Question - 1

Trees

SCORE: **5 points**

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

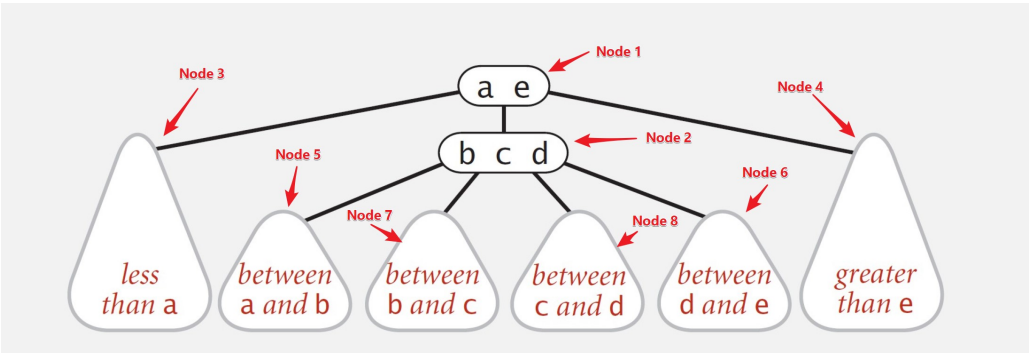
- Only Red-Black Tree is “perfectly balanced”
- Both 2-3 Tree and Red-Black Tree are “perfectly balanced”
- 2-3 Tree is “perfectly balanced” and Red-Black Tree is “perfectly BLACK balanced”
- 2-3 Tree is “perfectly balanced” and Red-Black Tree is “perfectly RED balanced”
- None of the above

Question - 2

2-3 tree

SCORE: **5 points**

While splitting a 4-node in a 2-3 Tree, which nodes will be touched (modified)?



- Node 1
- Node 1 and Node 2
- Node 1, Node 2, Node 3 and Node 4
- Node 1, Node 2, Node 3, Node 4, Node 5 and Node 6
- All of the above 8 Nodes (Node 1 to 8)

Question - 3

Red-Black Tree

SCORE: **5 points**

Help

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

- ( ) Every path from the root to null link has the same number of RED links
- (•) Every path from the root to null link has the same number of BLACK links
- ( ) Every path from the root to null link has the same number or (RED + BLACK) links
- ( ) None of the above

Question - 4

2-3 Tree

SCORE: **5 points**

Which of the following statements about the 2-3 tree's (theoretical) 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 - 5

Implementation

SCORE: **30 points**

Please implement *put*() and *rotateLeft*() for the Red-Black Tree. (15 pts for each)
put(): Line 72
rotateLeft(): Line 94

* You don't need to change any other code.
* Remember that the one thing that must remain invariant after rotate-left/right is that symmetric order must be maintained: smaller keys must precede middle keys which must precede larger keys.
* There is no *main*() method. Please use "Run Unit Tests" to test your code.
* The unit tests for this question takes some time. Please be patient while compiling...