

### Question - 1

#### 2-3 Tree and Red-Black Tree

SCORE: 5 points

Which of the following about 2-3 Tree and Red-Black Tree is correct?

- ☐ Only Red-Black Tree is "perfectly balanced"
- ☐ Both 2-3 Tree and Red-Black Tree are "perfectly balanced"

☒ 2-3 Tree is "perfectly balanced" and Red-Black Tree is "perfectly BLACK balanced"

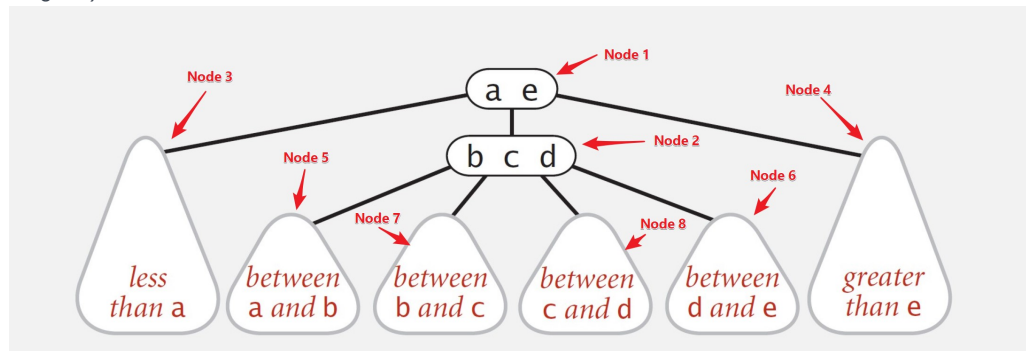
☐ 2-3 Tree is "perfectly balanced" and Red-Black Tree is "perfectly RED balanced"

### Question - 2

#### 2-3 Tree

SCORE: 5 points

While splitting a (temporary) 4-node in a 2-3 Tree (Node 2 in the diagram), which nodes will be modified?



- ☐ Node 1
- ☒ Node 1 and Node 2
- ☐ Node 1, Node 2, Node 3 and Node 4
- ☐ All of the above 8 Nodes (Node 1 to 8)

### Question - 3

#### 2-3 Tree

SCORE: 5 points

Which of the following about 2-3 tree's complexity is correct?

- ☐ Best case:  $\log_2 N$
- ☐ Worst case:  $\log_3 N$
- ☒ Between  $\log_2 N$  and  $\log_3 N$
- ☐ None of the above

#### Question - 4

##### Red Black Tree

SCORE: 5 points

When choosing a symbol table implementation for keys that are comparable, why are red-black trees generally preferred over hash tables?

☐ no they are not preferred



because red-black trees offer a better performance guarantee than do hash tables

☐ because they can be implemented using trees

☐ because they are balanced

#### Question - 5

##### Red-Black Tree Implementation

SCORE: 30 points

Please implement `put()` and `rotateLeft()` for Red-Black Tree. (15 pts for each)

`put()`: Line 72

`rotateLeft()`: Line 94

\* You don't need to change other code.

\* There is no `main()` method. Please use "Run Unit Tests" to test your code.

\* Unit tests for this question takes some time. Please be patient while compiling...