陳俊宇 資訊工程系 二甲 F64051114

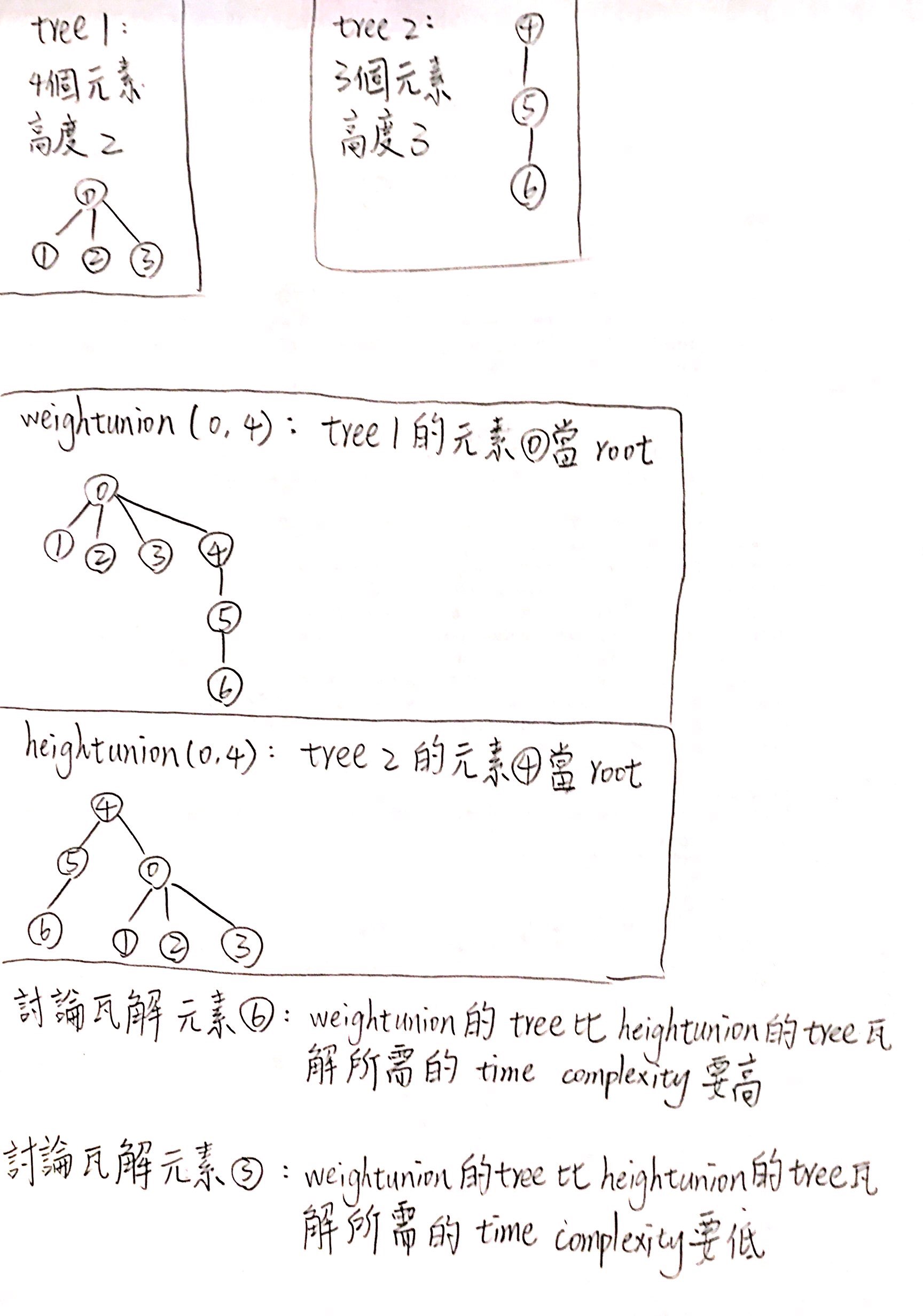
1. 此次的heightunion函數作法，我仿造原文書上weightUnion函數，差別在root的parent陣列值為樹高的負數，並按照題目要求，Union(0,1)、Union(0,2)、Union(0,3)、……、Union(0,n-1)是透過簡單for迴圈執行函數數次即可。

值得一提的是，如果兩棵樹原本高度相等時，union起來會直接變成樹高加一，除了這個特例之外，其他狀況都是union後的樹高肯定跟二樹中要當樹根者之樹高一樣。

還有一個特殊狀況，就是當要union的兩數本來就在同一棵樹中，則直接跳脫函數，我的做法是對兩數用for迴圈各自去尋找自己的root，如果root相等則表示兩數在同一聯集中，進而直接return不做此函數。又因為此功能，如果聯集的兩數不是樹根也沒關係，因為我進入函數之後，都是用for迴圈找到的root下去做聯集的後面運算。

1. 首先，我們必須知道collapsingFind函數的時間複雜度。假設我們要瓦解的元素，他跟root之間的樹高是h，則時間複雜度為O(h)，這表示要找的元素離root高度差越大，所花費時間越多。

個人覺得題目給的兩個situation難以討論和下結論，所以我自己假設了一個situation作分析，如下圖：



由我討論結果可看出，不同的瓦解元素與不同的樹高所做成的union，其瓦解結果(time complexity)都會不同。