```cpp
#include <iostream>
using namespace std;

class Node
{
public:
    int data;
    Node *left;
    Node *right;

    Node(int value)
    {
        data = value;
        left = right = nullptr;
    }
};

class BST
{
public:
    Node *root;
    BST()
    {
        root = nullptr;
    }
    Node *findmin(Node *r)
    {
        if (r == nullptr)
        {
            return nullptr;
        }
        else if (r->left == nullptr)
        {
            return r;
        }
        else
        {
            return findmin(r->left);
        }
    }
    Node *findmax(Node *r)
    {
        if (r == nullptr)
        {
            return nullptr;
        }
        else if (r->right == nullptr)
        {
            return r;
        }
        else
        {
            return findmax(r->right);
        }
    }
    Node *Delete(Node *r, int key)
    {
        if (r == nullptr) // Empty tree
        {
            return nullptr;
        }
        if (key < r->data) // item is in Left subtree
        {
            r->left = Delete(r->left, key);
        }
        else if (key > r->data) // item is in right subtree
        {
            r->right = Delete(r->right, key);
        }
        else
        {
            if (r->left == nullptr && r->right == nullptr) // leaf node
            {
                r = nullptr;
            }
            else if (r->left != nullptr && r->right == nullptr) // one child on the left
            {
                r->data = r->left->data;
                delete r->left;
                r->left = nullptr;
            }
            else if (r->left == nullptr && r->right != nullptr) // one child on the right
            {
                r->data = r->right->data;
                delete r->right;
                r->right = nullptr;
            }
            else
            {
                Node *max = findmax(r->left); // two children
                r->data = max->data;
                r->left = Delete(r->left, max->data);
            }
        }
        return r;
    }
};
```