壹、 題目定義

延續先前作業的程式，增加支援 **Tree** 的操作，樹的每一個節點 (node) 儲存 key 資訊，節點內使用指位器指向 value 資料。指令自行設計，功能為可以存取多棵樹，對任一棵樹新增/搜尋/刪除節點。

貳、 想法及做法

將自己寫的函式連結 cjson 函式庫並新增樹的功能，命名為 "hw4.h"形成函式庫 hw4lib.a，並導入 "hw4.c"，在 hw4.c 程式中呼叫 addTree(新增樹), addNodeToTree(新增樹節點), printAllTrees(輸出所有樹), searchNodeInAllTrees(搜尋節點), deleteNode(刪除節點), deleteTree(刪除整個樹), freeTree(歸還樹空間)的函式，此次繳交 homework4.c ,hw4.h , hw4lib.a 與此份解釋文檔共四個檔案，以下解釋程式碼:

hw4.h

第 6 行: 使用 cJSON 函式庫資源

20~23 行: 讓每個節點有樹的結構

31~54 行: **getNewJsonValue** 函式將 value 用 cJSON 函式讀進去

56~87 行: **insertNodeToTree** 函式將資料以二元樹的型態插入

74~85 行: **addTree** 函式利用以上兩個函式，新增一顆新的樹

114~128 行: **printAllTrees** 函式呼叫 **printTree** 函式輸出所有樹節點

130~149 行: **addNodeToTree** 函式用於新增節點到指定的樹中

151~159 行: **createNodeFromUserInput** 函式用於初始化節點

161~183 行: **searchNodeInAllTrees** 函式呼叫 **searchNodeInTree** 函式用遞迴寫法來搜尋節點

185~224 行: **deleteNode** 函式用於刪除節點，特別的是過程中會用到 **minValueNode** 函式以向左遍歷的方式來尋找最小的節點

226~233 行: **deleteTree** 函式用於刪除該整棵樹

235~242 行: **freeTree** 函式用於清除樹的記憶體空間

參、 編譯結果

```
weijhih@weijhih-VirtualBox:~/桌面/homework/hw4$ ./hw4
SET,GET,LPUSH,RPUSH,LPOP,RPOP,LLEN,LRANGE,DELETE,INSERT,UPGRADE,FREE
Enter 'add' to add a new tree, 'addnode' to add a new node, 'print' to print all trees,
Enter 'search' to search for a node, 'delete' to delete node, 'exit' to quit: add
Enter key-value pairs for the new tree (type 'exit' to stop):
Enter key: hh
Enter value in JSON format (type 'exit' to stop):
Example JSON format: {"value1": "string1", "value2": 42, "value3": [1, 2, 3]}
Enter value: {"v1": "ry","v2": 22}
Enter key: exit
SET,GET,LPUSH,RPUSH,LPOP,RPOP,LLEN,LRANGE,DELETE,INSERT,UPGRADE,FREE
Enter 'add' to add a new tree, 'addnode' to add a new node, 'print' to print all trees,
Enter 'search' to search for a node, 'delete' to delete node, 'exit' to quit: print
Tree 1:
Key: hh, Value: {
        "v1":   "ry",
        "v2":   22
}

SET,GET,LPUSH,RPUSH,LPOP,RPOP,LLEN,LRANGE,DELETE,INSERT,UPGRADE,FREE
Enter 'add' to add a new tree, 'addnode' to add a new node, 'print' to print all trees,
Enter 'search' to search for a node, 'delete' to delete node, 'exit' to quit: addnode
Add to which tree: hh
Enter key: io
Enter value in JSON format (type 'exit' to stop):
Example JSON format: {"value1": "string1", "value2": 42, "value3": [1, 2, 3]}
Enter value: {"vv1":36,"vv2":"uww"}
```

```
SET,GET,LPUSH,RPUSH,LPOP,RPOP,LLEN,LRANGE,DELETE,INSERT,UPGRADE,FREE
Enter 'add' to add a new tree, 'addnode' to add a new node, 'print' to print all trees,
Enter 'search' to search for a node, 'delete' to delete node, 'exit' to quit: print
Tree 1:
Key: hh, Value: {
        "v1":   "ry",
        "v2":   22
}
Key: io, Value: {
        "vv1":  36,
        "vv2":  "uww"
}

SET,GET,LPUSH,RPUSH,LPOP,RPOP,LLEN,LRANGE,DELETE,INSERT,UPGRADE,FREE
Enter 'add' to add a new tree, 'addnode' to add a new node, 'print' to print all trees,
Enter 'search' to search for a node, 'delete' to delete node, 'exit' to quit: search
Enter key to search: io
Found in 1. treeNode found. Key: io, Value: {
        "vv1":  36,
        "vv2":  "uww"
}
SET,GET,LPUSH,RPUSH,LPOP,RPOP,LLEN,LRANGE,DELETE,INSERT,UPGRADE,FREE
Enter 'add' to add a new tree, 'addnode' to add a new node, 'print' to print all trees,
Enter 'search' to search for a node, 'delete' to delete node, 'exit' to quit: delete
Enter 'node' to delete a node, 'tree' to delete a tree: node
Enter key of the node to delete: hh
SET,GET,LPUSH,RPUSH,LPOP,RPOP,LLEN,LRANGE,DELETE,INSERT,UPGRADE,FREE
Enter 'add' to add a new tree, 'addnode' to add a new node, 'print' to print all trees,
```

```
Enter 'add' to add a new tree, 'addnode' to add a new node, 'print' to print all trees,
Enter 'search' to search for a node, 'delete' to delete node, 'exit' to quit: print
Tree 1:
Key: io, Value: {
        "vv1":  36,
        "vv2":  "uww"
}

Tree 2:
Key: tr, Value: {
        "vvv1": "dii",
        "vvv2": [4, 2, 8]
}

SET,GET,LPUSH,RPUSH,LPOP,RPOP,LLEN,LRANGE,DELETE,INSERT,UPGRADE,FREE
Enter 'add' to add a new tree, 'addnode' to add a new node, 'print' to print all trees,
Enter 'search' to search for a node, 'delete' to delete node, 'exit' to quit: delete
Enter 'node' to delete a node, 'tree' to delete a tree: tree
Enter key of the tree to delete: io
SET,GET,LPUSH,RPUSH,LPOP,RPOP,LLEN,LRANGE,DELETE,INSERT,UPGRADE,FREE
Enter 'add' to add a new tree, 'addnode' to add a new node, 'print' to print all trees,
Enter 'search' to search for a node, 'delete' to delete node, 'exit' to quit: print
Tree 1:

Tree 2:
Key: tr, Value: {
        "vvv1": "dii",
        "vvv2": [4, 2, 8]
}
```

肆、 遇到的問題與解決辦法

遇到的問題	解決辦法
無法讀取 value 值	用 cJSON 的讀取函式並用該格式輸入
無法順利整合過往作業與此作業	進行多次測試與耐心 debug