1. (5 Points) Given a = {1, 1, 3, 1, 1, 3, 4, 4, 1, 3}, trace the following count sort implementation

```
7     public static void sort(int[] a, int k) {
8         int n = a.length;
9         // create count array and initialize elements to zero
10        int[] count = new int[k+1];
11        int[] aux = new int[n];
12
13        for (int i=0; i<n; i++) {
14            count[a[i]] += 1;
15        }
16        // count[i] now contains the number of elements equals to i
17
18        for (int i=1; i<=k; i++) {
19            count[i] += count[i-1];
20        }
21        // count[i] now contains the number of elements less than or equals to i
22
23        for (int i=n-1; i>=0; i--) {
24            count[a[i]] -= 1;
25            aux[count[a[i]]] = a[i];
26        }
27        // now the sorted results are in aux
28
29        for(int i=0; i<n; i++) {
30            a[i] = aux[i];
31        }
32        // now the sorted results are in a as well
33    }
```

a) provide content of count array after line 15.

Index	0	1	2	3	4
count	0	5	0	3	2

b) provide content of count array after line 20.

Index	0	1	2	3	4
count	0	5	5	8	10

2. (5 Points) Using radix sort to sort the following inputs, where each digit is sorted by count sort. Provide the sorted results after sorting each digit.

Input	14431, 51154, 33691, 1603, 76453, 42439, 57201, 35079, 9839, 67547
	14431, 33691, 57201, 1603, 76453, 51154, 67547, 42439, 35079, 9839
	57201, 1603, 14431, 42439, 9839, 67547, 76453, 51154, 35079, 33691
	35079, 51154, 57201, 14431, 42439, 76453, 67547, 1603, 33691, 9839
	51154, 1603, 42439, 33691, 14431, 35079, 76453, 57201, 67547, 9839
	1603, 9839, 14431, 33691, 35079, 42439, 51154, 57201, 67547, 76453

3. (15 points) Given the following BST with keys, list keys in
a) in-order traversal order
b) pre-order traversal order
c) post-order traversal order

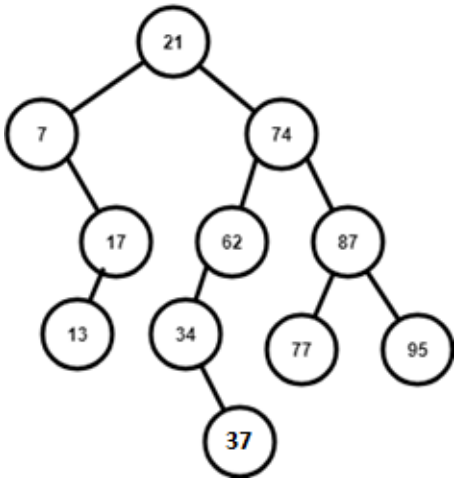
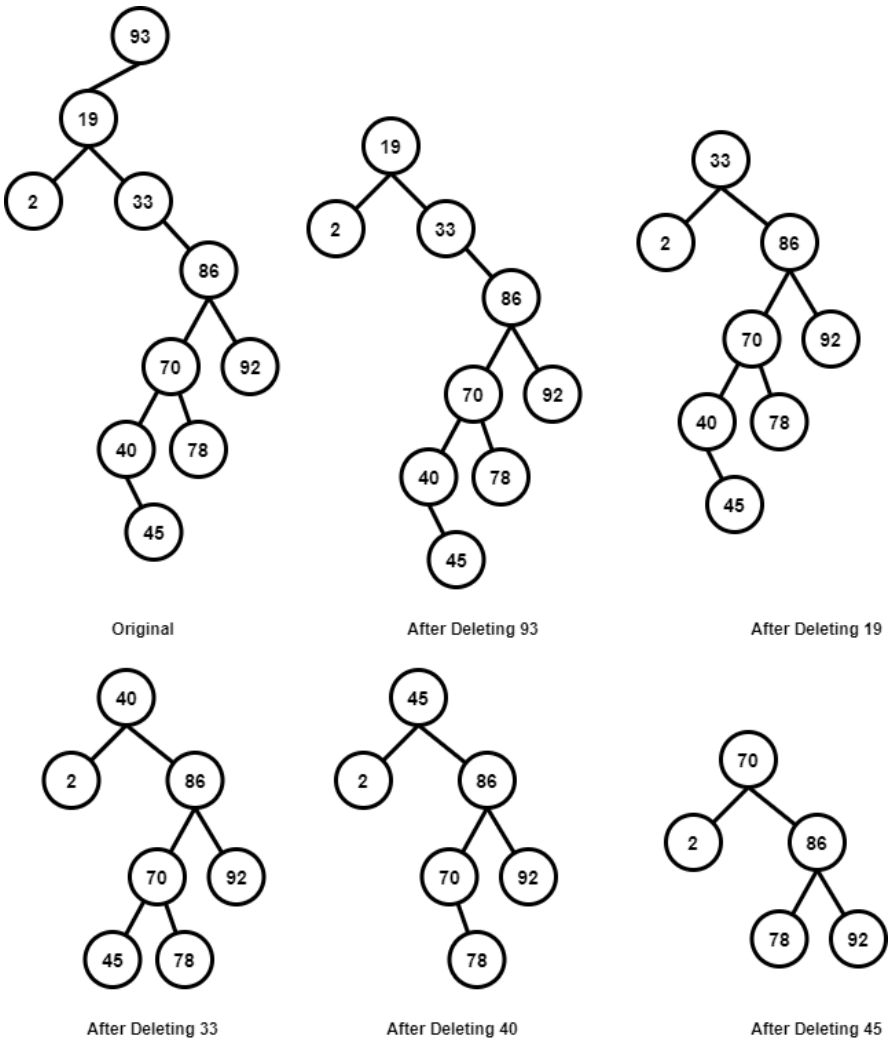


Figure 1

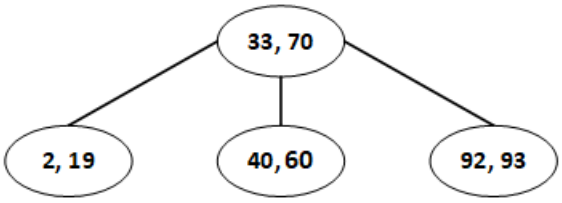
Answer:
In-order: 7,13, 17, 21, 34, 37, 62, 74, 77, 87, 95
Pre-order: 21, 7, 17, 13, 74, 62, 34, 37, 87, 77, 95
Post-order: 13, 17, 7, 37, 34, 62, 77, 95, 87, 74, 21

4. (5 Points) The following figure shows a BST, where key is displayed inside each node. Using the BST deletion algorithm provided in class, do deletion 5 times, each time delete the root node. Provide the result tree for each deletion. In total, there should be 5 result trees.

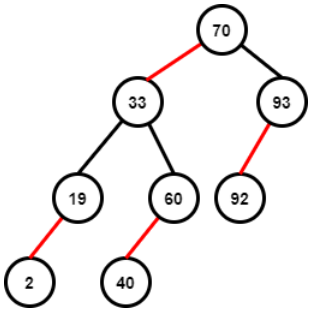
Answer:



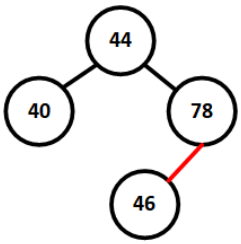
5. (5 Points) Given the following 2-3 tree, provide the corresponding red-black BST.



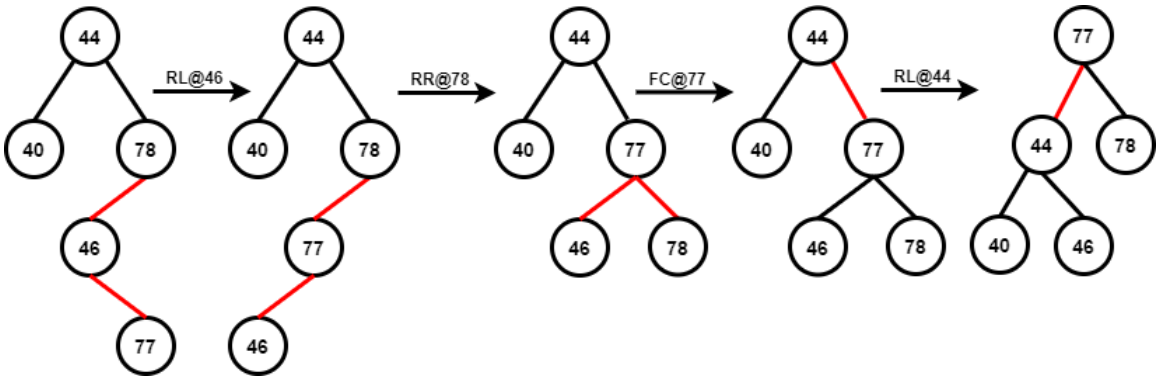
Answer:



6. (5 Points) Given the following red-black BST, provide step by step results for inserting 77 into the tree. Your answer should include the intermediate tree after initial insertion and tree after each transformation.



Answer:



7. (5 Points) Fill the blanks in the following table.

Number of nodes in BST	Number of null links
1	2
2	3
3	4
4	5
n	n+1

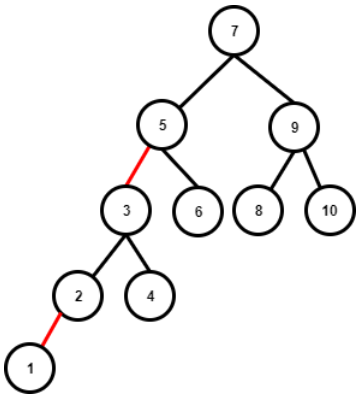
8. (5 Points) Suppose that we have numbers between 1 and 1000 in a binary search tree, and we want to search for the number 363. Which of the following sequences could not be the sequence of nodes examined?
a. 2, 252, 401, 398, 330, 344, 397, 363.
b. 924, 220, 911, 244, 898, 258, 362, 363.
c. 925, 202, 911, 240, 912, 245, 363.
d. 2, 399, 387, 219, 266, 382, 381, 278, 363.
e. 935, 278, 347, 621, 299, 392, 358, 363.
Answer: c and e

9. (5 Points) We can sort a given set of n numbers by first inserting numbers one by one into an initially empty BST and then printing the keys via in-order traversal order. What are the worst case and best-case running times for this sorting algorithm? Briefly explain your answer.
Answer:

Best-case Running Time	Worst-case Running Time
1) O(nlgn), assuming numbers are distinct If each insertion results in a balanced tree, the total running time of n insertion is O(nlgn). In-order traversal running time is O(n). Thus, the total running time is O(nlgn). 2) This sorting algorithm will not work is there are duplicate numbers in the array, as the BST we studied does not allow duplicate keys, inserting numbers into the BST will results in fewer number of nodes in BST than the input array length.	$O(n^2)$ If we insert numbers in increasing order in to BST, we get the worst-case insertion running time $O(n^2)$. In-order traversal running time is O(n). Thus, the total running time is $O(n^2)$.

10. (5 Points) Draw a highest red-black BST using 10 distinct keys between 1 and 10.

Answer:



11. (10 Points) Implement concat method inside provided MyBST class.

```
/**
 * Concatenate another BST (bst2) with the current BST.
 * Every key in bst2 is greater than all the keys in the
current BST.
 * Let h = max(this.getHeight(), bst2.getHeight()),
 * the worst-case running time should be O(h).
 */
```

```
public void concat(BST<Key, Value> bst2) {
    if (bst2.size() > 0) {
        if (this.size() ==0) {
            this.root = bst2.root;
        } else {
            Node<Key, Value> oldRoot = this.root;

            Key key = bst2.min();
            Value value = bst2.get(key);
            this.root = new Node<Key, Value>(key, value, 1);
            this.root.left = oldRoot;

            bst2.deleteMin();
            this.root.right = bst2.root;
            this.root.n = size(this.root.left) + size(this.root.right) + 1;
        }
    }
}
```

Submission Note

1) For written part of the questions:
   a. Write your answers inside a text document (in plain text, MS Word, or PDF format)
   b. Name the file as csc130.firstname.lastname.assignment3.txt(doc, docx, or pdf) with proper file extension
2) For programming part of the questions
   a. Use JDK 1.8 and Junit5
   b. Put your full name at the beginning of every source file you created or modified. **2 points will be deducted if your names are not included in the source files.**
   c. Do not change the provided package, class, or method name. You can add extra classes or methods if they are needed.
   d. **If your code does not compile, you will get zero point**.
   e. Use the provided tests to verify your implementation. **Extra tests might be used for grading.**
   f. Zip all the source files into csc130.firstname.lastname.assignment3.zip
3) Submit both of your files (text document and zip file) via Canvas course web site.
4) Due Nov 5, 11:59 PM