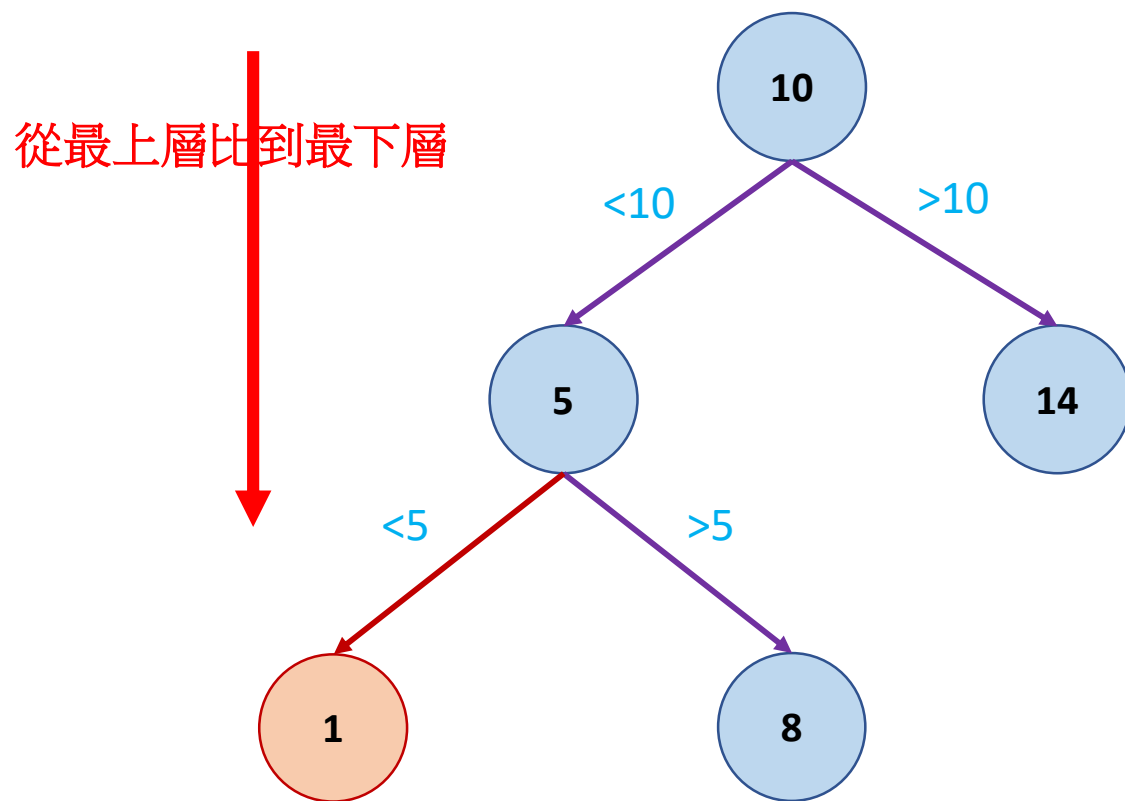


Binary Search Tree 功能

巨資四B 袁嘉謙 05170229

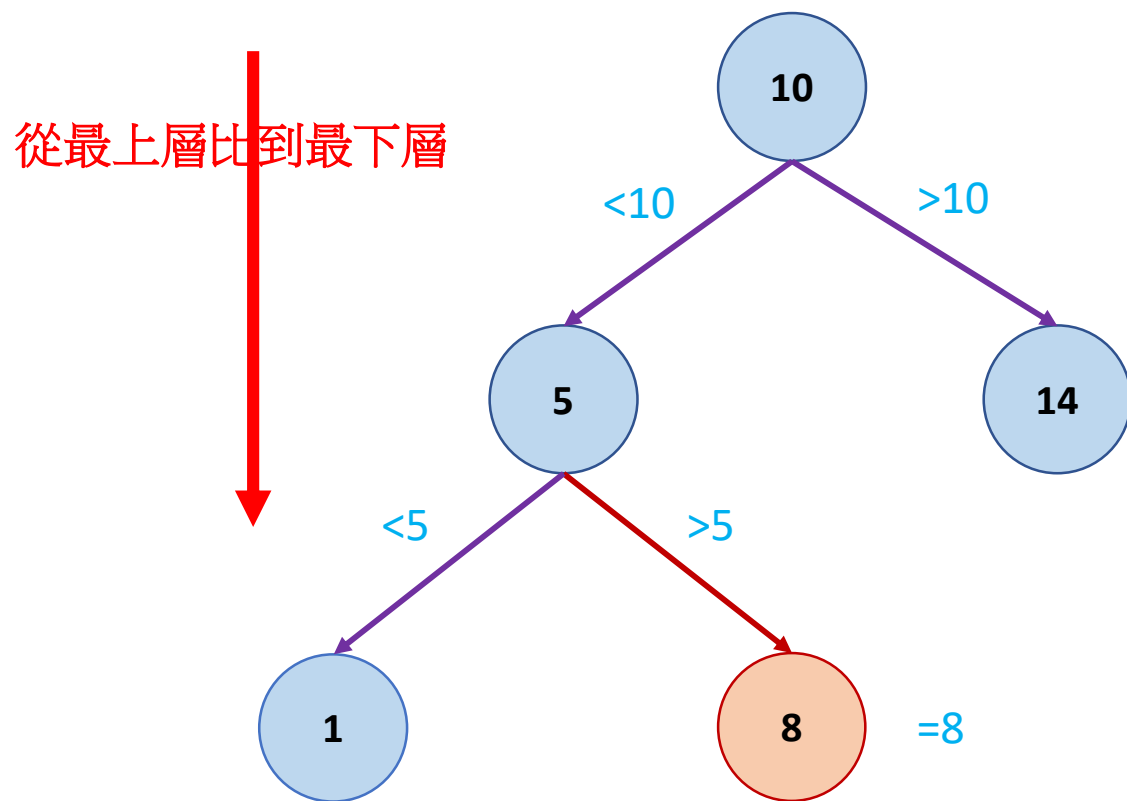
新增



1. 從最上方開始尋找空間。
2. 和節點相比：
小於節點往左邊比，大於節點往右邊比。
3. 沒有節點可比較時，加入值成為新節點。

Ex. 假設我要新增1，
先和10比較， $1 < 10$ 所以往左，
再和5比較， $1 < 5$ 再往左，
此時已無其他可比較，故可加入1。

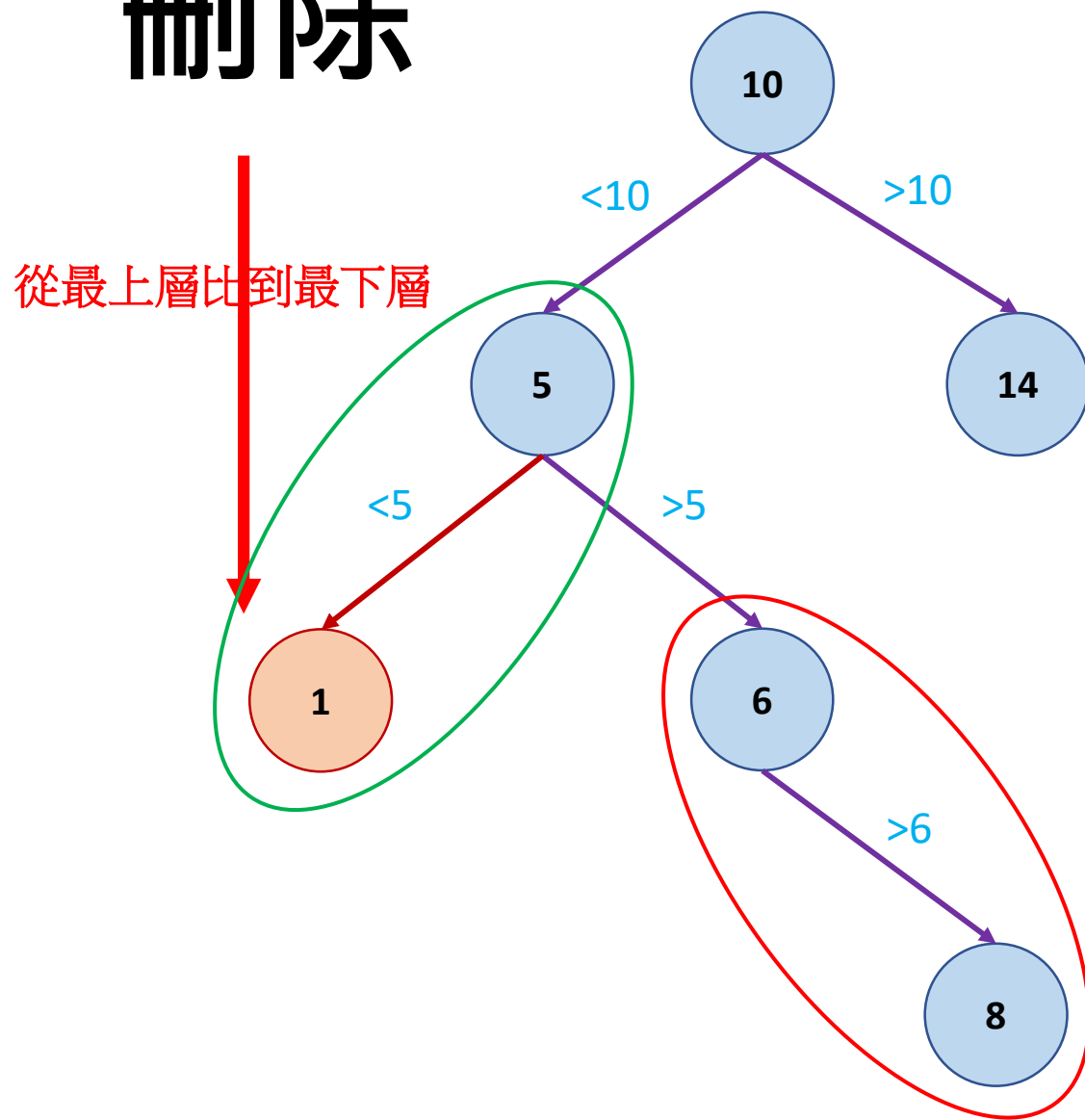
查詢



1. 從最上方開始尋找值。
2. 和節點相比：
小於節點往左邊比，大於節點往右邊比。
3. 直到找到相等的數值節點。

Ex. 假設我要找8，
先和10比較， $8 < 10$ 所以往左，
再和5比較， $8 > 5$ 所以往右，
此時 $8 = 8$ ，便找到查詢到值。

刪除



1. 從最上方開始尋找值。
2. 和節點相比：
小於節點往左邊比，大於節點往右邊比。
3. 找到相等的數值節點在往下看。
4. 若下層無其他節點則直接刪除值。

Ex. 假設我要刪除1，
先和10比較， $1 < 10$ 所以往左，
再和5比較， $1 < 5$ 再往左，
此時 $1 = 1$ 且下層無其他值，可直接刪除。

5. 若下層有一個節點，
則先刪除後把節點上移。

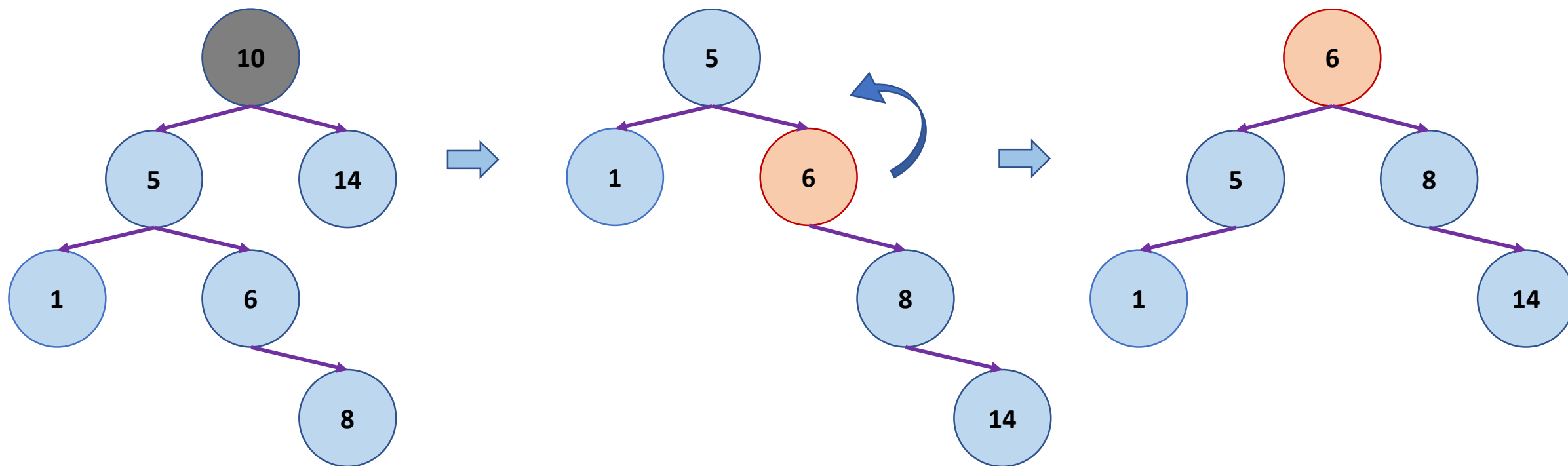
Ex. 假設我要刪除6，
找到6後因底下還有一個8，
故先刪除6，再把8往上移置6原先的位置。

4. 若下層有兩個節點，
先刪除後把左邊最大或右邊最小的節點上移。

Ex. 假設我要刪除5，
找到5後因底下還有1和6，
故先刪除5，再把1往上移置5原先的位置。

修改

1. 最基本為符合BST的原則。
2. 下圖為修改後數高 \leq 原樹高。



參考資料：<http://alrightchiu.github.io/SecondRound/red-black-tree-rotationxuan-zhuan.html>