CSE 222 HOMEWORK #11

RB, 2-3 and B-Trees

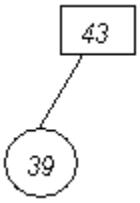
Orhan Aksoy

09104302

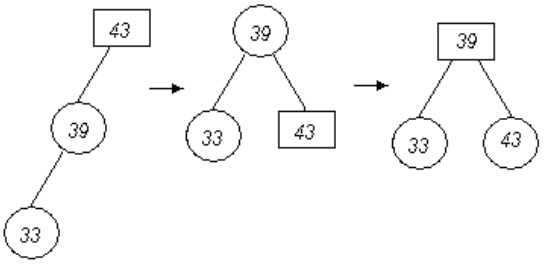
1. Answer 1

1.1 Add 43 as red and color it black (Root should be black)

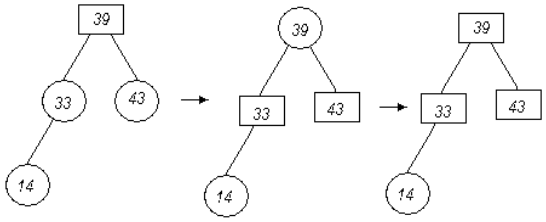
1.2 Add 39 as red and keep it.



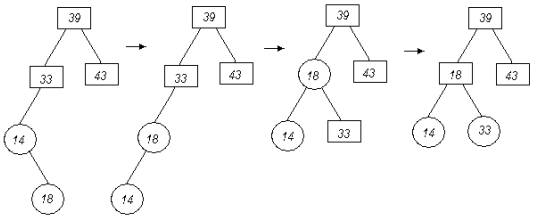
1.3 Add 33 as red, rotate right around 43, recolor 39 and 43.



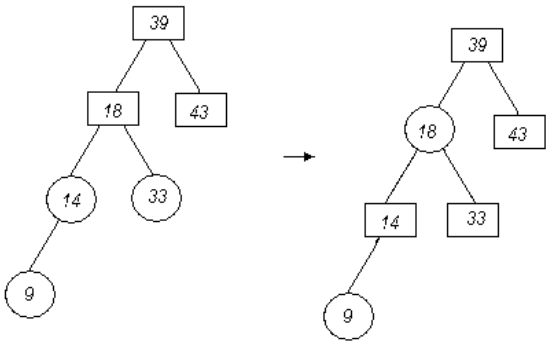
1.4 Add 14 as red, recolor parent and parent's sibling as black, and grandparent as red. Then recolor grandparent as black (root should be black)



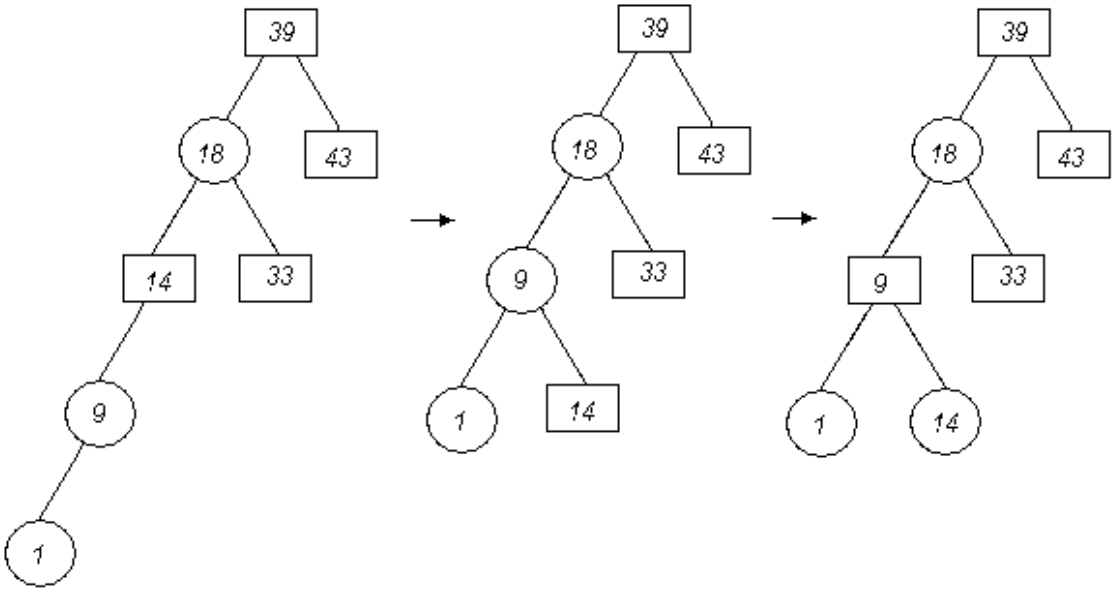
1.5 Add 18 as red, rotate left around 14, rotate right around 33, recolor 18 and 33.



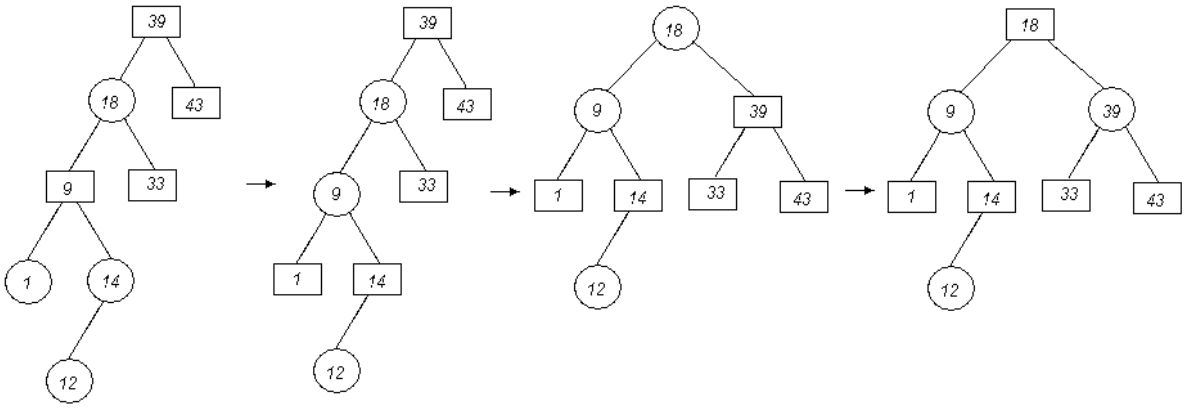
1.6 Add 9 as red, recolor parent and parent's sibling as black, and grandparent as red.



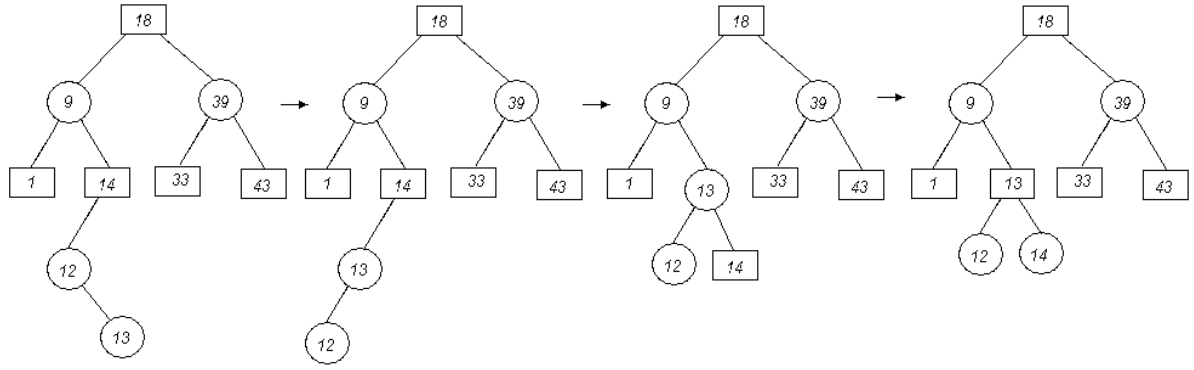
1.7 Add 1 as red, rotate right around 14, recolor 9 and 14.



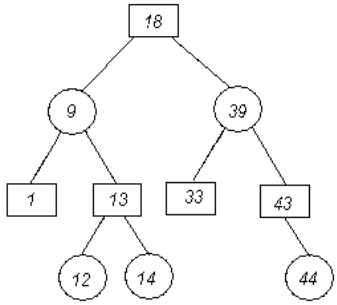
1.8 Add 12 as red, recolor parent and parent's sibling as black, and grandparent as red. Then, rotate right around 39 and recolor 18 and 39.



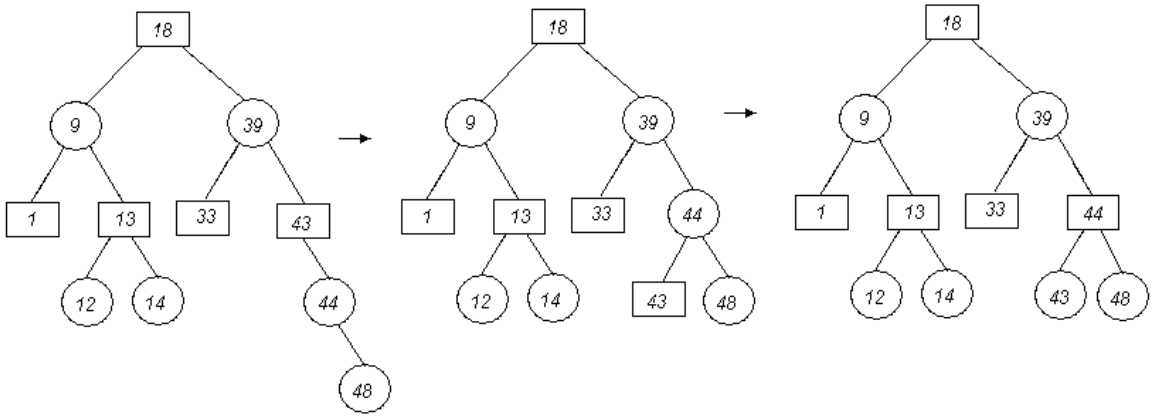
1.9 Add 13 as red, rotate left around 12, rotate right around 14, recolor 13 and 14.



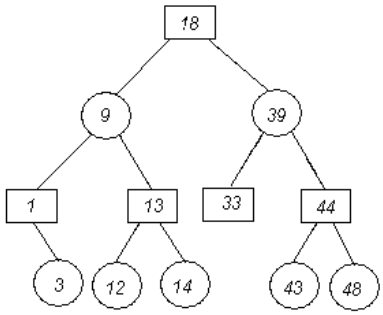
1.10 Add 44 as red. It's parent (43) is black, so keep it as it is.



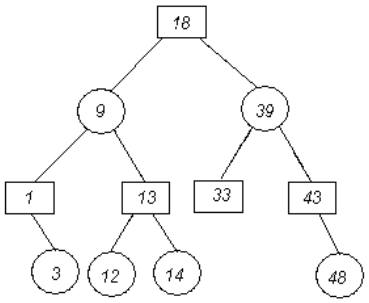
1.11 Add 48 as red, rotate left around 43, recolor 43 and 44.



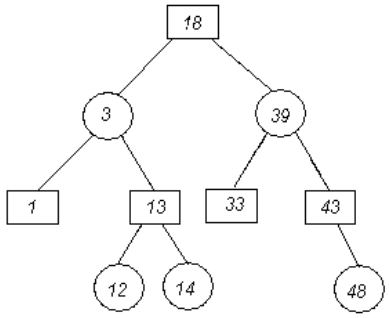
1.12 Add 3 as red. It's parent (1) is black, so keep it as it is.



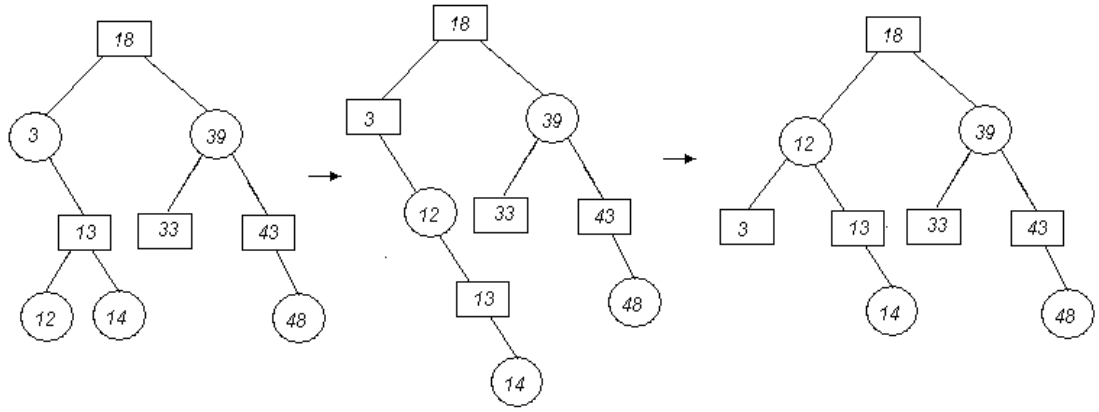
1.13 In order to remove 44, replace it with its in-order predecessor, which is 43. Since it is red, do nothing else.



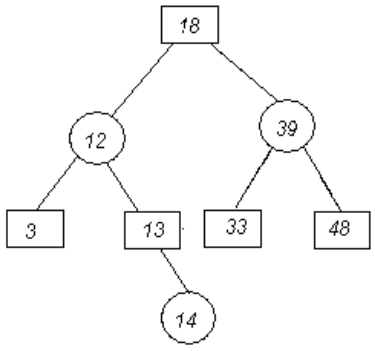
1.14In order to remove 9, replace it with its in-order predecessor, which is 3. Since it is red, do nothing else.



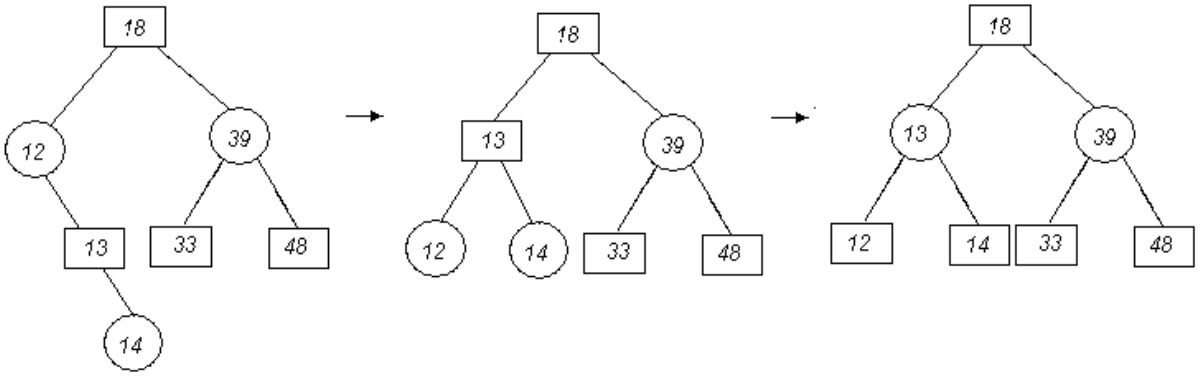
1.15 After removal of '1', rotate right around 13 and color parent (3) as black. Then, rotate left around 3.



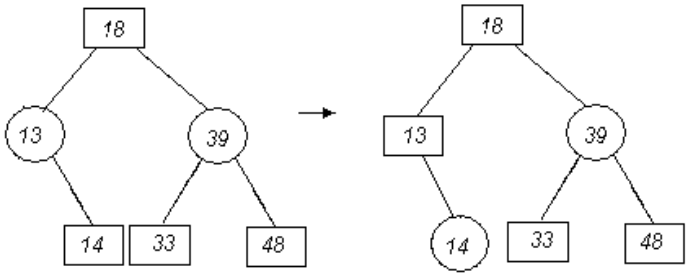
1.16In order to remove 43, replace its content with its red child, 48. Then, remove the red child.



1.17After removal of 3, rotate left around 12, and recolor 12, 13 and 14..



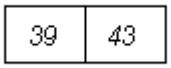
1.18After removal of 12, recolor its parent and its sibling.



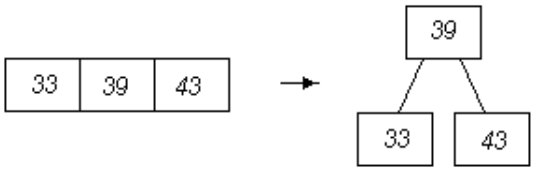
2. Answer 2

2.1. Add 43; 43 is the root.

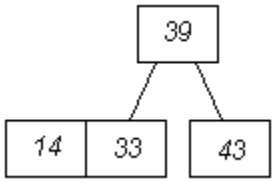
2.2. Add 39.



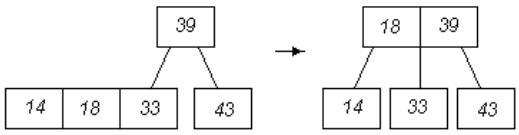
2.3. Add 33. Split the root, and set the middle element the new root



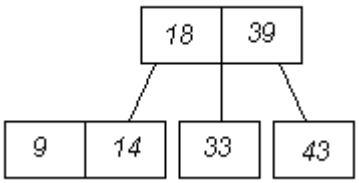
2.4. Add 14.



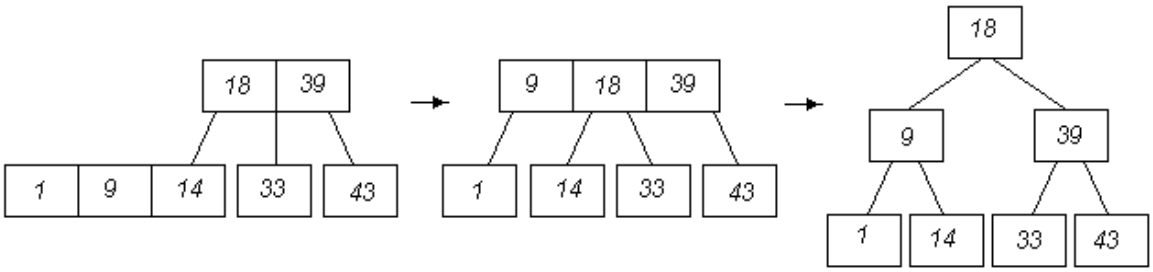
2.5. Add 18. Move 18 to the parent and reconnect children for the 3 node.



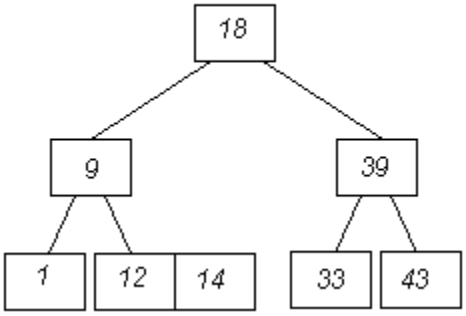
2.6. Add 9.



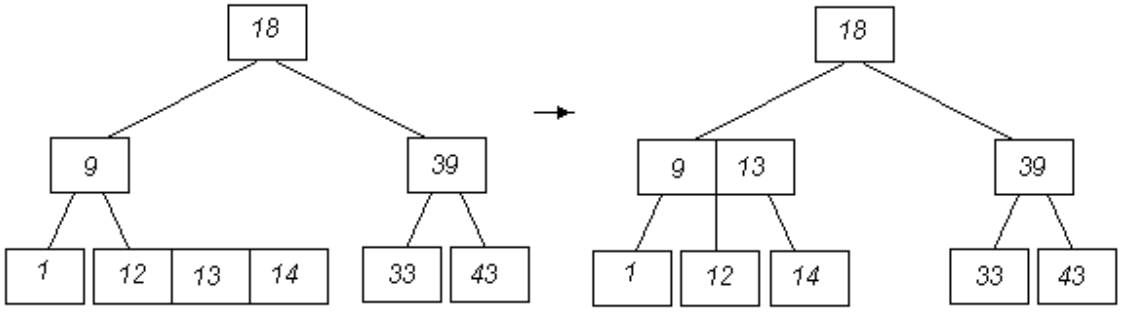
2.7. Add 1. Move 9 to the parent. Then, split the parent and set 18 the new root.



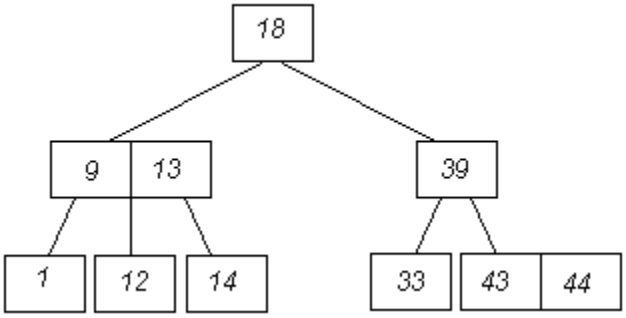
2.8. Add 12.



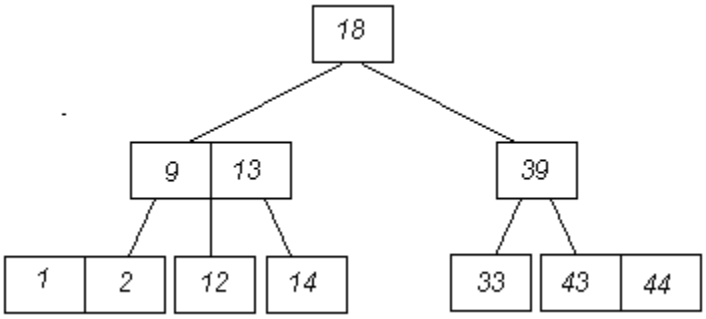
2.9. Add 13. Move 13 to its parent and reconnect children for the 3 node.



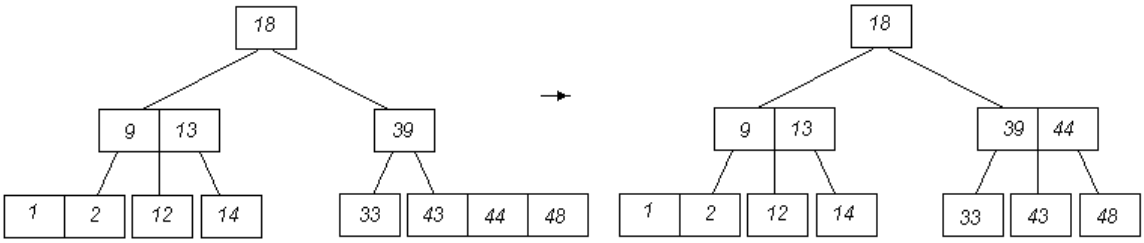
2.10. Add 44.



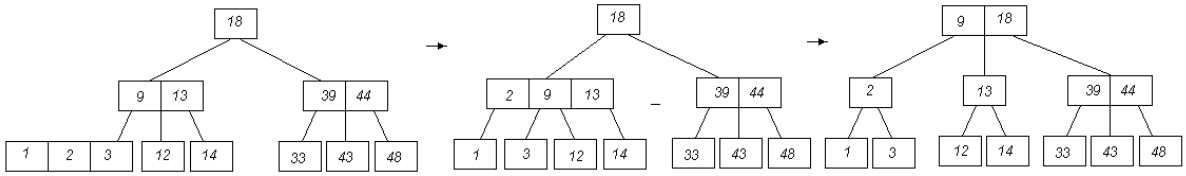
2.11. Add 2



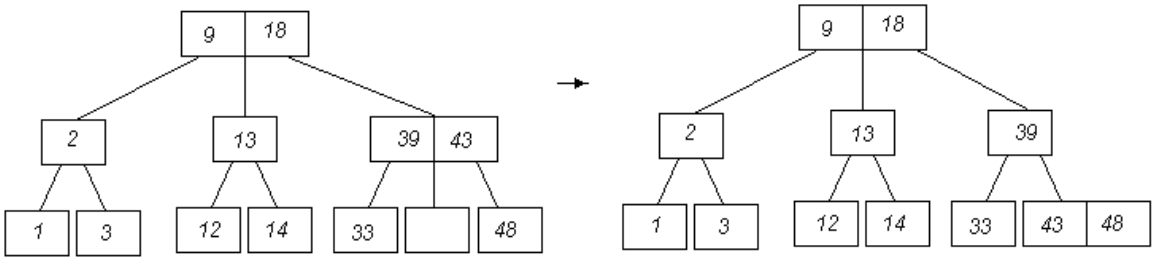
2.12. Add 48. Move 44 to the parent and reconnect children for the 3 node.



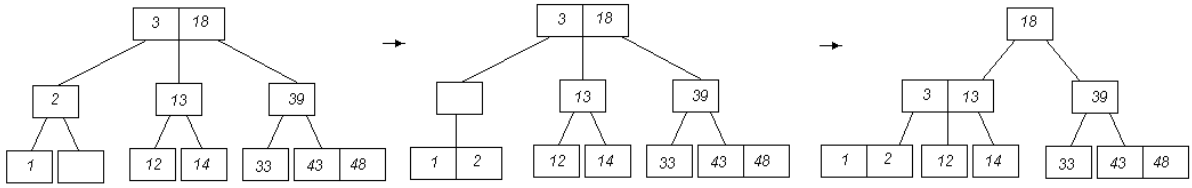
2.13. Add 3. Move 2 to the parent and reconnect children for the 3 node. Now, move 9 to the parent and reconnect the children for the 3 node.



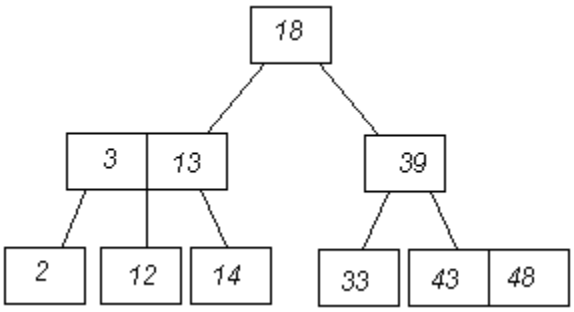
2.14. Delete 44. Swap 44 with its in-order predecessor, which is 43. Then, delete the empty leaf by converting the 3 node to a 2 node and merging 43 and 48.



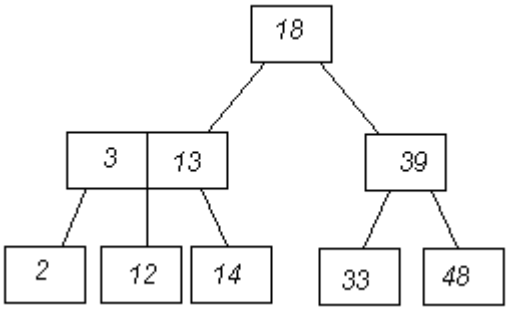
2.15. Delete 9. First, replace it with its in-order predecessor, which is 3. Then to remove the leaf, first, merge it with its parent. Then, the empty parent is filled by merging the parent of the empty parent and its sibling.



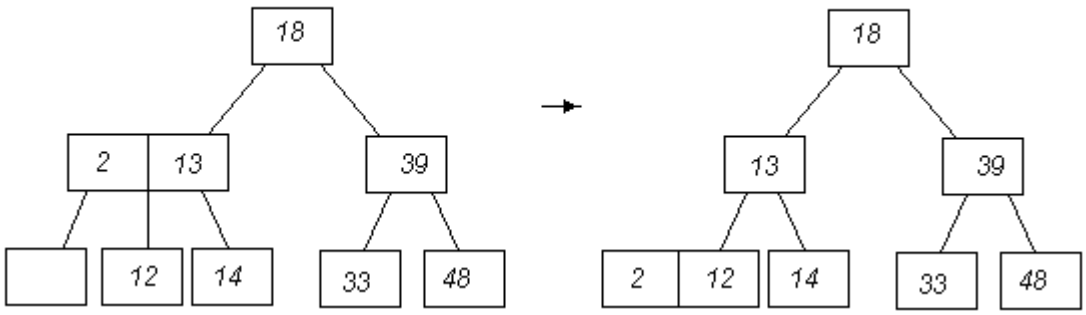
2.16. Delete 1



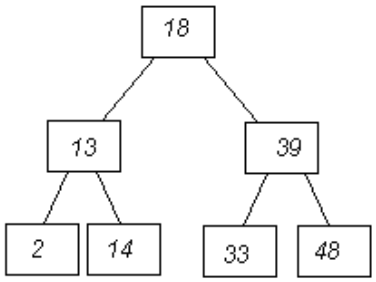
2.17. Delete 43



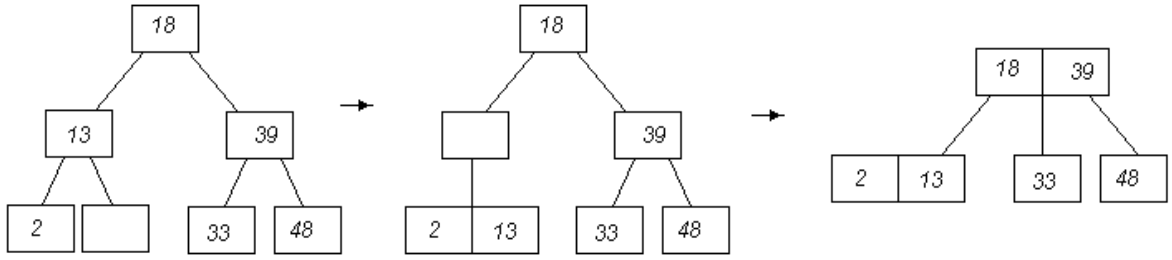
2.18. Delete 3. First, replace it with its in-order predecessor, which is 2. Then, merge the parent and right sibling of the empty leaf, making the parent a 2-node.



2.19. Delete 12.



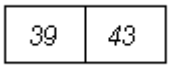
2.20. Delete 14. Then, merge its sibling with its parent. Then, the empty parent is filled by merging the parent of the empty parent and its sibling.



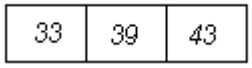
3. Answer 3

3.1. Add 43; 43 is the root.

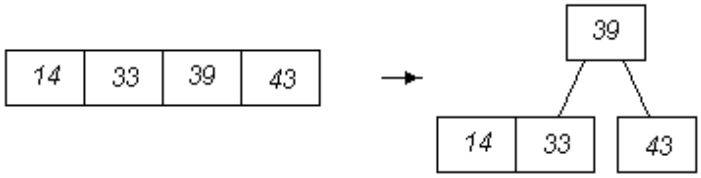
3.2. Add 39.



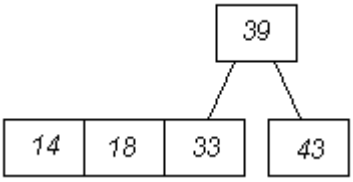
3.3. Add 39.



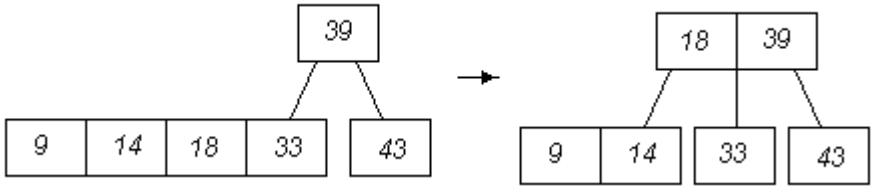
3.4. Add 14. Split the root, and set 39 the new root.



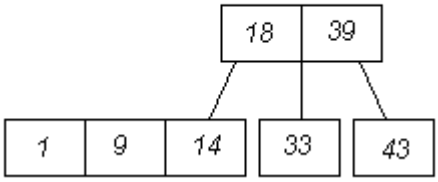
3.5. Add 18.



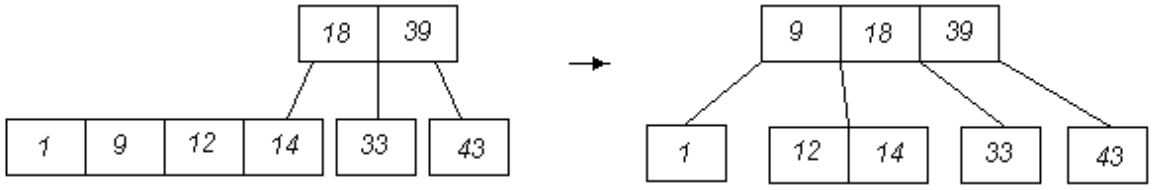
3.6. Add 9. Move 18 to the parent, and make the parent a 3 node. Then re-arrange children connections.



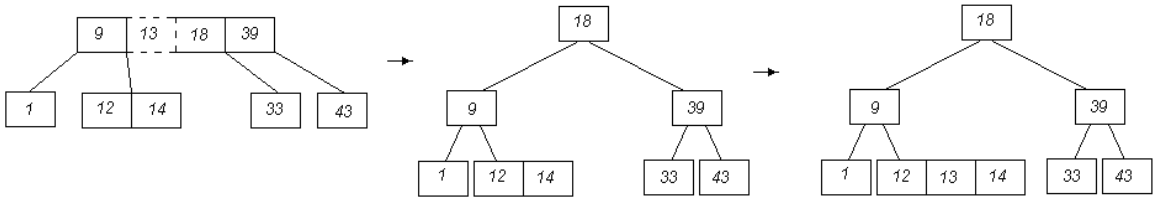
3.7. Add 1.



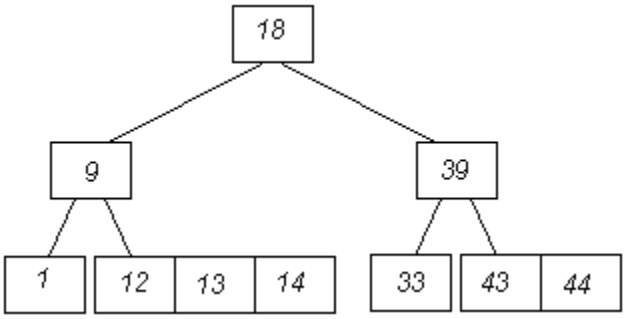
3.8. Add 12. Move 9 to the parent, making the parent a 4-node. Then, re-arrange children connections.



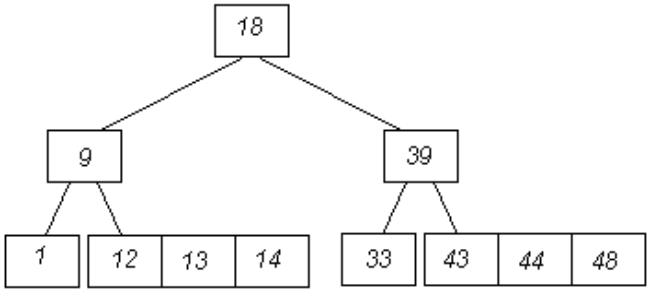
3.9. Add 13. When comparing with the root, the root is split by setting 18 as the new root. Then, 13 is added.



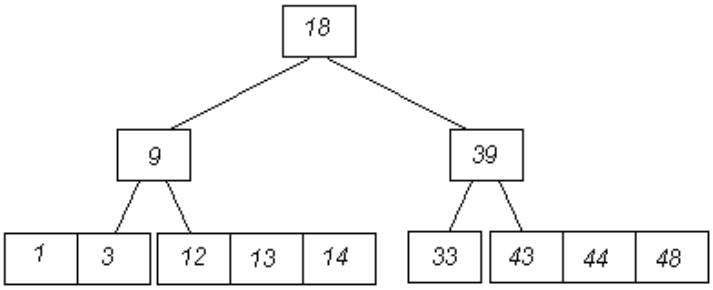
3.10. Add 44.



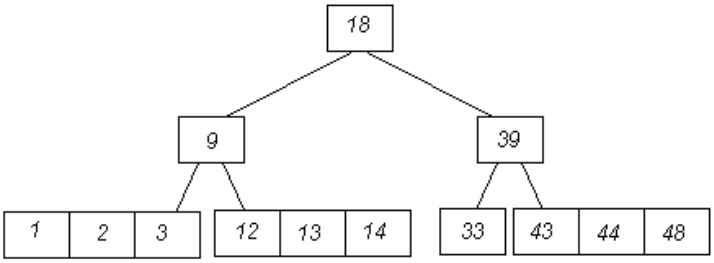
3.11. Add 48.



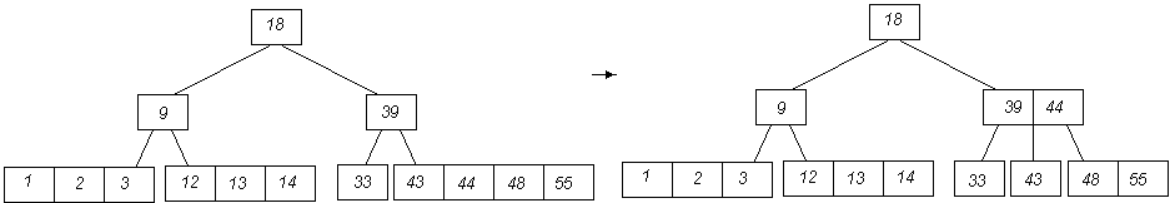
3.12. Add 3.



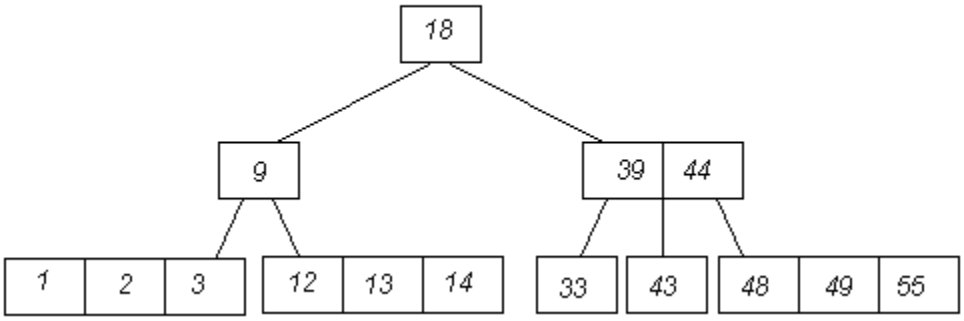
3.13. Add 2.



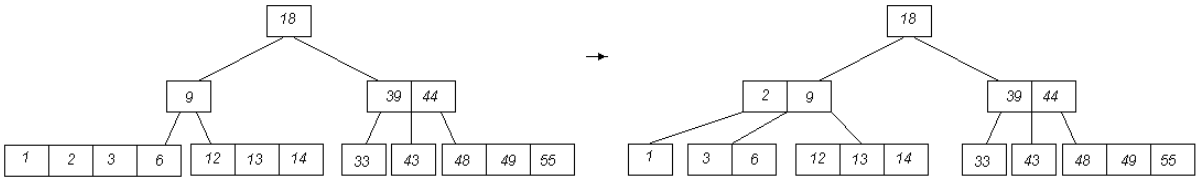
3.14. Add 55. Split the node and move 44 to the parent.



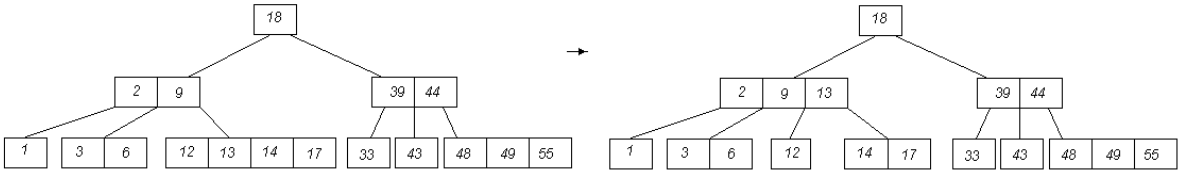
3.15. Add 49.



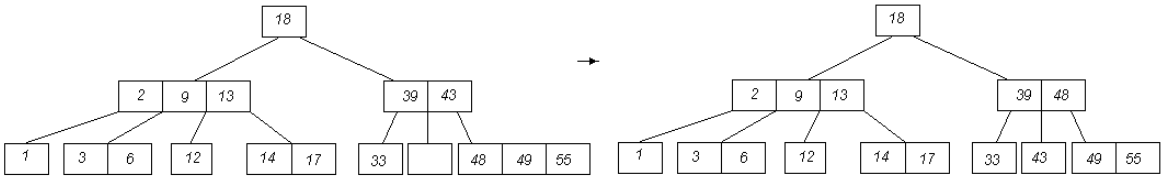
3.16. Add 6. Split the node and move 2 to the parent.



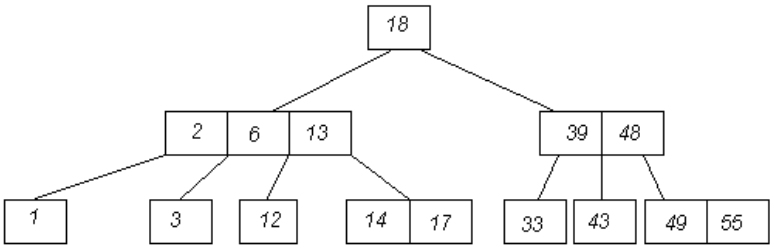
3.17. Add 17. Split the node and move 13 to the parent. Re-arrange the children connections.



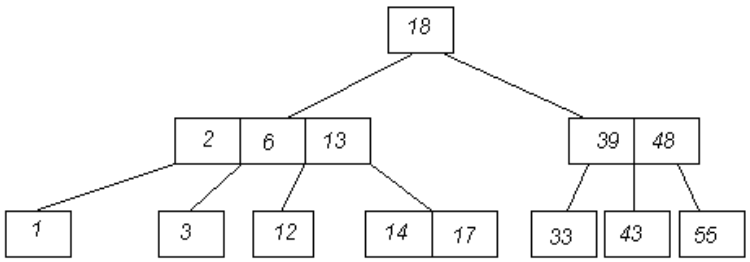
3.18. Remove 44. First, replace it with its in-order predecessor, which is 43. Then, fill the empty leaf with its right-parent, and its right parent with the smallest right sibling.



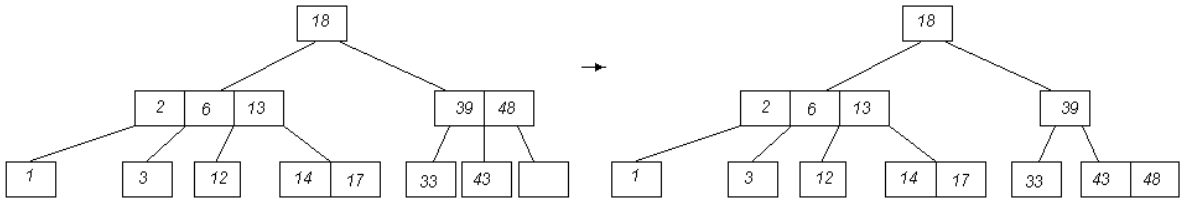
3.19. Remove 9. Simply replace it with its in-order predecessor, which is 6.



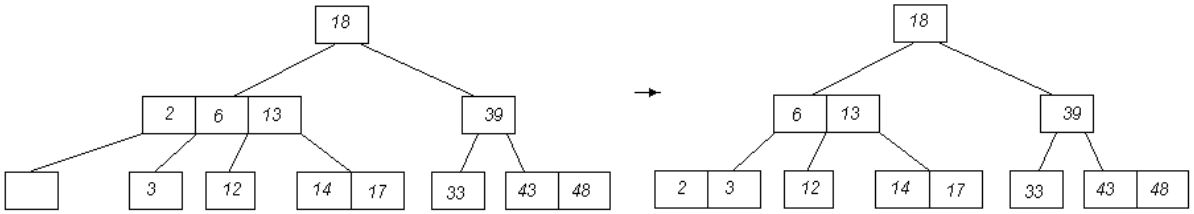
3.20. Remove 49.



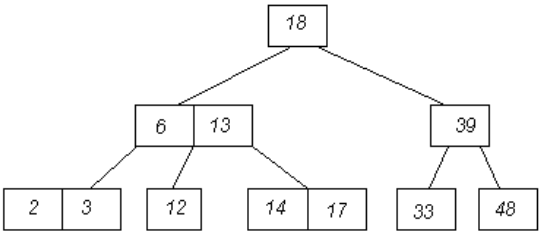
3.21. Remove 55. Merge its parent with its left sibling, making the parent a 2-node.



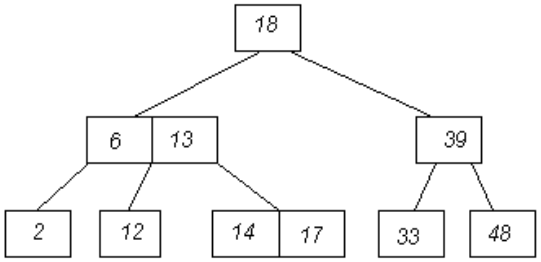
3.22. Remove 1. Merge its parent with its right sibling, making the parent a 3-node.



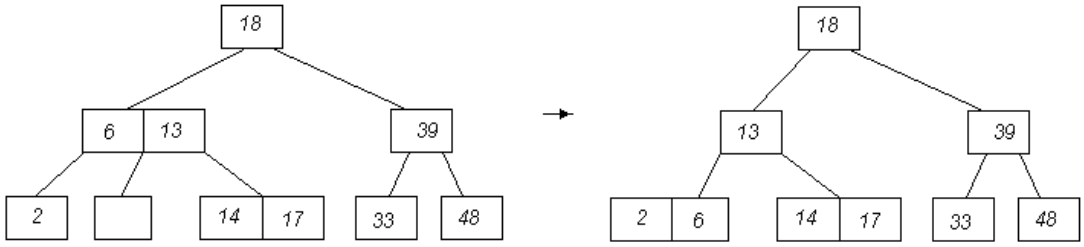
3.23. Remove 43.



3.24. Remove 3.



3.25. Remove 12. Merge the removed leaf's parent and its left sibling.



4. Answer 4

4.1. Maximum number of black nodes.

Let N(b) =Maximum number of black nodes. A full tree made up of black nodes only is a valid red-black tree. So, if it is possible to set up a full tree of height h using k elements, then the maximum number of black nodes using k elements is 2^h-1. So,

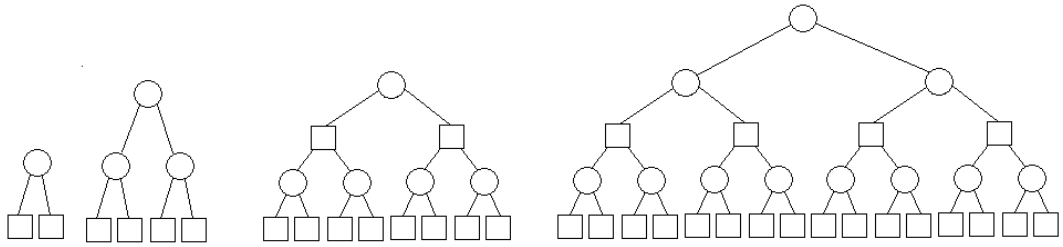
$$h = \lfloor \log_2(k+1) \rfloor$$
$$N(b) = 2^{\lfloor \log_2(k+1) \rfloor}$$

This means that for the values of k where 2^n-1 <= k < 2^{n+1}-1, N(b) = 2^n-1

4.2. Maximum number of red nodes.

Let N(b) =Maximum number of black nodes

Considering the full trees only, the following diagrams have the maximum number of red nodes (Squares represent red nodes):



For the even values of h,

$$N(r) = 2^1 + 2^3 + ... + 2^{h-1} = \frac{2}{3}(2^h - 1)$$

For the odd values of h,

$$N(r) = 2^2 + 2^4 + ... + 2^{h-2} = \frac{2^{h+1} - 4}{3}$$

For big values of h, N(r) =2/3(2^h-1).

The ratio between the maximum number of black nodes and maximum number of red nodes is

$$r = \frac{\frac{2}{3}2^h - 1}{2^h - 1} \approx \frac{2}{3}$$

For non-complete trees, all of the missing nodes are red, so for the smaller ratio, k should be 1 + complete tree node count.

5. Answer 5

5.1. Similarities between B- and B+ Trees:

- The leaves are all at the same depth in both trees.
- Stores sorted data
- Nodes store variable number of items.

5.2. Differences between B- and B+ Trees:

B- Tree	B+ Tree
Keys and data are stored in the nodes together	Keys and data are separated.
Internal nodes contain both keys and data	Internal nodes contain keys only. Data is stored in leaves only.
No connection between leaves	Leaves form a linked list.

5.3. Advantages of B- Trees:

- Search is easy, because of a possible match above leaf level.

5.4. Advantages of B+ Trees:

- Traversal easy since the leaves form a linked list.
- Since the data is separated from keys, the data can be stored in a secondary storage where access time is high. This access is done rarely (depending on the maximum number of items in a node).