Session 8: Binary Search Trees (2)

Data Structures and Algorithm 1 - Lab

Yahya Tawil
26 Nov 2021

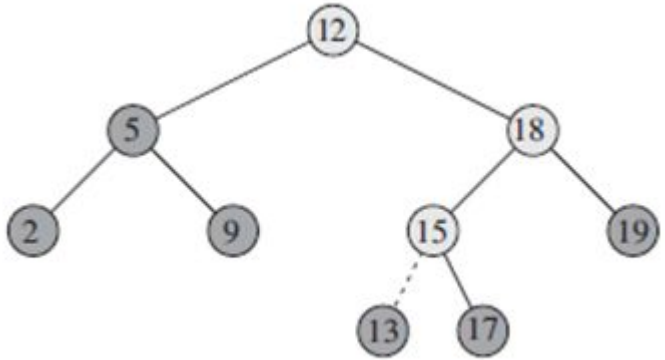
Binary Search Tree Operation: Insertion

- In Insertion and deletion, The data structure must be modified to reflect this change, but in such a way that the binary-search-tree property **continues to hold**.

```
TREE-INSERT(T, z)
 1  y = NIL
 2  x = T.root
 3  while x ≠ NIL
 4        y = x
 5        if z.key < x.key
 6              x = x.left
 7        else x = x.right
 8  z.p = y
 9  if y == NIL
10        T.root = z          // tree T was empty
11  elseif z.key < y.key
12        y.left = z
13  else y.right = z
```

Binary Search Tree Operation: Insertion

Inserting an item with key 13 into a binary search tree



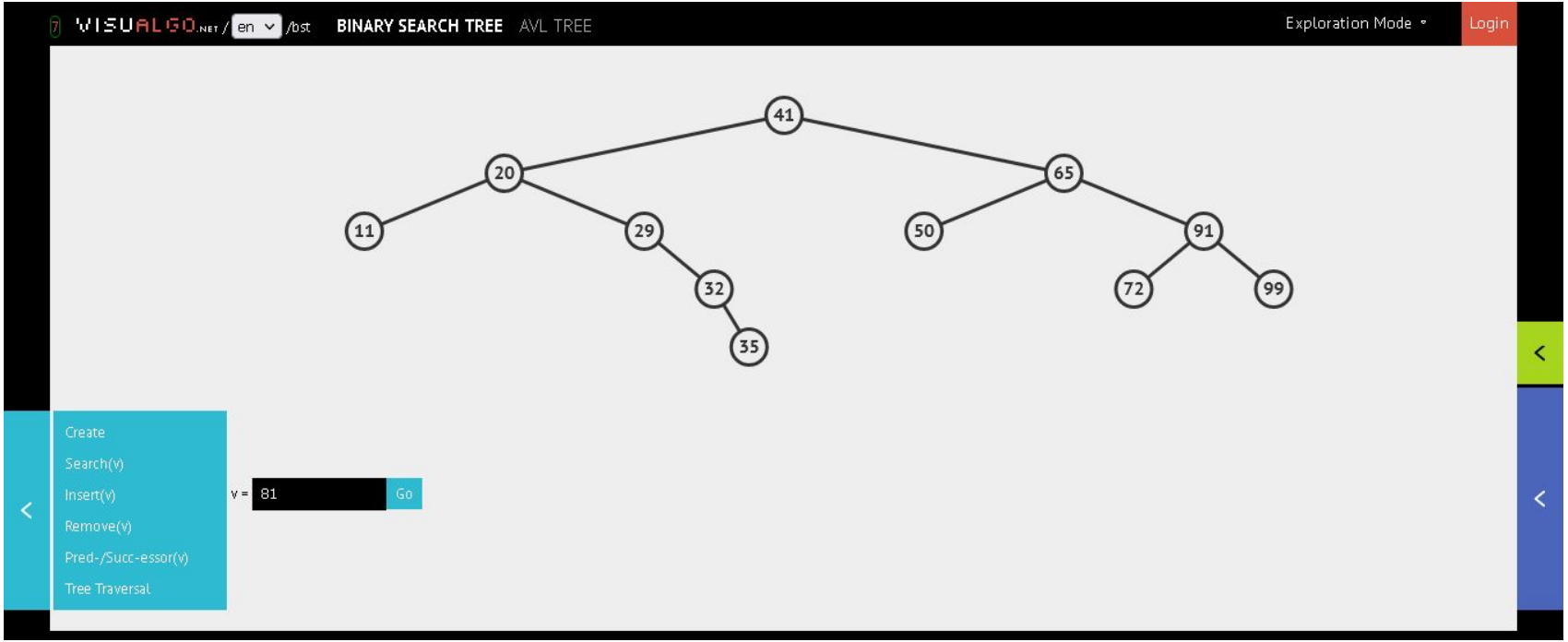

```
TREE-INSERT(T, z)
 1  y = NIL
 2  x = T.root
 3  while x ≠ NIL
 4        y = x
 5        if z.key < x.key
 6              x = x.left
 7        else x = x.right
 8  z.p = y
 9  if y == NIL
10        T.root = z          // tree T was empty
11  elseif z.key < y.key
12        y.left = z
13  else y.right = z
```

Binary Search Tree Operation: Insertion

```
class  BST{
    public:
    int Key;
    BST * left;
    BST * right;
    BST(){Key = 0;}
    BST(int key) {Key = key;};

    BST* Insert(BST * root,
int key);
};
```

```
BST * BST::Insert(BST * root, int key)
{
    if(root == nullptr)
    {
      return new BST(key);     }
    if(key < root->Key)     {
        root->left =
Insert(root->left,key);}
    else{
        root->right =
Insert(root->right,key);
    }
return root;
}
```

Binary Search Tree Operation: Insertion



https://visualgo.net/en/bst

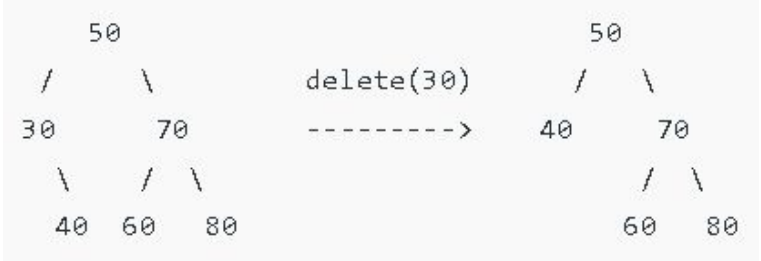
Assignment 8: Exercise 1

Add to `BST` class shown in this session **any** traversal algorithm as a new method and call this new method in `main` to print the tree nodes.

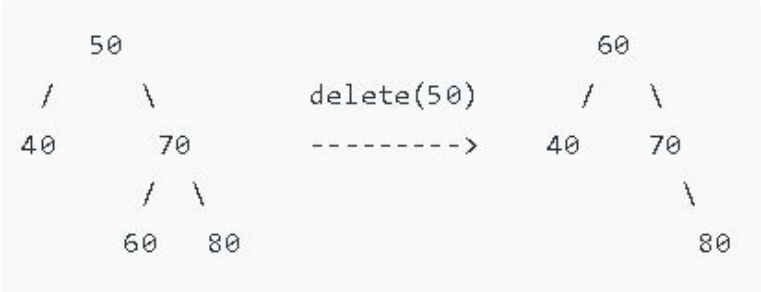
Binary Search Tree Operation: Deletion

Case 1

Case 2



Case 3



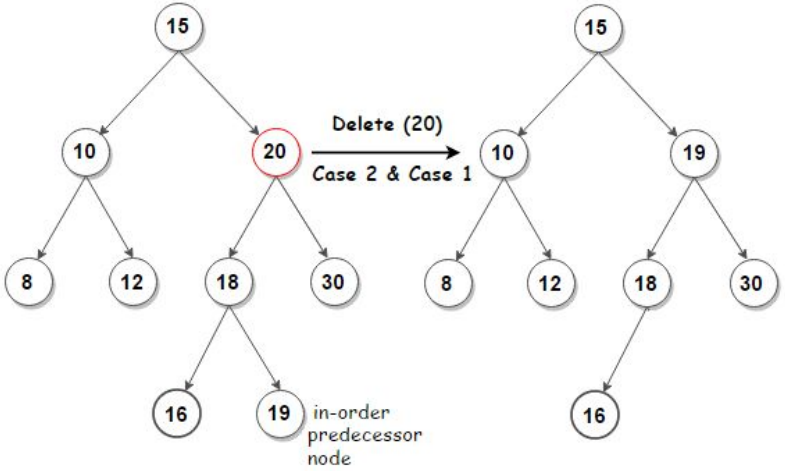
https://www.geeksforgeeks.org/binary-search-tree-set-2-delete/

Binary Search Tree Operation: Deletion

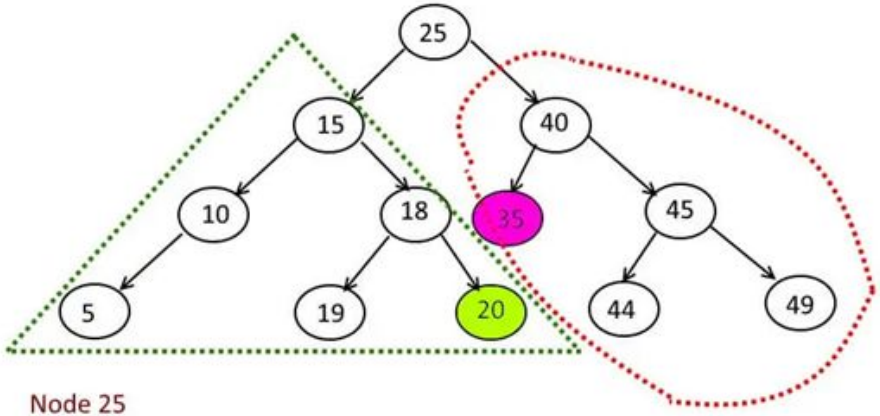
```
struct node* deleteNode(struct node* root, int key)
{   if (root == NULL) return root;

    if (key < root->key)
        root->left = deleteNode(root->left, key);

    else if (key > root->key)
        root->right = deleteNode(root->right, key);
     else {
        if (root->left==NULL and root->right==NULL)
            return NULL;

        else if (root->left == NULL) {
            struct node* temp = root->right;
            free(root);
            return temp;
        }
        else if (root->right == NULL) {
            struct node* temp = root->left;
            free(root);
            return temp; }

        struct node* temp = minValueNode(root->right);

        root->key = temp->key;
        root->right = deleteNode(root->right, temp->key);
    }
    return root;
}
```


https://www.geeksforgeeks.org/binary-search-tree-set-2-delete/

Binary Search Tree Operation: Deletion with Pred/Successor



What is Pred/Successor ?



Predecessor of node 25 will the right mode element in the left subtree.

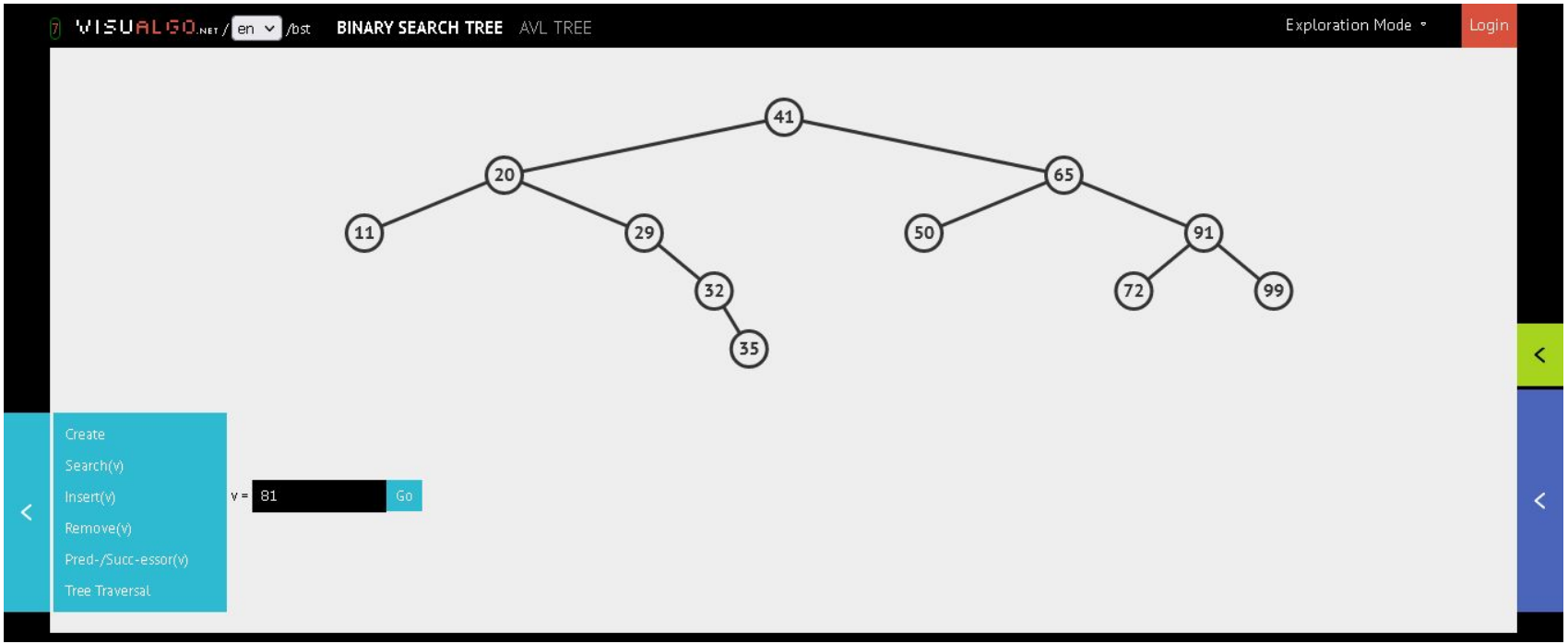
which is 20

Successor of node 25 will the left most element in the right subtree

which is 35

```
    50                              60
  /    \          delete(50)      /   \
40      70        --------->    40     70
       /  \                           \
      60   80                          80
```

Binary Search Tree Operation: Deletion



https://visualgo.net/en/bst

Assignment 8: Exercise 2

Add to `BST` class shown in this session **the delete algorithm** as a new method. Delete a node and call then the print method using traversing algorithm to show that it was removed correctly.

Example:

```
root = T.Insert(root,100);
T.Insert(root,20);
T.Insert(root,10);
T.Insert(root,500);
T.Insert(root,30);

T.delete(100);

T.print();
```

Lab Project

Write a program do the following:

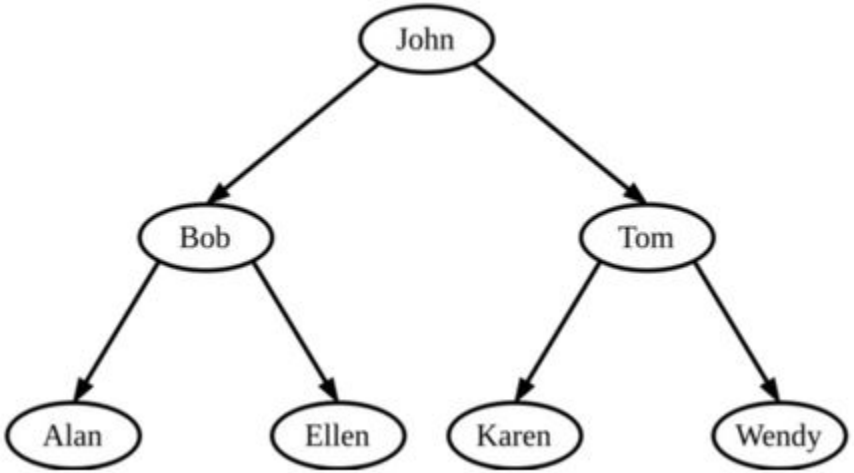
- Takes a Goodreads dataset and extract author names and book titles. (20%)
- Build a BST for authors and another BST for books title. (30%)
- Fetch a list of all books that start with the query string .i.e (an introduction) to fetch all books start with this string. (20%)
- The same thing while searching for authors name. (20%)

Note: An extra marks will be assigned for good code design, documenting the code through comments, and making a simple GUI. (10%)

Dataset Link: https://www.kaggle.com/jealousleopard/goodreadsbooks

Lab Project: Second week resources

lexicographical order




```
#include <iostream>
#include <string>
#include <vector>
#include <algorithm>
using namespace std;

int main()
{
std::vector<std::string> stringarray;
stringarray.push_back("Ellen");
stringarray.push_back("Tom");
stringarray.push_back("Wendy");
stringarray.push_back("Alan");
stringarray.push_back("Karen");
stringarray.push_back("Bob");
stringarray.push_back("John");

std::sort(stringarray.begin(), stringarray.end());

for (std::vector<std::string>::iterator it = stringarray.begin() ; it != stringarray.end(); ++it)
  cout<<*it<<",";
return 0;
}
```

https://www.geeksforgeeks.org/construct-all-possible-bsts-for-keys-1-to-n/