THIS DOCUMENT WAS CREATED IN SPRING, 2019 BY A GROUP OF TUTORS TO PREPARE FOR A REVIEW SESSION, AND MAY NOT ACCURATELY REPRESENT THE MATERIAL TAUGHT IN LATER COURSE OFFERINGS

SOLUTIONS:

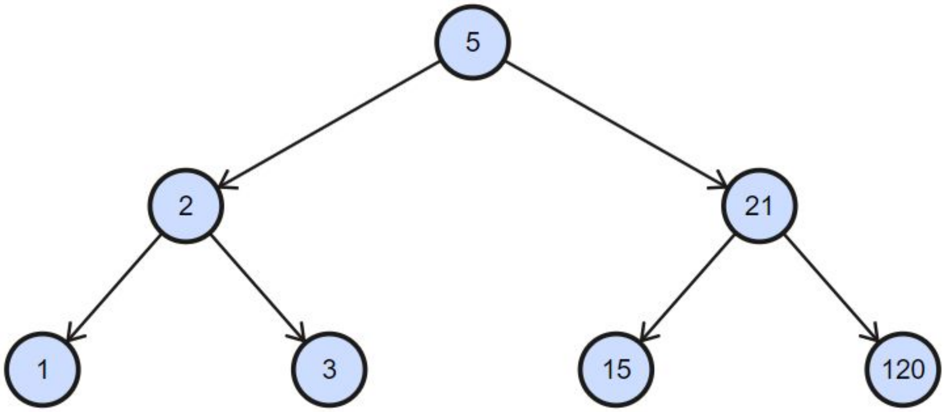
1.

O(NlogN)

2.

Data Structure	*Remove*	*Find*	*Insert*	*Find Min*	*Find Max*
Balanced BST	*O(logN)*	*O(logN)*	*O(logN)*	*O(logN)*	*O(logN)*
BST (Not necessarily balanced)	*O(N)*	*O(N)*	*O(N)*	*O(N)*	*O(N)*
Singly Linked List in ascending order (stores the head and tail)	*O(N)*	*O(N)*	*O(N)*	*O(1)*	*O(1)*
Unsorted Singly Linked List (stores the head and tail)	*O(N)*	*O(N)*	**at tail O(1)*	*O(N)*	*O(N)*
Sorted Array in ascending order, using binary search	*O(N)*	*O(logN)*	*O(N)*	*O(1)*	*O(1)*
Unsorted Array	*O(N)*	*O(N)*	*O(N)*	*O(N)*	*O(N)*

3.



4.

```
std::vector<int> IntBST::giveAllLarger(int value){
        std::vector<int> list;
        IntBST::Node* target = getNodeFor(value, this->root);
        if(!target) return list;
        target = getSuccessorNode(target->info);
        while(target){
                list.push_back(target->info);
                target = getSuccessorNode(target->info);
        }
        return list;
}
```

5.

No, 38 cannot be in the right subtree of 56.

Preorder: 56, 48, 17, 12, 52, 57, 64
Postorder: 12, 17, 52, 48, 64, 57, 56
Inorder: 12, 17, 48, 52, 56, 57, 64