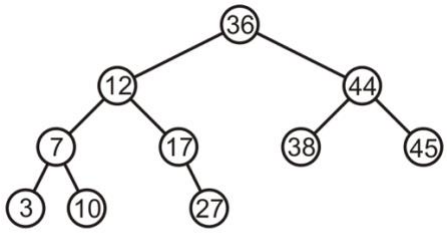
Instructions

1. It is an individual exam.
2. You have 50 minutes. You may not leave during the last 10 minutes of the exam.
3. Do NOT open exams until told to. Write your SIDs in the top left corner of every page.
4. If you need to go to the bathroom, bring us your exam, phone, and SID. We will record the time.
5. In the interest of fairness, we want everyone to have access to the same information. To that end, we will not be answering questions about the content.
6. The exam is open book.
7. Mark your answers - ON THE EXAM IN THE ANSWER AREAS. We will not grade anything on scratch paper.

Student ID (SID):
Student's Name:

Question 1: Draw all AVL trees with 1, 2, 3, and 4 nodes (10 points)

Question 2: This tree is AVL balanced or not? (5 points)



Question 3:

1. Insert node 23 into the AVL tree in Question 2 (5 points)
2. Is this updated tree AVL balance or not? (5 points)
3. If not, please rebalance it. (10 points)

Question 4: Draw all Red-Black trees with 1, 2, 3, and 4 nodes: (10 points)

Question 5: Programming questions

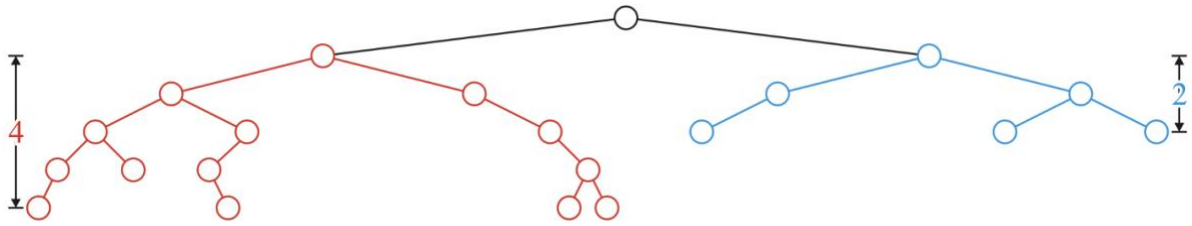
```
class Binary_node {
    protected:
        int node_value;
        Binary_node *p_left_tree;
        Binary_node *p_right_tree;
    public:
```

```
        Binary_node( Type const & );

        int value() const;
        Binary_node *left() const;
        Binary_node *right() const;
        bool is_leaf() const;
        int size() const;
        int height() const;
        void clear();
};

bool Binary_node::is_leaf() const {
        return (left() == nullptr) && (right() == nullptr);
}

int Binary_node::height() const {
    if ( left() == nullptr ) {
        return ( right() == nullptr ) ? 0 : 1 + right()->height();
    } else {
        return ( right() == nullptr ) ?
        1 + left()->height() :
        1 + left()->height() + right()->height();
    }
```



1. Write the function to count the number of leaves in a binary tree. (25 points)

2. Write the functions to delete the highest leaf in a binary tree. If there is more than one, you can delete any of them. (30 points)