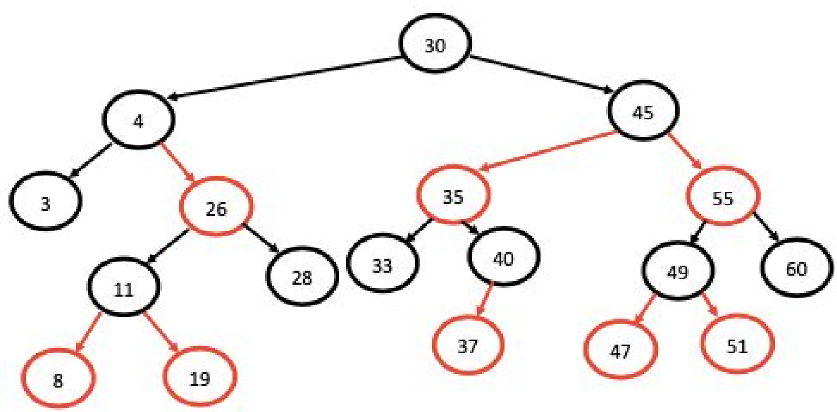
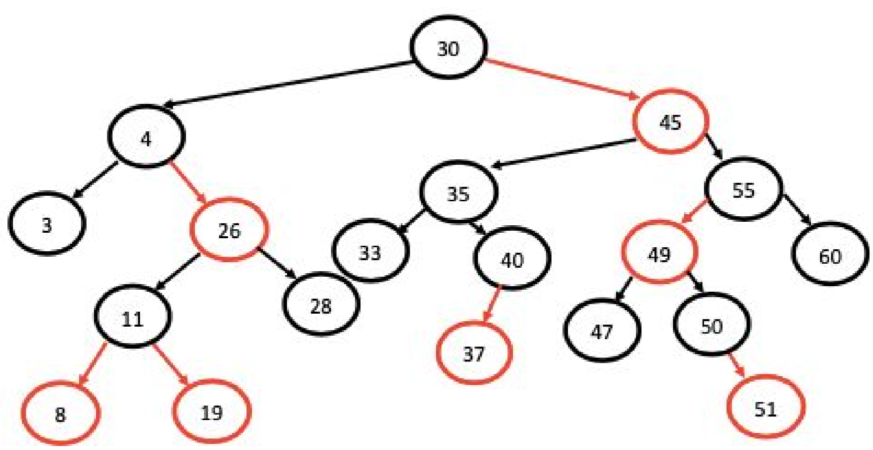
Title: Binary Search Trees and Hashing
Author: Deniz Çalkan
ID: 21703994
Section: 1
Assignment: 4
Description: Answers of questions 1,2 and 3.

Q1)
a)



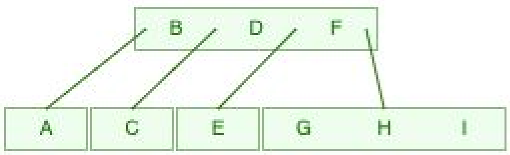
b)

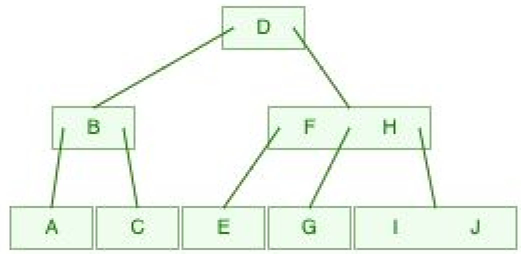


Q2)

a) The maximum number of elements 2-3 tree with height h can hold is 3^h -1. Because, 2-3 trees are balanced trees and the maximum number of elements a node can hold is 2 with 3 children.

b) After inserting the letter ‘J’ tree grows to height 3.





c) Applying in-order traversal would work because red black trees act like binary search trees. So the answer is O(n).

d) No, because red-black tree should always have a black root node and at each subtree rooted with a red node will violate this property.

Q3) a) We can use a hash table to solve this problem. We can insert the elements to an empty hash table then we can start from index 0 of the array then checking if sum minus this value exists in the hash table. If it does we can print these two values and so on.

b) Linear probing:

Slot	0	1	2	3	4	5	6
Content	30	15	22	11	14	11	-

Quadratic probing:

Slot	0	1	2	3	4	5	6
Content	30	15	22	11	14	-	11