

Kathmandu University
Department of Computer Science and Engineering
Dhulikhel, Karve



COMP 202

Lab 3 Report

Submitted By:

Mohit Pokharel

Roll No.-42

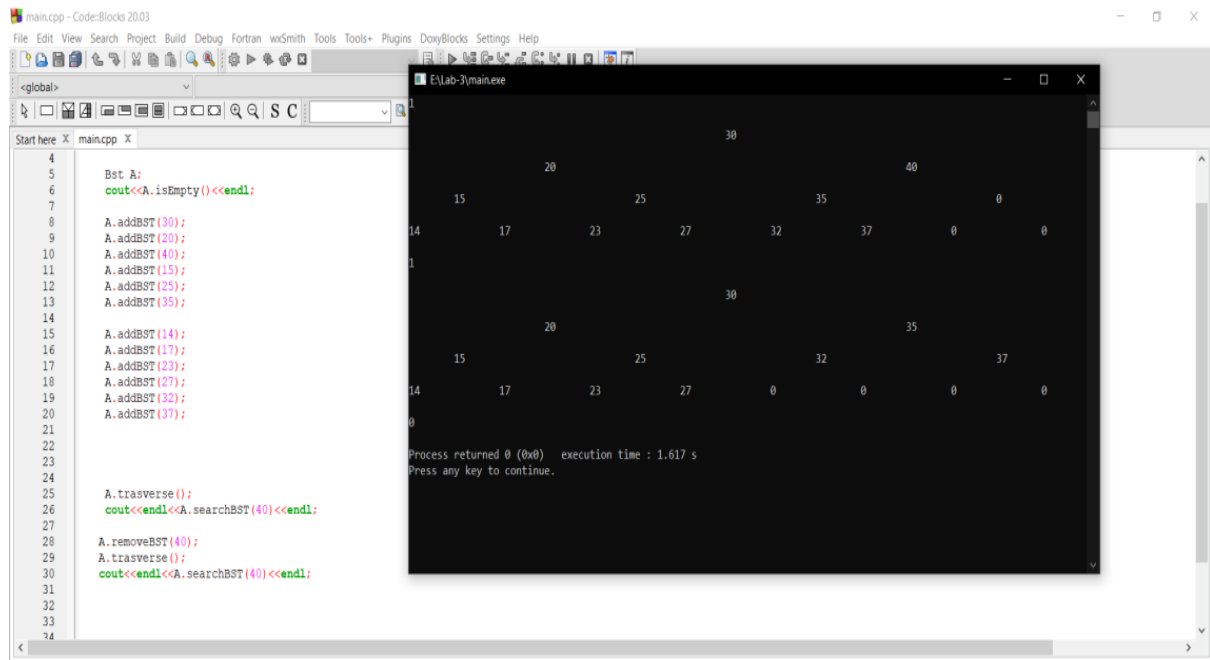
CE, 2nd Year, 1st Sem

Submitted to:

Rajani Chulyadyo

Department of Computer Science and Engineering

I have done the implementation of BST using arrays.



The screenshot shows the Code::Blocks IDE with a C++ program for a Binary Search Tree (BST) implemented using arrays. The program is named `main.cpp` and is located in the `E:\Lab-3\main.exe` directory. The code defines a `Bst` structure with an array `A` and a `size` variable. It includes functions for adding, removing, and traversing the BST. The execution output shows the tree structure after adding and removing elements.

```
4  Bst A;  
5  cout<<A.isEmpty()<<endl;  
6  
7  
8  A.addBST(30);  
9  A.addBST(20);  
10 A.addBST(40);  
11 A.addBST(15);  
12 A.addBST(25);  
13 A.addBST(35);  
14  
15 A.addBST(14);  
16 A.addBST(17);  
17 A.addBST(23);  
18 A.addBST(27);  
19 A.addBST(32);  
20 A.addBST(37);  
21  
22  
23  
24  
25 A.traverse();  
26 cout<<endl<<A.searchBST(40)<<endl;  
27  
28 A.removeBST(40);  
29 A.traverse();  
30 cout<<endl<<A.searchBST(40)<<endl;  
31  
32  
33  
34
```

Execution output:

```
Process returned 0 (0x0)   execution time : 1.617 s  
Press any key to continue.
```

Github link: https://github.com/Mohit-42/CE_2019_42_43/tree/ArrayBST