Lab 7A

Red Black Tree

In this lab session, we will continue to learn and implement an advanced tree data structure using a simple color-coding scheme to adjust the tree after each modification: **Red-Black Tree**. Source code needs to be distributed in a single file named **StudentID_1.cpp**.

Each node in the Red-Black Tree has the following basic structure:

```
1 enum Color
2 {
3     RED,
4     BLACK
5 };
6
7 struct Node
8 {
9     int key;
10     Color color;
11     Node* left;
12     Node* right;
13     Node* parent;
14 };
```

Implement these basic functions and operations on the Red-Black Tree as follows:

1. Create a new Node with a given value.

```
Node* newNode(int data)
```

2. Insert a new value into a Red-Black Tree.

```
Node* insert(Node* root, int data)
```

3. Search for a Node with a given value in the Red-Black Tree. Return NULL if not found.

```
Node* search(Node* root, int data)
```

4. Delete a Node with a given value from the Red-Black Tree.

```
Node* deleteNode(Node* root, int data)
```

5. Traversal in Level-order.

```
void LevelOrder(Node* root)
```

Lab 7B

B Tree

In this lab session, we will learn and implement an advanced tree data structure: the **B-Tree**. Source code needs to be distributed in a single file named **StudentID_2.cpp**.

Each node in the B-Tree has the following basic structure:

Implement the following basic functions and operations for the B-Tree:

1. Create a new node.

```
Node* newNode(bool isLeaf)
```

2. Insert a new value into the B-Tree.

```
Node* insert(Node* root, int data)
```

3. Search for a node containing a given value in the B-Tree. Return NULL if not found.

```
Node* search(Node* root, int data)
```

4. Delete a node containing a given value from the B-Tree.

```
Node* deleteNode(Node* root, int data)
```

5. Traversal in Pre-order, In-order, Post-order and Level-order.

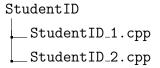
```
void NLR(Node* root)
void LNR(Node* root)
void LRN(Node* root)
void LevelOrder(Node* root)
```

Regulations

Please follow these regulations:

- You are allowed to use any IDE.
- After completing assignment, check your submission before and after uploading to Moodle.
- Prohibited libraries: <set>, <unordered_set>, <map>, <unordered_map>, <algorithm>, and
 <bits/stdc++.h>.
- You can use <vector> or any libraries that are not in the prohibited libraries listed above.

Your source code must be contributed in the form of a compressed file and named your submission according to the format StudentID.zip. Here is a detail of the directory organization:



The end.