Answer 1.

1. Pre-Order +, \*,3,7, ^, 4,2
2. In-Order 3,\*7,+,4^,2
3. Post-Order 3,7,\*,4,2,^,+

Answer 2.

18

13 25

4 15 25 28

1 14 17

3

2

When we remove node 12 we want to maintain the binary tree property that right child is bigger than the node and left child is smaller than the node, so we want to find the smallest element which is bigger than the node we removed so that the property can be maintained. We replace the node with the left most child of the right sub tree of the node being deleted

Answer 3.

1. Insertion Sort – n^2
2. Heap Sort – n log n
3. Quick Sort – n^2
4. Red-Black trees – n log n

Answer 4.

17

11 19

3 13

2 5

7

Answer 5.

7

3 18

10 22

8 11 26

15

We insert new node 15 as a right child of 11 to maintain one of the tree property, since it’s a child of a red node and also a leaf node it has to be black. However when we add the node 15, the black height property becomes invalid. To maintain this property we will do a left rotation on the branch connection node 7 and 18. And we will get the resulting tree as shown below

18

7 22

3 10 26

8 11

15

This will cause an issue with the root being of color red, to fix this we change the color of the root to black. Now we will modify the color

18

7 22

3 10 26

8 11

15

Still we have issue with the black height so we change the colors to adjust the height of the tree and get the following tree

18

7 22

3 10 26

8 11

15

We do another set of left rotation on branch connecting 7 and 10 resulting in which is a legal red bacl tree.

18

10 22

7 11 26

3 8 15