Exercise #2 Spec

(Due Date: 2024/12/10 12:00)

Problems

In this exercise, you need to implement a red-black tree, including the following two operations:

1. Insert a node
2. Delete a node

Note: The red-black tree properties should hold after each operation.

Properties of red-black tree (p. 308)

1. Every node is either red or black.
2. The root is black.
3. Every leaf (NIL) is black.
4. If a node is red, then both its children are black.
5. For each node, all simple paths from the node to descendant leaves contain the same number of black nodes.

Input

(!!!) Your program needs to read the input.txt file to obtain the test cases.

The first line denotes that there are **t** tasks in the following inputs. For each task, the first two inputs **op**, **n** denote that this task is either **inserting elements (op = 1)** or **deleting elements (op = 2)**, and the number of the elements being operated is **n**. The next line contains **n** elements to be operated.

Output

(!!!) Your program needs to output an output.txt file to reserve your results.

Your red-black tree needs to satisfy the **5 properties** of a red-black tree as described above.

Please output the

(1) value, (2) parent node, (3) left child node, (4) right child node and (5) color of each node.

The output should obey the following rules:

1. Display the tree in **level-order traversal**.
2. Format: **"Node: {}, Parent: {}, Left: {}, Right: {}, Color: {}"**
3. If the parent or child node is nil (leaf node/root node's parent), just output the string **"nil"**.
4. No need to output leaf nodes (nil).
5. There is a newline character after each task.

Sample Input

```
3
1 8
5 11 9 7 6 12 4 1
2 2
11 5
1 2
2 3
```

Sample Output

```
Insert: 5, 11, 9, 7, 6, 12, 4, 1
Node: 9, Parent: nil, Left: 6, Right: 11, Color: Black
Node: 6, Parent: 9, Left: 4, Right: 7, Color: Red
Node: 11, Parent: 9, Left: nil, Right: 12, Color: Black
Node: 4, Parent: 6, Left: 1, Right: 5, Color: Black
Node: 7, Parent: 6, Left: nil, Right: nil, Color: Black
Node: 12, Parent: 11, Left: nil, Right: nil, Color: Red
Node: 1, Parent: 4, Left: nil, Right: nil, Color: Red
Node: 5, Parent: 4, Left: nil, Right: nil, Color: Red

Delete: 11, 5
Node: 9, Parent: nil, Left: 6, Right: 12, Color: Black
Node: 6, Parent: 9, Left: 4, Right: 7, Color: Red
Node: 12, Parent: 9, Left: nil, Right: nil, Color: Black
Node: 4, Parent: 6, Left: 1, Right: nil, Color: Black
Node: 7, Parent: 6, Left: nil, Right: nil, Color: Black
Node: 1, Parent: 4, Left: nil, Right: nil, Color: Red

Insert: 2, 3
Node: 6, Parent: nil, Left: 2, Right: 9, Color: Black
Node: 2, Parent: 6, Left: 1, Right: 4, Color: Red
Node: 9, Parent: 6, Left: 7, Right: 12, Color: Red
Node: 1, Parent: 2, Left: nil, Right: nil, Color: Black
Node: 4, Parent: 2, Left: 3, Right: nil, Color: Black
Node: 7, Parent: 9, Left: nil, Right: nil, Color: Black
Node: 12, Parent: 9, Left: nil, Right: nil, Color: Black
Node: 3, Parent: 4, Left: nil, Right: nil, Color: Red
```

(*You can check out the format of input and output in input.txt and output.txt files.)

Requirements

1. All the datasets will always insert nodes first.
2. All the datasets will not delete non-existent elements.
3. 0 < t <= 15, 0 < n <= 20.
4. The data type of every element is **Int32**.
5. You only need to output one of the trees that does not violate the red-black tree rules.
6. You can only include standard header files.
7. You cannot use any sorting algorithm.

Submission Policy

Language: C or C++

(Please check your program can be compiled successfully by gcc/g++)

(0 pts for other languages)

(do not include bits/stdc++.h)

Submission

- **Due date: 2024/12/10 12:00**

- **Submit: please submit your homework to E3 in ONE zip file with the following format.** Your submission should only include the two files: a source code and a report.

  (Please notice the following format for submission!)

  - **Exercise2_STUDENT_ID.zip**

    |- Exercise2_STUDENT_ID.cpp/.c

    |- Report.pdf
- **Source code**
  - You should name your file as **Exercise2_STUDENT_ID.cpp/.c**.
- **Report**
  - Environment (OS, compiler version, IDE)
    - How to compile and run your program
  - Results
    - Methods
    - Analyze the running time of your algorithm (time complexity of using scale)
    - Anything you want to share

Scores

- There are 5 testing dataset D1-D5.
  - D1 (40%) : D1 is already provided in input.txt.
  - D2-D5 (60%) : hidden test cases in D2-D5. (15 pts for each testing dataset)
  - Report (5%)
- **Total scores: 105 pts**
- **Penalty**
  - Use of a language other than C/C++ will get 0 pts
  - Failure to use file I/O: -10 pts (should NOT use standard I/O e.g. cout)
  - Compilation error: -10 pts
  - Incorrect output format: -10 pts
  - Incorrect file name: -10 pts

Cheating Policies

- **0 points for any cheating on assignments.**
- Allowing another student to examine your code is also considered cheating.

Late Submission

- Late submission dates: **2024/12/10 12:00 - 12/17 12:00**
- Late submission will get a **20%** penalty, with your final scores multiplied by 0.8.
- Submission more than one week late will not be accepted and will receive a score of 0.

If you have any questions, please feel free to email the TAs or come to EC126 after booking the office time with a TA via email.