

Binary Search Tree

巨資四B 袁嘉謙 05170229

BST原理

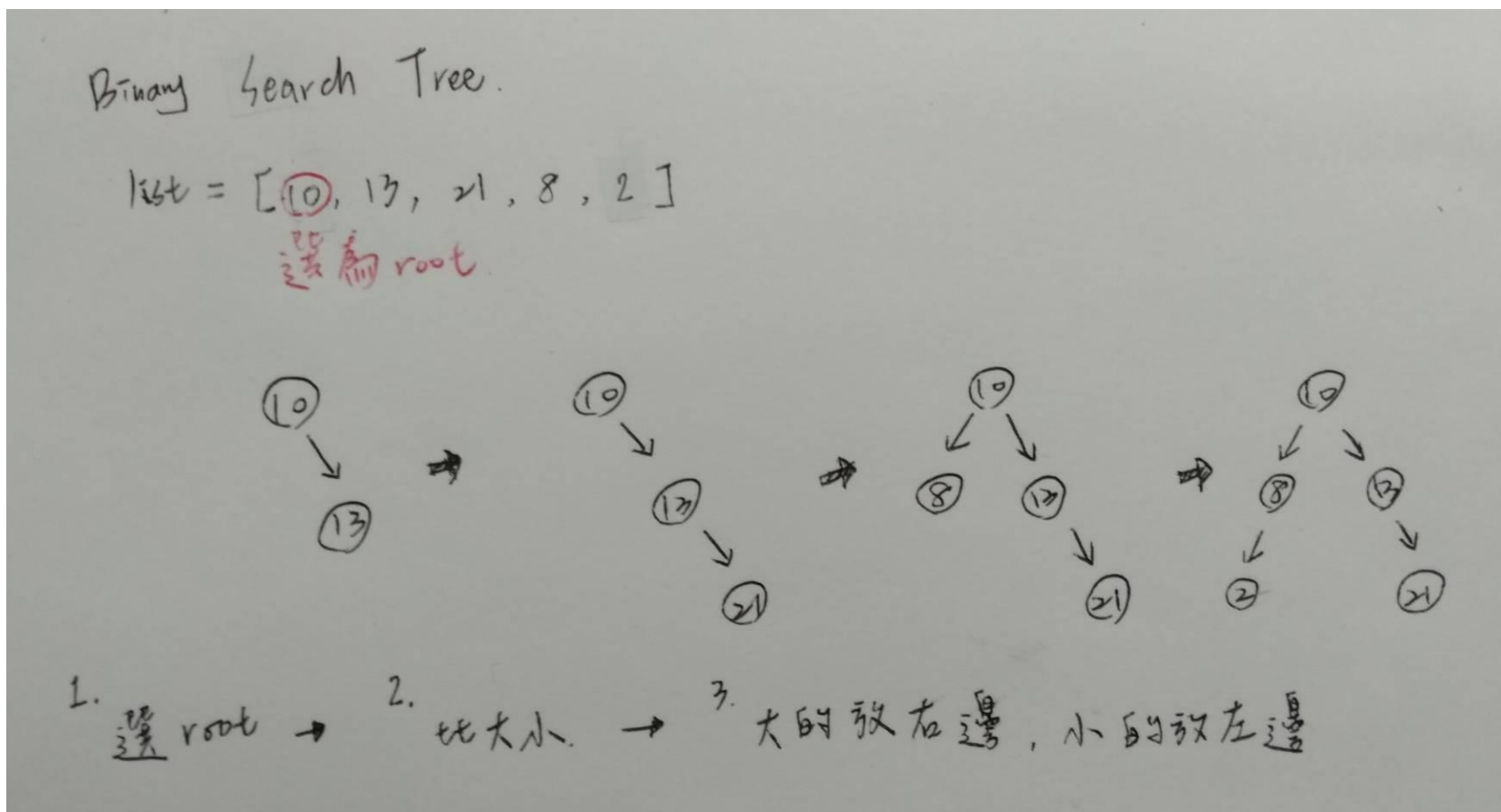
最後統整出binary search tree的以下幾點規則：

1. 先選定一個數當作根節點 (root)
2. 左邊的節點(node)必小於root，右邊必大於root
3. 各節點同上述規則繼續往下延伸形成子樹
4. 部分說法不能存在任何鍵值(key/value)相等的節點
>>待確認

參考資料：
<http://alrightchiu.github.io/SecondRound/binary-search-tree-introjian-jie.html>
<http://alrightchiu.github.io/SecondRound/binary-search-tree-searchsou-xun-zhi-liao-insertxin-zeng-zhi-liao.html>
<http://alrightchiu.github.io/SecondRound/binary-search-tree-sortpai-xu-deleteshan-chu-zhi-liao.html>

學習歷程

根據老師提供講義中的影片，
先筆記畫出簡易紀錄，
初步理解為與HeapSort相似的演算法。



學習歷程

此次怕受影響先不看其他的程式碼範例，
嘗試先依自己的邏輯編寫「新增資料」的功能。

改為 <=

```
def insert(self, root, val):  
    if root == None:  
        root = TreeNode(val)  
        return root  
    elif val < root.val:  
        if left == None:  
            left = TreeNode(val)  
            return left  
        else:  
            root = left  
            return self.insert(left, val)  
    elif val > root.val:  
        if right == None:  
            right = TreeNode(val)  
            return right  
        else:  
            root = right  
            return self.insert(right, val)
```

想法是如果沒有root就先設定root，
後面根據節點與root值比較大小，
並看其左右是否有其他值存在。

但我又想到一個問題，
若是有重複的值可能就無法解決？

從我看到的參考資料中理解為不能有相同的值，
可是看作業測試範本樹中有同樣的兩個 3，
且說明也要求把重複值放left。

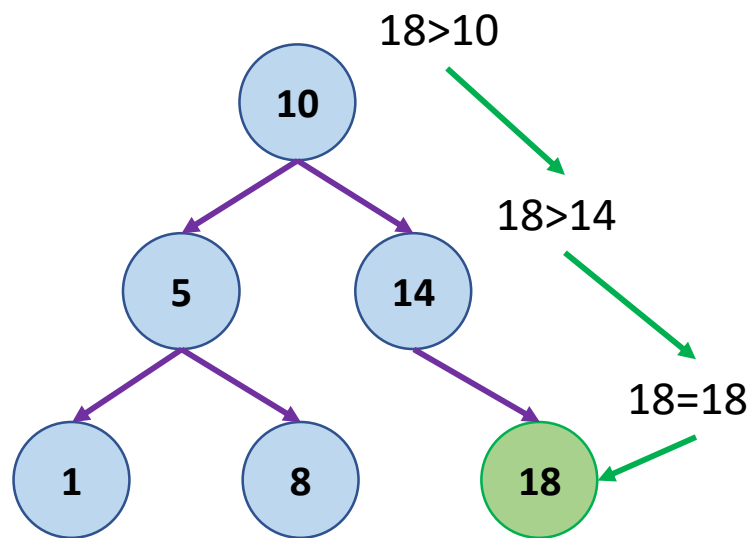
所以我先照要求設定，之後再確認是否為參考資料的
定義錯誤。

流程圖

Search

List = [10, 5, 14, 8, 1, 18]

↓
root



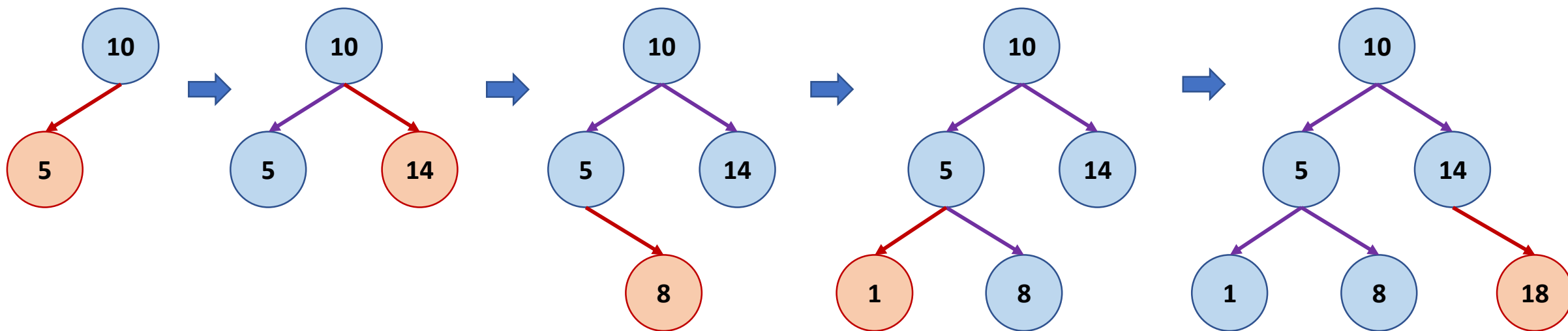
就很一般的找下去

流程圖

Insert

List = [10, 5, 14, 8, 1, 18]

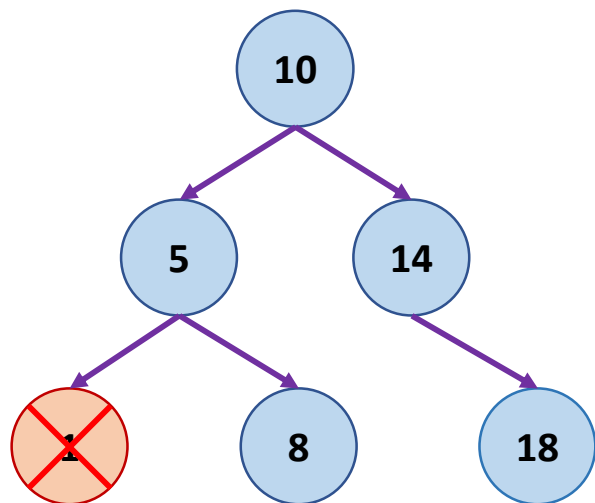
root



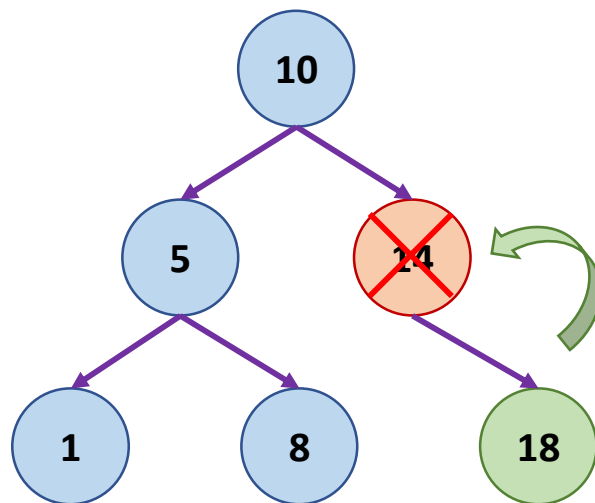
流程圖

Delete

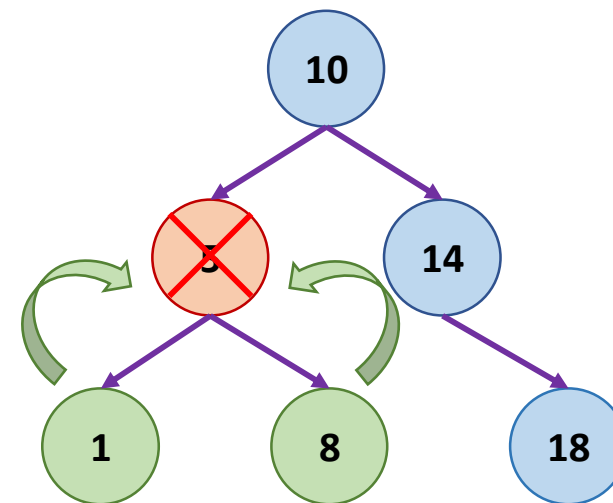
List = [10, 5, 14, 8, 1, 18]
↓
root



底下無Node，
則可直接刪除。



底下1個node，
刪除後把node上提。



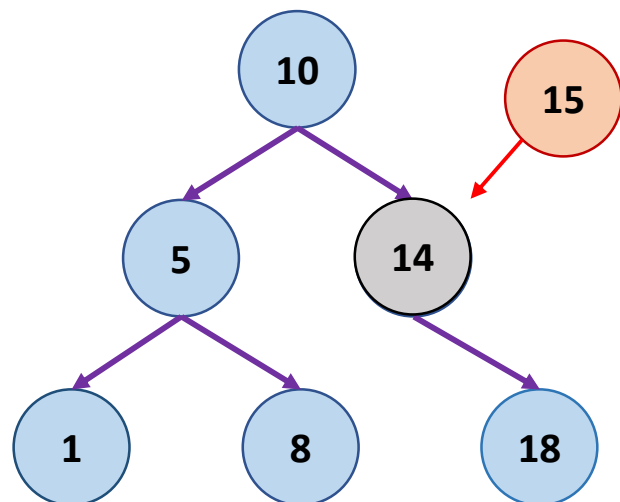
底下2個node，
刪除後把左邊最大
或右邊最小上提。

流程圖

Modify

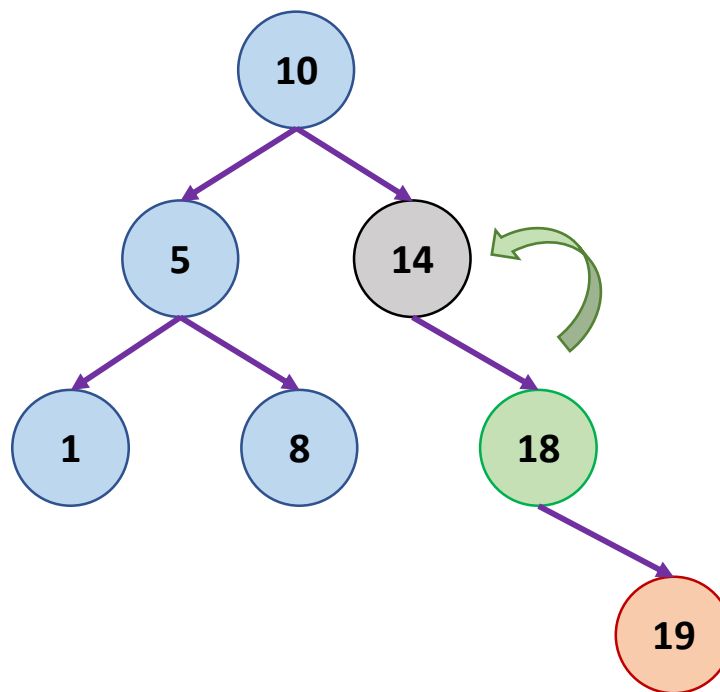
List = [10, 5, 14, 8, 1, 18]

root
↓



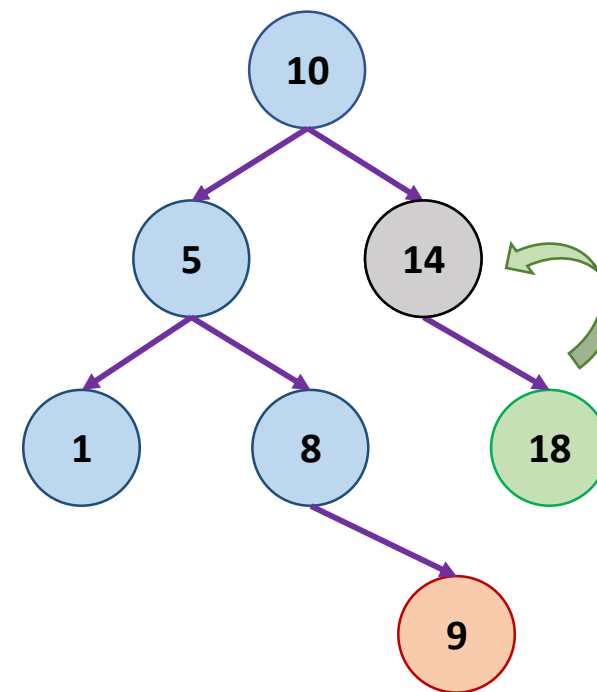
14改為15
 $10 < 15 < 18$ 符合BST

可直接修改



14改為19
 $19 > 18$

先刪除14把18上提
再新增19進去



14改為9
 $9 < 10$

先刪除14把18上提
再新增9進去