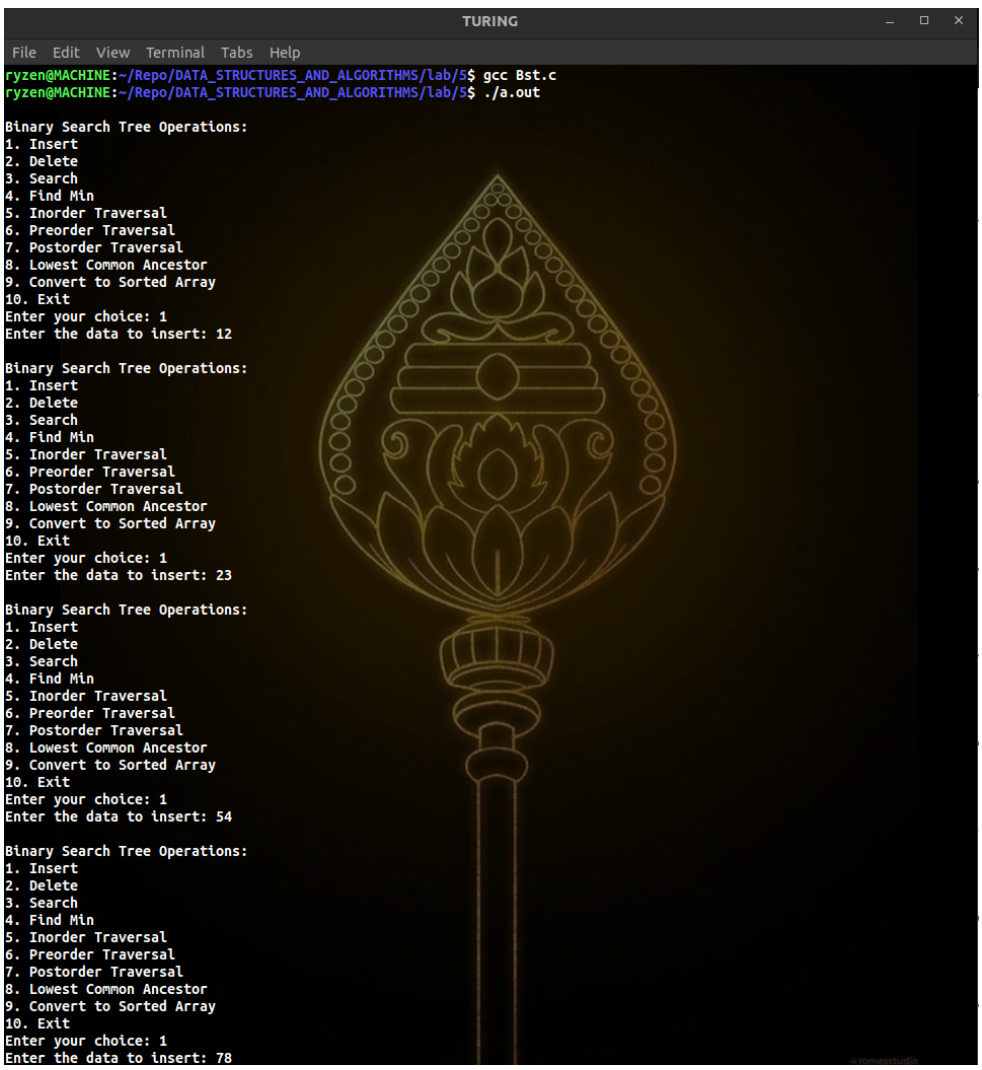
Data Structures and Algorithms

Assignment - v

1. 1. Implement BST with the following functions:

   a. Insert



b. Delete

```
Binary Search Tree Operations:
1. Insert
2. Delete
3. Search
4. Find Min
5. Inorder Traversal
6. Preorder Traversal
7. Postorder Traversal
8. Lowest Common Ancestor
9. Convert to Sorted Array
10. Exit
Enter your choice: 2
Enter the key to delete: 1

Binary Search Tree Operations:
1. Insert
2. Delete
3. Search
4. Find Min
5. Inorder Traversal
6. Preorder Traversal
7. Postorder Traversal
8. Lowest Common Ancestor
9. Convert to Sorted Array
10. Exit
Enter your choice: _
```

c. search

```
Binary Search Tree Operations:
1. Insert
2. Delete
3. Search
4. Find Min
5. Inorder Traversal
6. Preorder Traversal
7. Postorder Traversal
8. Lowest Common Ancestor
9. Convert to Sorted Array
10. Exit
Enter your choice: 3
Enter the key to search: 1
Key not found in the tree.

Binary Search Tree Operations:
1. Insert
2. Delete
3. Search
4. Find Min
5. Inorder Traversal
6. Preorder Traversal
7. Postorder Traversal
8. Lowest Common Ancestor
9. Convert to Sorted Array
10. Exit
Enter your choice: _
```

d. find min

```
Binary Search Tree Operations:
1. Insert
2. Delete
3. Search
4. Find Min
5. Inorder Traversal
6. Preorder Traversal
7. Postorder Traversal
8. Lowest Common Ancestor
9. Convert to Sorted Array
10. Exit
Enter your choice: 4
Minimum value in the tree: 4
```

e. reverse (includes preorder, postorder)

```
Binary Search Tree Operations:
1. Insert
2. Delete
3. Search
4. Find Min
5. Inorder Traversal
6. Preorder Traversal
7. Postorder Traversal
8. Lowest Common Ancestor
9. Convert to Sorted Array
10. Exit
Enter your choice: 5
Inorder Traversal: 4 12 23 54 78

Binary Search Tree Operations:
1. Insert
2. Delete
3. Search
4. Find Min
5. Inorder Traversal
6. Preorder Traversal
7. Postorder Traversal
8. Lowest Common Ancestor
9. Convert to Sorted Array
10. Exit
Enter your choice: 6
Preorder Traversal: 12 4 23 54 78

Binary Search Tree Operations:
1. Insert
2. Delete
3. Search
4. Find Min
5. Inorder Traversal
6. Preorder Traversal
7. Postorder Traversal
8. Lowest Common Ancestor
9. Convert to Sorted Array
10. Exit
Enter your choice: 7
Postorder Traversal: 4 78 54 23 12
```

f. find the lowest common ancestor of two nodes in a binary search tree

```
Binary Search Tree Operations:
1. Insert
2. Delete
3. Search
4. Find Min
5. Inorder Traversal
6. Preorder Traversal
7. Postorder Traversal
8. Lowest Common Ancestor
9. Convert to Sorted Array
10. Exit
Enter your choice: 8
Enter the values of two nodes to find their lowest common ancest
```

g. convert a binary search tree to a sorted array

```
Binary Search Tree Operations:
1. Insert
2. Delete
3. Search
4. Find Min
5. Inorder Traversal
6. Preorder Traversal
7. Postorder Traversal
8. Lowest Common Ancestor
9. Convert to Sorted Array
10. Exit
Enter your choice: 9
Sorted Array: 4 12 23 54 78
```

2. write a program to check whether a given tree is a valid BST.