

# 二元搜尋樹

今天我們要練習二元搜尋樹的幾項基本功能：

1. 創建二元樹  
在函式內配置記憶體，並完成初始化。
2. 搜尋二元樹內的某個點  
輸入一個 key 值，搜尋二元樹中 key 值相同的節點，找到的話，回傳節點位址。
3. 在二元樹增加某個點  
輸入一個 key 值，根據二元樹的規則，插入二元樹。
4. 刪除某個二元樹的點  
輸入一個 key 值，根據二元樹的規則，刪除。  
**刪除規則：優先選擇左子樹最大值。**
5. 以 preorder 印出二元樹。
6. 清空並釋放二元樹記憶體  
走訪整個二元樹，將全部節點的記憶體都釋放。

簡單介紹：

```
p_bst_t bst = (p_bst_t)malloc(sizeof(bst_t));
```

宣告型態	名稱	強制轉換	malloc	記憶體大小
------	----	------	--------	-------

```
p_node_t root = (p_node_t)malloc(sizeof(node_t));
```

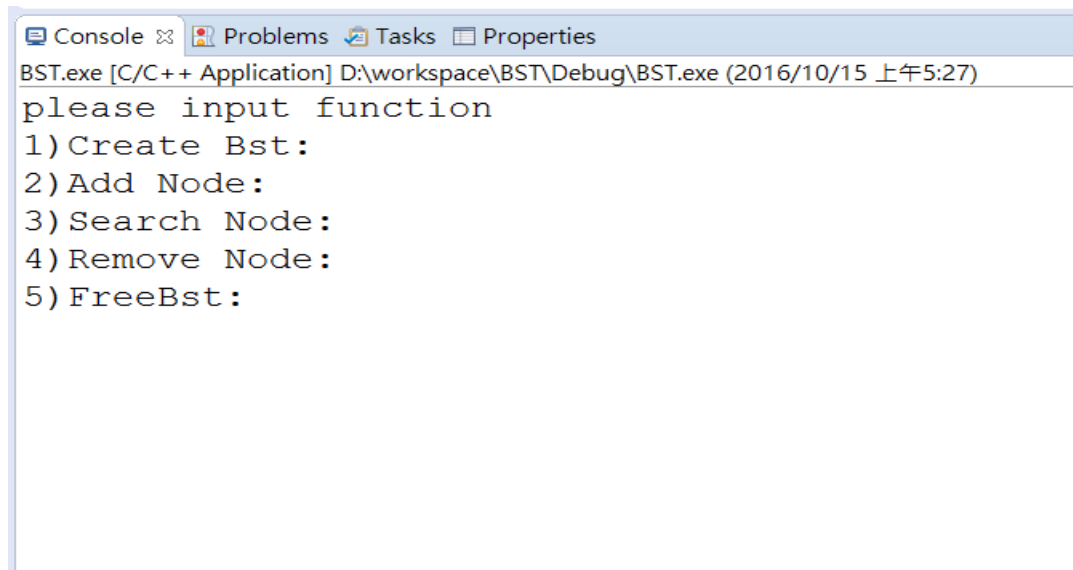
宣告型態	名稱	強制轉換	malloc	記憶體大小
------	----	------	--------	-------

## .h 檔

```
#include <stdbool.h>
#include <stdlib.h>
typedef struct node{
    struct node* lchild;
    struct node* rchild;
    int key;
}node_t, *p_node_t;
//一般型態的節點，以及指標型態的節點。內含 key 值，子樹的位址。
typedef struct{
    p_node_t root;
}bst_t, *p_bst_t;
//一般結構型態的二元樹，以及指標型態的二元樹。
p_node_t CreateNode(int);
//傳入 KEY 值，建立一個新的節點，然後回傳位址。
p_bst_t CreateBst(int); //建立一個新的二元樹，配置好記憶體，完成初始化，傳入 key 值，然後回傳位址。
bool Add(p_node_t, int); //在二元樹中，插入特定 KEY 值的節點。回傳表示成功失敗與否。
p_node_t Search(p_node_t, int); //在二元樹中尋找，符合特定 key 值的節點，符合的話回傳節點位址，不符合的話，回傳 NULL。
p_node_t ModifySearch(p_node_t, int); //修改過的 search，回傳可以接節點的位址
bool Remove(p_node_t, int); //在二元樹中，刪除某個特定 key 值的節點。回傳表示成功失敗與否。
void ListPreorder(p_node_t); //以 preorder 方式，將二元樹中的 key 值及位址印出。
void FreeBst(p_node_t); //將二元樹刪除並釋放記憶體

void preorder(p_node_t root) { //為了方便測試，附上 preorder 程式碼
    if (root) {
        printf("%d:(%p) ", root->key, root);
        Preorder(root->lchild);
        Preorder(root->rchild);
    }
}
```

main 檔完成後，應顯示為以下。



```
Console Problems Tasks Properties
BST.exe [C/C++ Application] D:\workspace\BST\Debug\BST.exe (2016/10/15 上午5:27)
please input function
1) Create Bst:
2) Add Node:
3) Search Node:
4) Remove Node:
5) FreeBst:
```

功能 1：輸入一個 root 後，產生 BST。

再以 preorder 印出 key 值及位址。

注意:(Create 一次就好了，之後輸入都是輸入其他功能)。

功能 2：輸入一個 node key 後，插入二元樹。

再以 preorder 印出 key 值及位址。

失敗印出 error。

功能 3：輸入一個 node key 後，尋找。

以 %p 印出節點所在記憶體位址。

成功:find in 位址。

失敗:error。

功能 4：輸入一個 node key 後，刪除。

再以 preorder 印出 key 值及位址。

失敗印出 error。

功能 5：free 整個二元樹。

功能 1 2 3 (4 分)功能 4 5 (4 分)

功能不完整(ex:沒 error 判斷)

0 分!!!!

測資：依次新增 50(root) 100 25 75 40 15 5

建樹：

輸入 1 enter 50 enter (只建樹一次)

印出：50:(00BF1278)

新增：

輸入 2 enter 25 enter

印出：50:(00BF1278) 25:(00BF1290)

```
Console Problems Tasks Properties
BST_Recursive.exe [C/C++ Application] D:\workspace\BST_Recursive\Debug\BST_Recursive.exe (2016/10/19 下午4:12)
please input function
1)Create Bst:
2)Add Node:
3)Search Node:
4)Remove Node:
5)FreeBst:
1
please input your root key:50
50:(00A31248)
please input function
1)Create Bst:
2)Add Node:
3)Search Node:
4)Remove Node:
5)FreeBst:
2
please input your node key:25
50:(00A31248) 25:(00A31220)
please input function
1)Create Bst:
2)Add Node:
3)Search Node:
4)Remove Node:
5)FreeBst:
```

以此類推，最後會印出以下

印出：50:(00BF1278) 25:(00BF1290) 15:(00BF12D8)

5:(00BF12F0) 40:(00BF12C0) 100:(00BF1220) 75:(00BF12A8)

位址數字參考用，位址相對關係對就好。

移除：5

輸入 4 enter 5 enter

印出：50:(00BF1278) 25:(00BF1290) 15:(00BF12D8)

40:(00BF12C0) 100:(00BF1220) 75:(00BF12A8) 位址參考用

移除：50

輸入 4 enter 50 enter

印出：40:(00BF1278) 25:(00BF1290) 15:(00BF12D8)

100:(00BF1220) 75:(00BF12A8)

40 變新 root。而不是 75。

移除：25

輸入 4 enter 25 enter

印出：40:(00BF1278) 15:(00BF1290) 100:(00BF1220)

75:(00BF12A8)

原本 25 的地方變成 15 去接了。

搜尋：

40 輸入 3 enter 40 enter

印出：find in (00BF1278)要跟前面一樣。

**p. S** .h 檔跟程式碼僅供參考，歡迎多加修改。