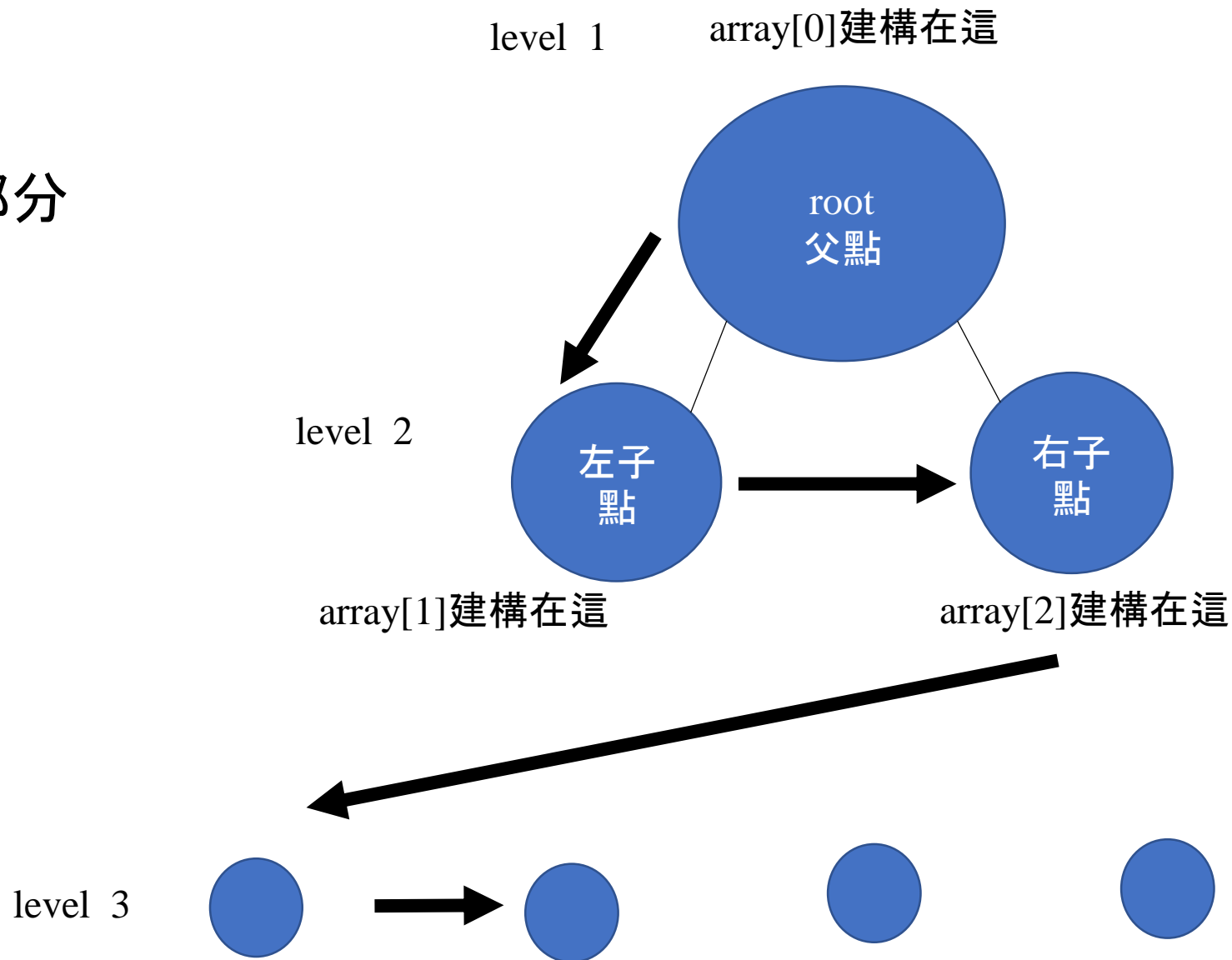


108資料結構

第三次作業

製作樹

- 右邊是名詞介紹
- “complete建構法”如黑色部分
- tree權重計算方式為，
該點值乘以該點所在的層數
ex: root為23則root層權重為
 $1 * 23 = 23$
- parent node: 有子點的節點
- leaf node: 沒有子點的節點



製作樹-1 (20%)

- 問題描述

- 給定一串數列array1[n], $1 \leq n \leq 100$ 。數列中的值皆大於等於0, 小於等於100。代表一個tree的後序。array1[n]中數字皆不同。
- 給定一串數列array2[n], $1 \leq n \leq 100$ 。數列中的值皆大於等於0, 小於等於100。代表一個tree的中序。array2[n]中數字皆不同。
- 根據後序中序建構出這個binary tree並且判斷此tree是不是一個"BadTree"。
- array1[n]與array2[n]中的數字重複但排列順序不一樣。
- "BadTree" 定義如下
 - 若不是空tree, 則每個node x。其 $\text{shortest}(\text{LeftChild}(x)) \geq \text{shortest}(\text{RightChild}(x))$ 。
 - $\text{shortest}(x)$ 為x到leaf node的路徑長, LeftChild(x)為x左子點, RightChild(x)為x右子點。
 - $\text{shortest}(x) = \{ 1 + \min\{\text{shortest}(\text{LeftChild}(x)), \text{shortest}(\text{RightChild}(x))\}$ 是 $\text{shortest}(x)$ 的計算方式。
 - $\text{shortest}(\text{LeftChild}(x)) = 0$ 當x沒有左子點。 $\text{shortest}(\text{RightChild}(x)) = 0$ 當x沒有右子點。
 - 各節點的值小於其子點的值。

- Input

- 一串數列(int)以空格間隔, 代表後序
- 一串數列(int)以空格間隔, 代表中序

- Output

- 輸出1或0 (int)(1代表是"BadTree", 0代表不是"BadTree")

製作樹-1

example

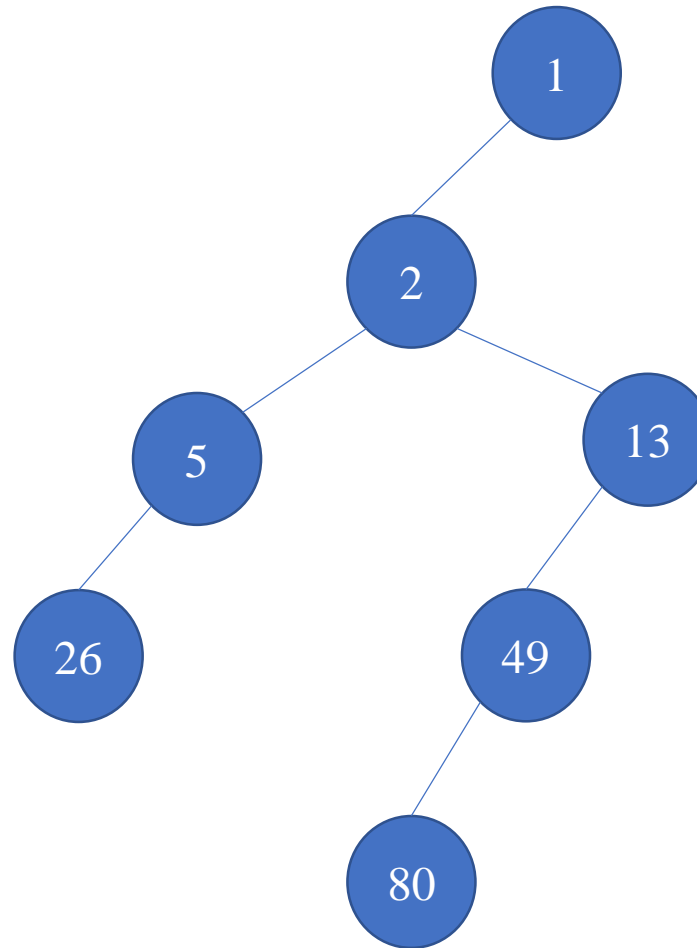
- Input

26 5 80 49 13 2 1

26 5 2 80 49 13 1

- Output

1



製作樹-1

額外測資

Input

17 20 9 11 4 19 10 6 5 3

17 9 20 4 11 3 19 10 5 6

Output

1

Input

9 7 16 17 4 2 6 8 3 1

9 7 4 17 16 2 1 6 3 8

Output

0

製作樹-2 (20%)

- 問題描述

- 給定兩串數列 $array1[n]$, $array2[n]$, $1 \leq n \leq 100$ 。數列中的值皆大於等於0, 小於等於100。代表tree1的**前序和中序**。**array1與array2要使tree1是一個"BadTree"**。array1中數字皆不同。
- 給定兩串數列 $array3[m]$, $array4[m]$, $1 \leq m \leq 100$ 。數列中的值皆大於等於0, 小於等於100。代表tree2的**後序和中序**。**array3與array4要使tree2是一個"BadTree"**。array3中數字皆不同。
- array1, 2中的數字重複但排列順序不一樣。array3, 4中的數字重複但排列順序不一樣。array1, 3數字不同。
- m不一定等於n, 建構出tree1和tree2並且把tree1與tree2 "merge"在一起。

merge(T1, T2) //T1是一個tree, T2是另外一個tree, 兩個merge在一起

Step1: 找出 $\min\{T1.root, T2.root\}$ 作為新樹的root //假如T1.root比較小

Step2: 以T1.root為新root且將T1**左子樹**保留 //保留T1的整棵左子樹

Step3: merge(T1**右子樹**, T2)

Step4: 檢查是否滿足"BadTree"的規定, 若不滿足則交換左右

- Input

- 一串數列(int)以空格間隔, 代表**前序**
- 一串數列(int)以空格間隔, 代表**中序**
- 一串數列(int)以空格間隔, 代表**後序**
- 一串數列(int)以空格間隔, 代表**中序**

- Output

- 把這個merge以後的tree以中序方式輸出(int)(以空格間隔)

製作樹-2 example

• Input

2 7 11 15 50 80

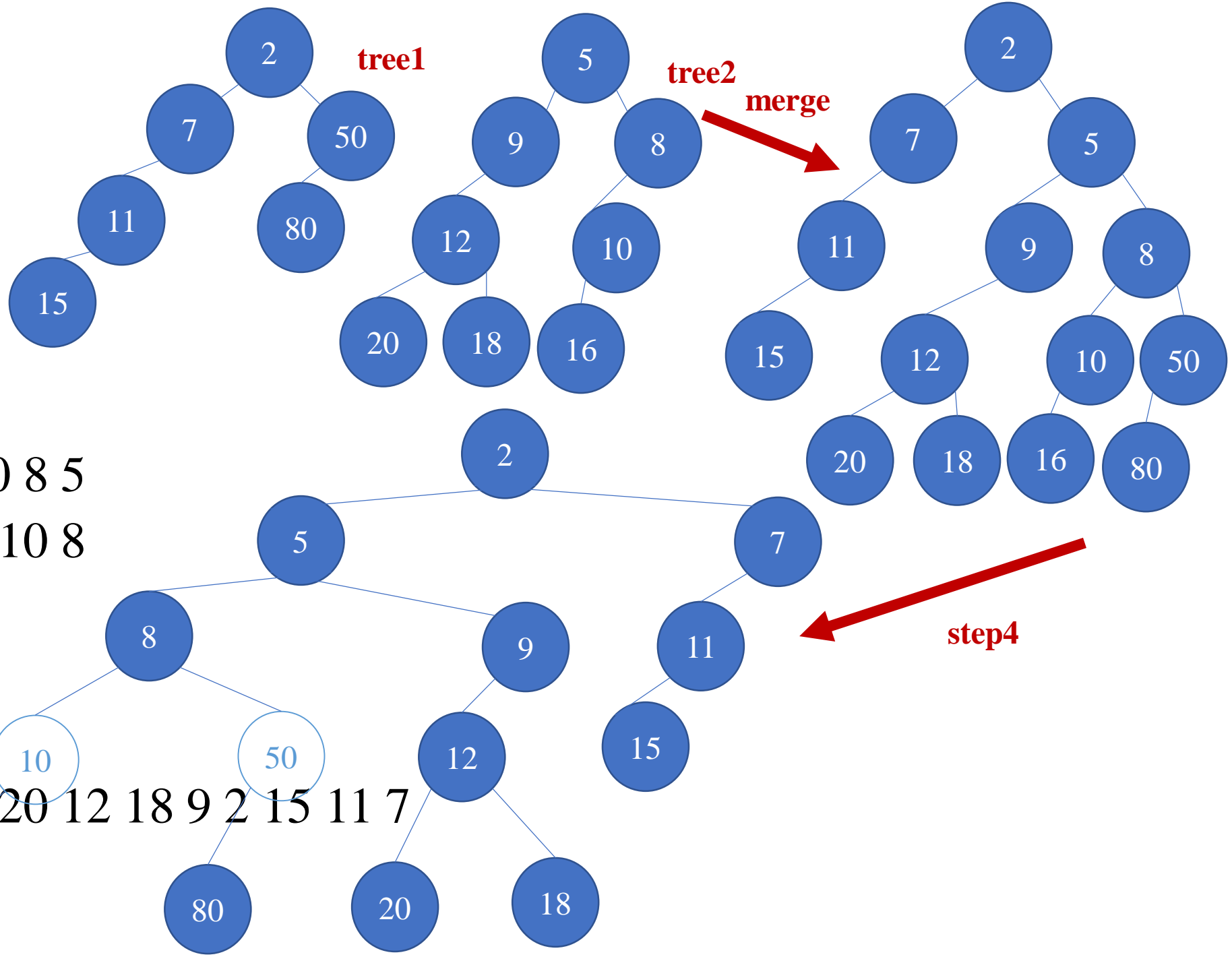
15 11 7 2 80 50

20 18 12 9 16 10 8 5

20 12 18 9 5 16 10 8

• Output

16 10 8 80 50 5 20 12 18 9 2 15 11 7



製作樹-2

example

- 怕大家不理解”BadTree”, 就不會寫這一題的input。所以我再寫出兩個input和output供大家測試自己的code。但這個不一定就是我的隱藏測資。

Input

1 3 6 7 4 8 9
6 3 7 1 8 4 9
11 10 5 2
11 10 5 2

Output

6 3 7 1 8 4 9 2 11 10 5

Input

21 46 59 77 83 23 25
83 77 59 46 21 25 23
60 47 20 14
60 47 14 20

Output

60 47 14 83 77 59 46 21 25 23 20

製作樹-3 (20%)

- 問題描述

- 給定一串數列 $array[n]$, $1 \leq n \leq 100$ 。數列中的值皆大於等於0, 小於等於100, 但 $array$ 中的每個數皆不相同。
- 我們有學過heap, 但一般情況下heap是以從上往下建立。而現在我要從下往上建立heap。
- 方式如下

- Step1: 先建構出complete binary tree(用“complete建構法”如第二頁)
- Step2: 從最後一個父點往回調整到root依序調整子樹成min-heap或max-heap。

調整方式是若這個父點與他的左右子點不符合規定則父點與左右子點中最大的或最小的(看是min-heap還是max-heap)交換值, 若符合規定則處理其他父點(跟它同一層左邊一個點或是上一層的最右邊點)。假設父點與左子點交換了位置則以令左子點為父節點重新做調整(右子點同理)。

- Input

- 輸入max或min(你想建的heap)(string)
- 一串數列 $array[n]$ (int)

- Output

- 把調整之後的tree以中序輸出(int)

製作樹-3 example

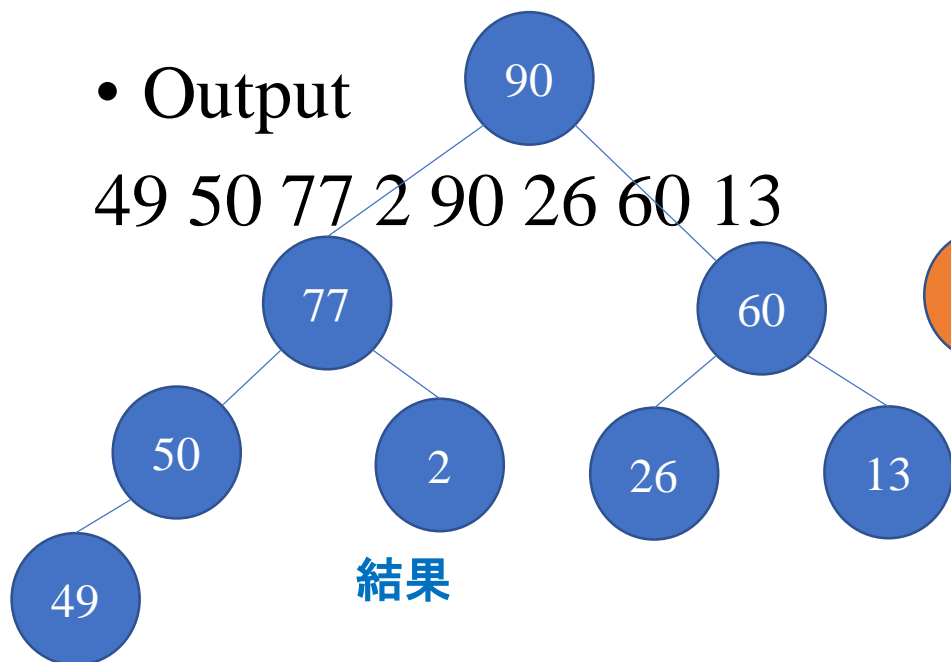
• Input

max

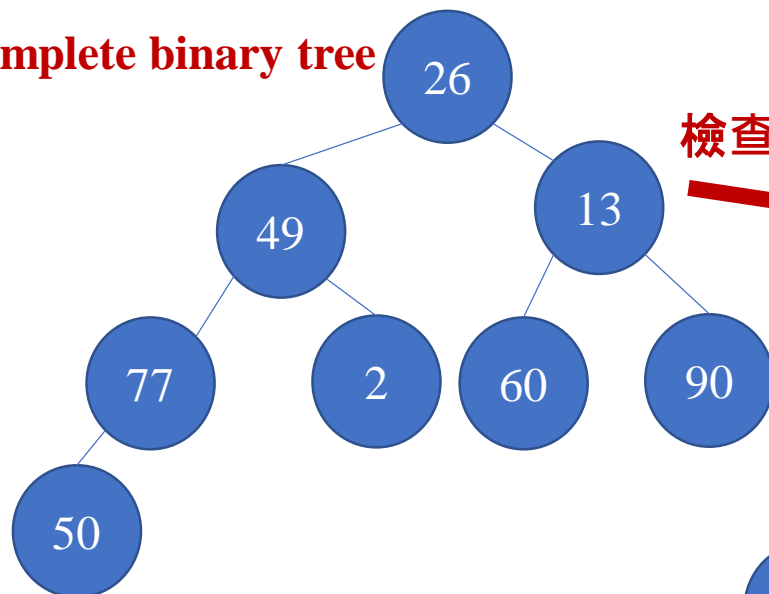
26 49 13 77 2 60 90 50

• Output

49 50 77 2 90 26 60 13



建好complete binary tree

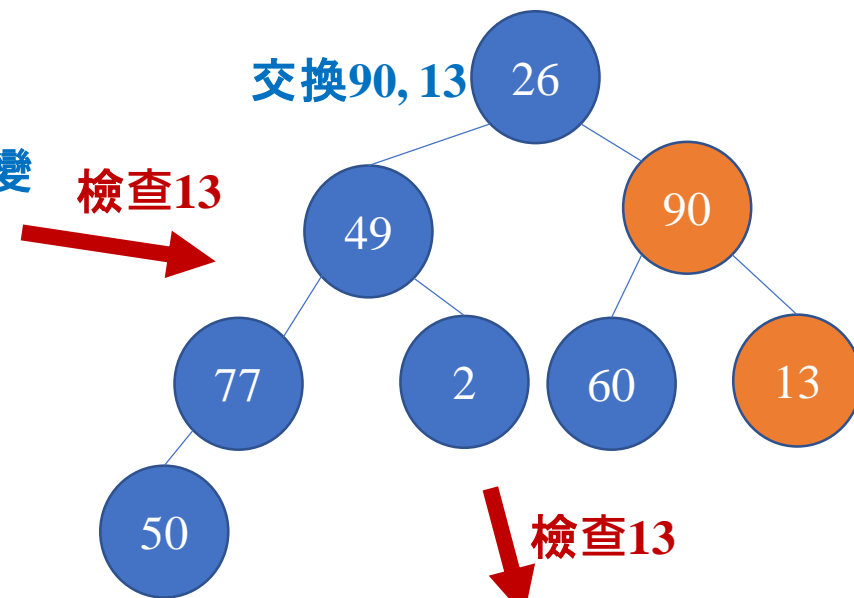


檢查77

tree不變

檢查13

交換90, 13



檢查13

tree不變

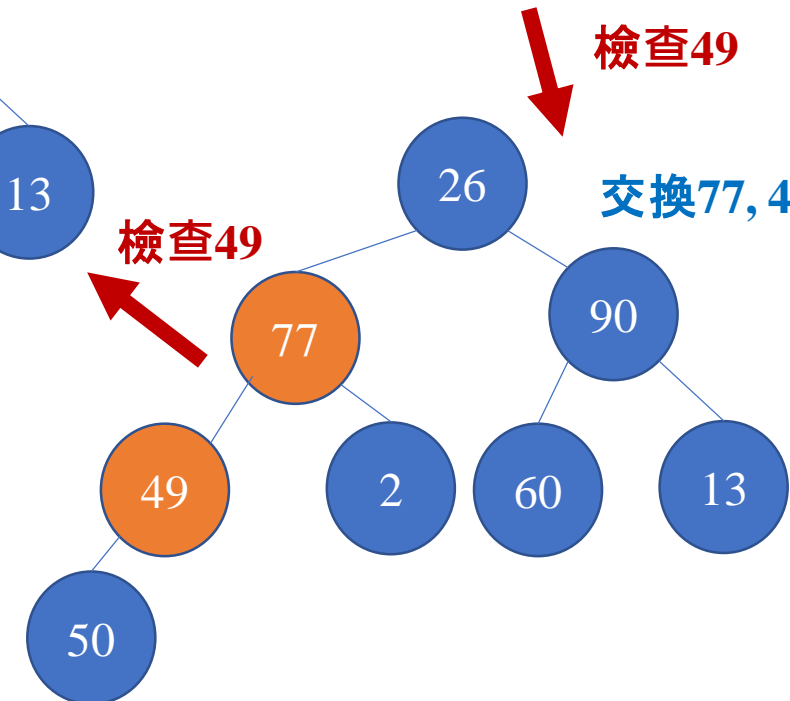
檢查49

交換77, 49

檢查49

交換50, 49

中間省略



製作樹-3

額外測資

Input

max

7 9 13 6 8 47 51

Output

6 9 8 51 7 47 13

Input

min

88 41 59 2 17 63 85 98 13 52 77 23 24 56

Output

98 41 88 13 52 17 77 2 63 24 59 23 85 56

製作樹-4 (20%)

- 問題描述

- 給定一串數列 $\text{array}[n]$, $1 \leq n \leq 100$ 。數列中的值皆大於等於0, 小於等於100。
- $\text{array}[n]$ 中每一個點皆為leaf, 而每一個點的值為leaf的權重。
- 建構出一個binary tree未必是complete binary tree, tree中的每個node的值皆來自他的兩個兒子的值的和。
- 請計算出tree的leaf的權重。

- Input

- 一串數列(int), 以空格間隔

- Output

- leaf的最小權重(int)

製作樹-4 example

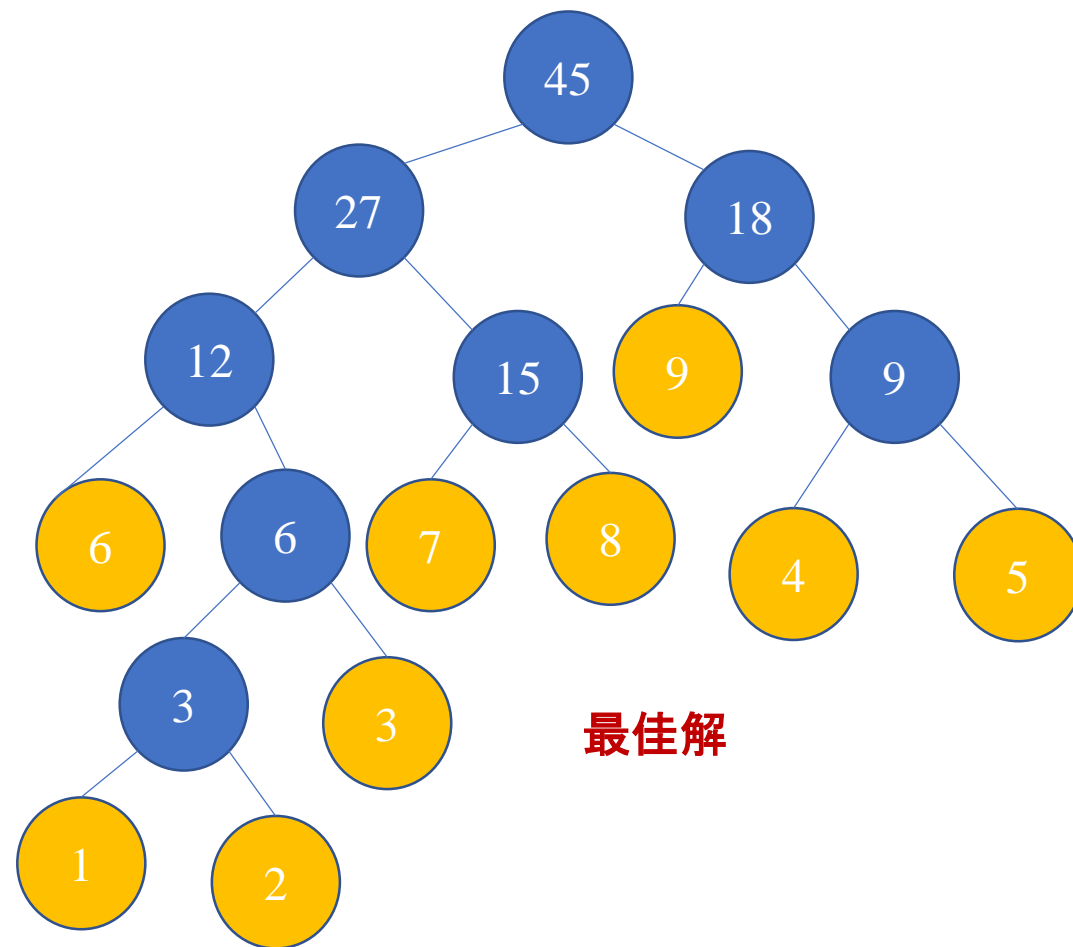
- Input

4 5 2 6 3 1 9 7 8

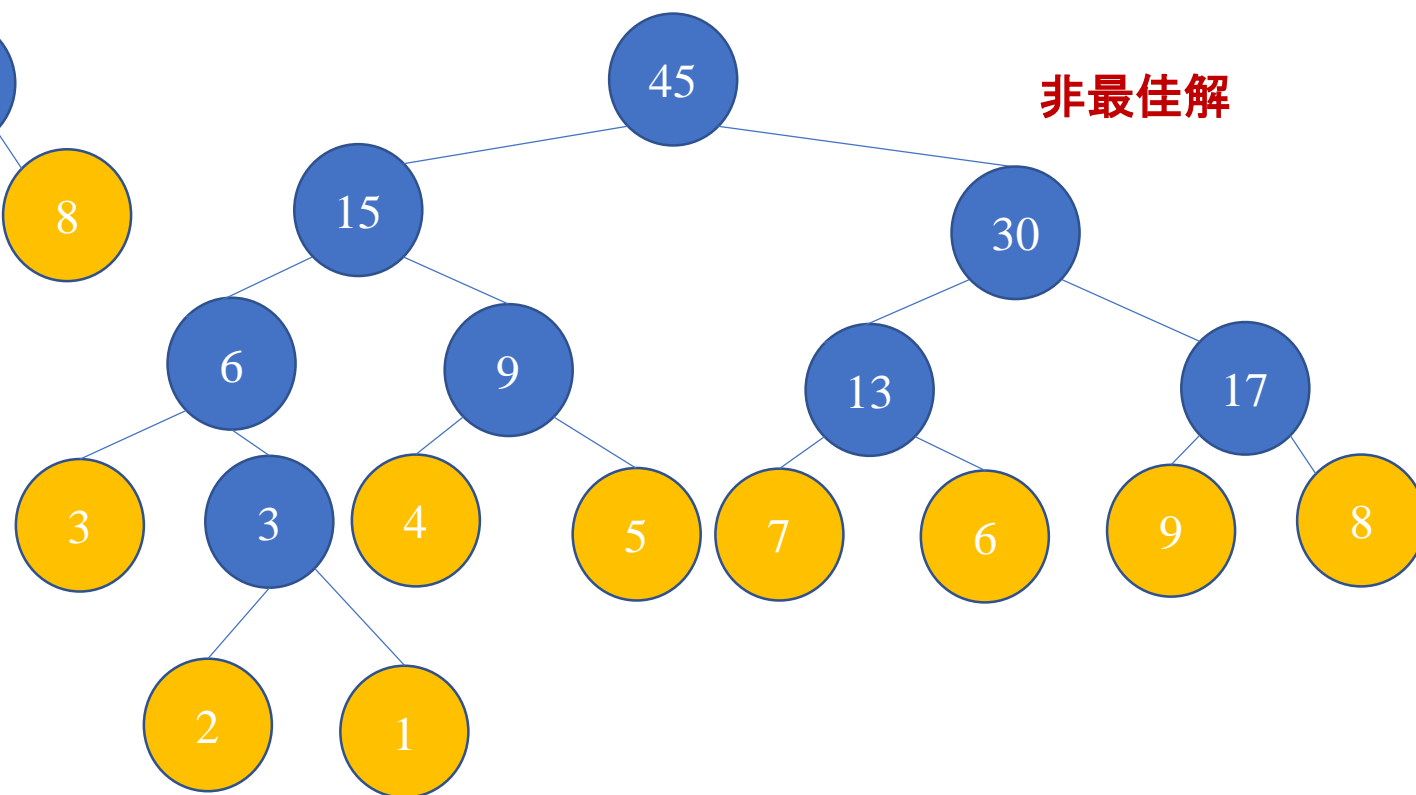
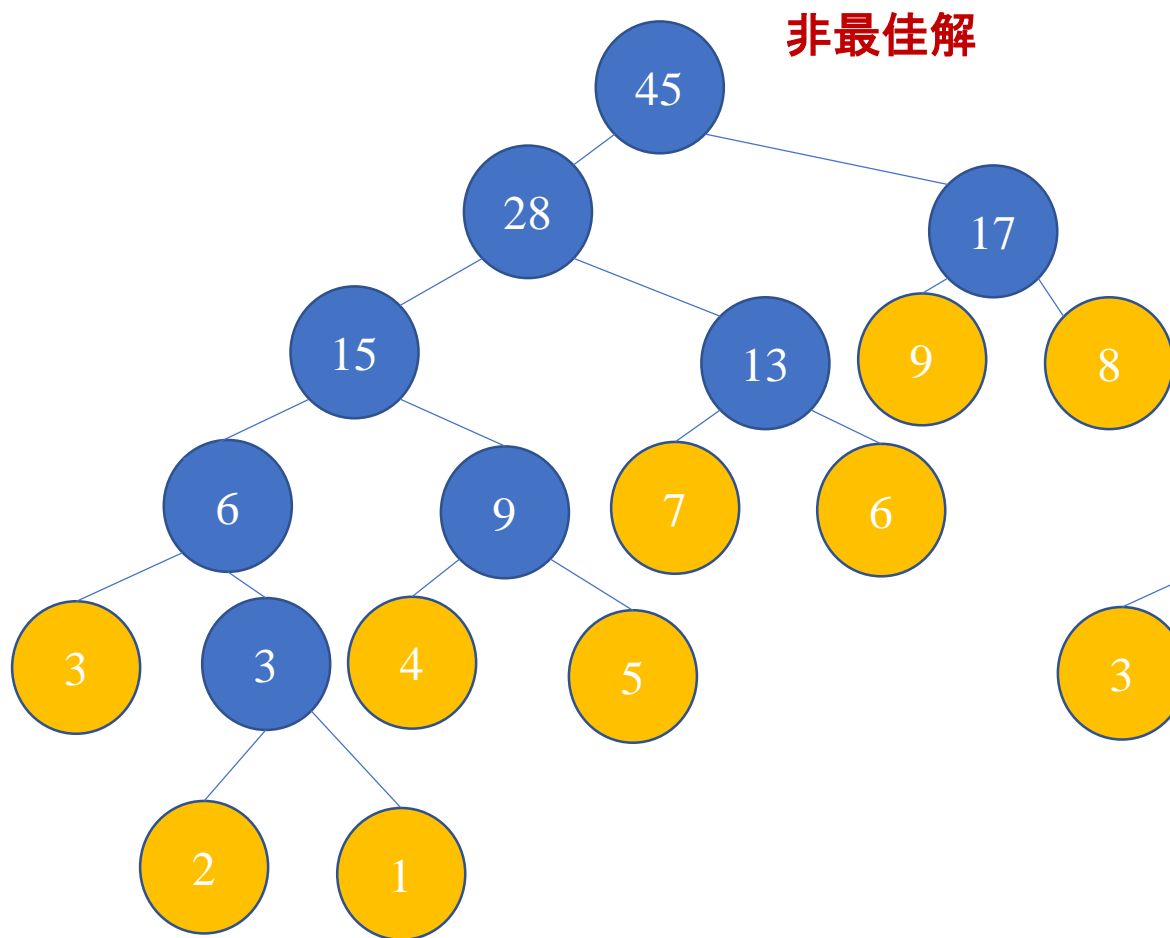
- Output

180

$//(1+2)*6+3*5+(6+7+8+4+5)*4+9*3$



製作樹-4



製作樹-4

額外測資

Input

6 10 11 19 5 2 7 23

Output

310

Input

9 7 13 2 6 1 8

Output

165

製作樹-5 (20%)

//解法不是DP喔!很像OBST但不是,因為有改過.這題可以使用暴力法解(就是把可能的樹找出來找最佳,我的測資不會很大)更好的解法我還沒想到.

- 問題描述

- 給定兩串數列array1[n], array2[m], $2 \leq m \leq 100$, $n = m - 1$ 。兩串數列中的值皆大於等於0, 小於等於100。
- array1表示parent node, array2表示leaf node。
- 建構出一個binary tree, 找到整棵binary tree的最小權重。

- Input

- 第一列表示parent node, 中間以空格間隔(int)
- 第二列表示leaf node, 中間以空格間隔(int)

- Output

- 整棵樹的最小權重(int)

製作樹-5

example

//之後我會再給兩個測資

- Input

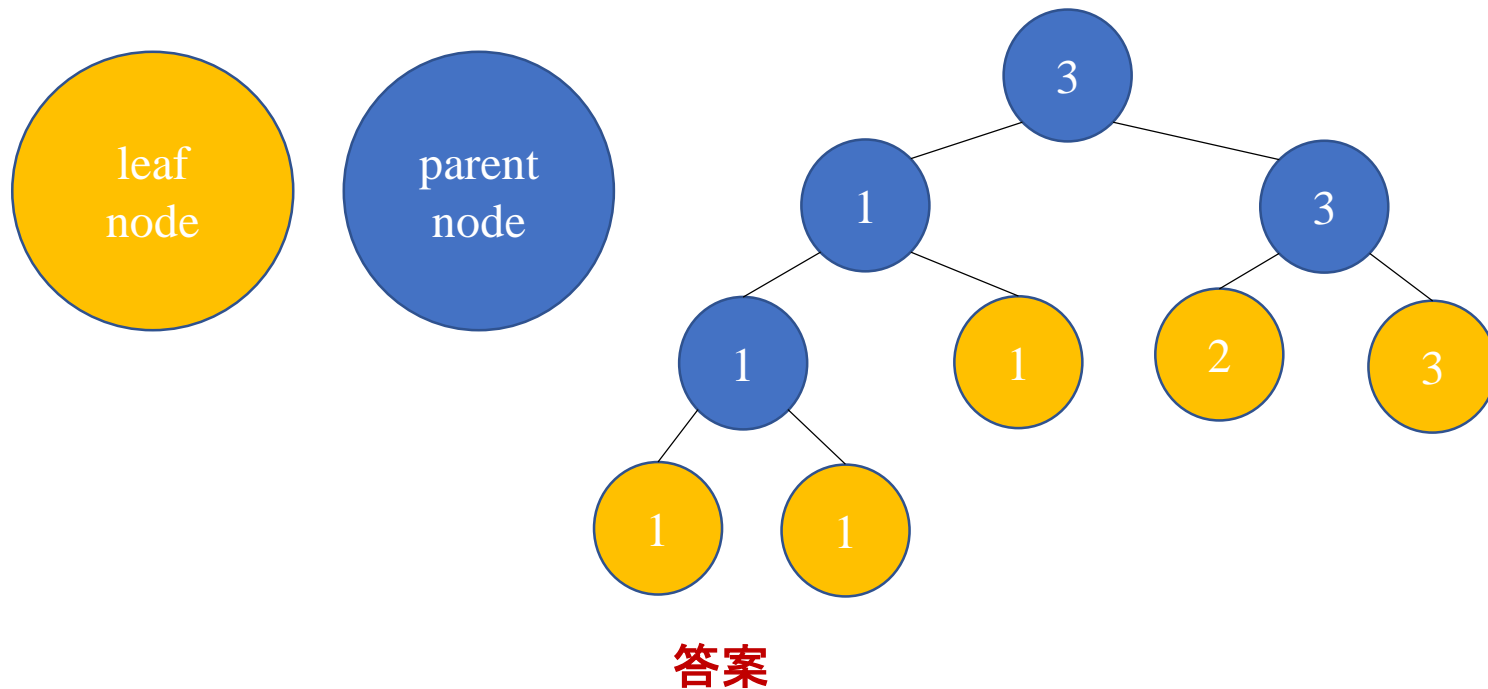
3 3 1 1

2 3 1 1 1

- Output

40

// $3*1+(3+1)*2+(2+3+1+1)*3+(1+1)*4=40$



製作樹-5

額外測資

Input

3 1 2

2 4 3 1

Output

39

Input

9 7 13

6 8 99 13

Output

345

繳交方式

- Accepted C/C++
- Deadline : 2019/12/16(一) 中午12:00
- Input必須是read file!(input1.txt, input2.txt , ...)
- Output必須是read file!(output1.txt, output2.txt, ...)
- 檔名 : main.c若有很多檔案請標清楚題號比如, main1.c, main2.c, ...
- 繳交方式:將你寫的所有檔案(包含main檔和input檔, 但不需要output檔)放在一起壓縮, 以i-learning方式繳交, 檔名為 “學號+姓名+homework3”。
- 禁止抄襲 , 可以的話打上註解方便閱讀
- 我們會以Dev-C++來做批改