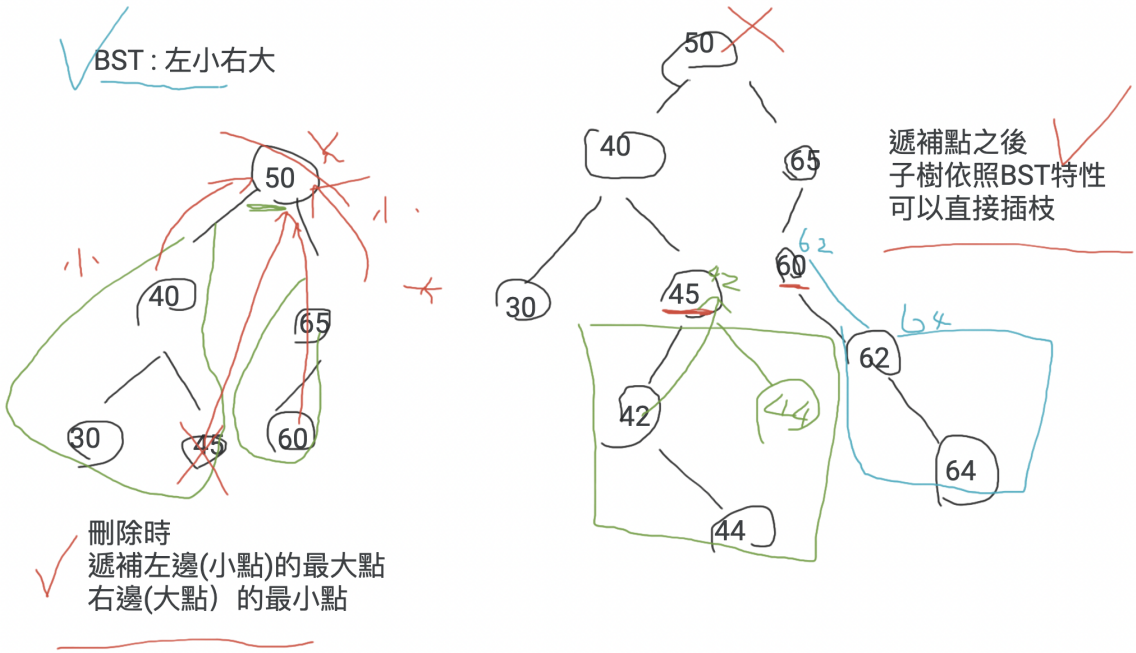
Phill-DS-0207

Binary Search Tree

- 整合
- remove




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

struct TreeNode{
  int data;
  TreeNode* left;
  TreeNode* right;

  TreeNode(int value): data(value), left(nullptr), right(nullpt
};
```

```
class BinarySearchTree{
  private:
    TreeNode *root;

    TreeNode* insert(TreeNode* node, int val){ //名字
      //檢查
      if(node==nullptr)
        return new TreeNode(val);
      if(val < node->data){
        node->left = insert(node->left, val); //向左遞迴
      }
      else if(val > node->data){
        node->right = insert(node->right, val);
      }
      return node;
    }

  TreeNode* remove_tree(TreeNode* node, int val){
    if(node==nullptr)
      return nullptr;
    if(val < node->data){
      node->left = remove_tree(node->left, val);
    }
    else(val > node->data){
      node->right = remove_tree(node->right, val);
    }
    else{//是我
      if(node->left==nullptr){
        TreeNode* temp = node->right;
        delete node;
        return temp;
      }
      else if(node->right==nullptr){
        TreeNode* temp = node->left;
        delete node;
```

```
        return temp;
      }
      TreeNode* temp = minValueNode(node->right); //右邊最小的
      node-> data = temp -> data;
      node->right= remove_tree(node->right, temp->data); //遞迴
    }
    return node;
  }

  minValueNode(){

  }

  public: //abstract 精神
    BinarySearchTree() : root(nullptr){}

    void insert(int value){ //正式
      root = insert(root, value);
    }

    void remove_tree(int value){
      root = remove_tree(root, value);
    }

    //~BinarySearchTree(){
    //  destroyTree(root);
    //}
};

int main()
{
    //.insert()
    return 0;
}
```

minValueNode() 尚未完成

B F S測資 insert