

電機系 資料結構緒論 作業三

姓名：郭庭維

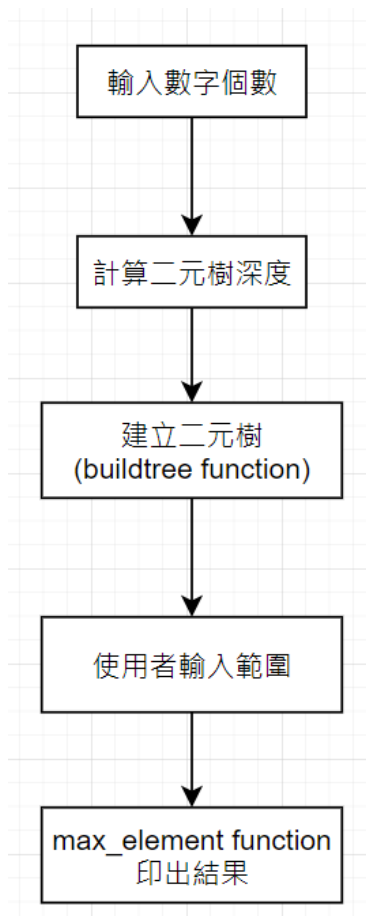
系級：會計 112

學號：H14086030

HW3 使用二元樹找最大值

一、總體流程圖與說明

整體流程：



Buildtree :

```
int buildtree(node* r,int depth,int currentd,int index,int* data,int size)
```

r 當前處理的node

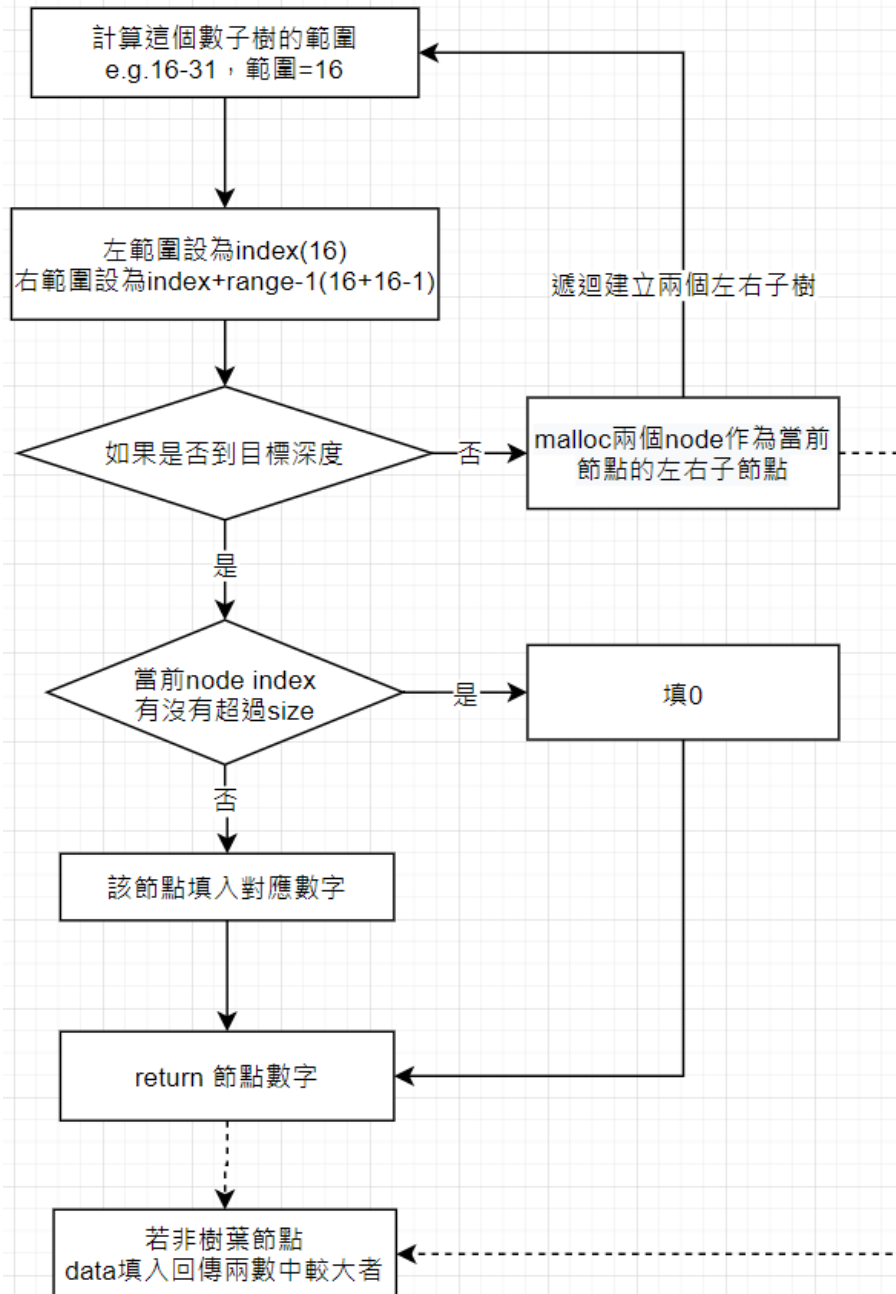
depth 該樹的目標深度

currentd 該節點的深度

index 起始範圍

data 該節點的數值

size 已開始隨機生成數的數量



max_element

```
int max_element(node* root,int lowrange,int uprange,int* choosednum,int max)
```

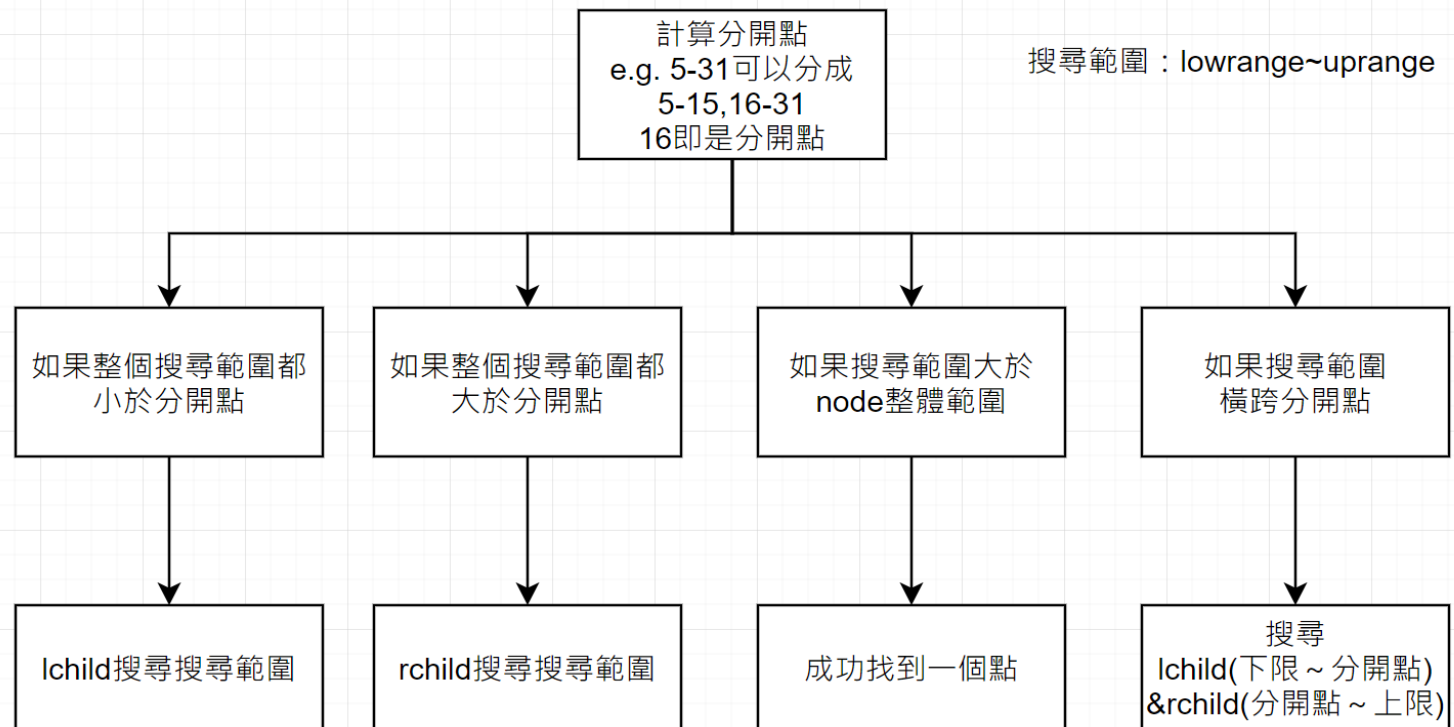
root 當前節點

lowrange 搜尋下限

uprange 搜尋上限

choosednum 被選中的點的數量

max 本次搜尋找到的最大值



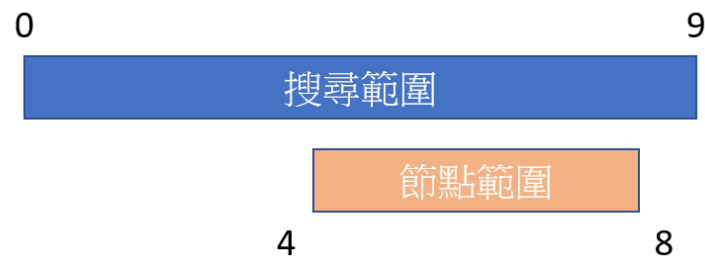
二、程式碼說明與程式設計

1. 二元樹的建立：為了克服給予每個 **node** 數值還有其子樹 **range** 的問題，我決定一開始先建立一個足夠大的 **empty tree** 然後再把值填進去。所以當數量為 2^n 次方時直接建立高度 n 的樹，否則就建立高度為 $\log_2(n) + 1$ ，並且填好每個 **node** 子數的範圍。接下來當該節點深度等於目標深度時，填入一開始產生的值，並回傳給父節點，父節點會根據大小填入對應的值。

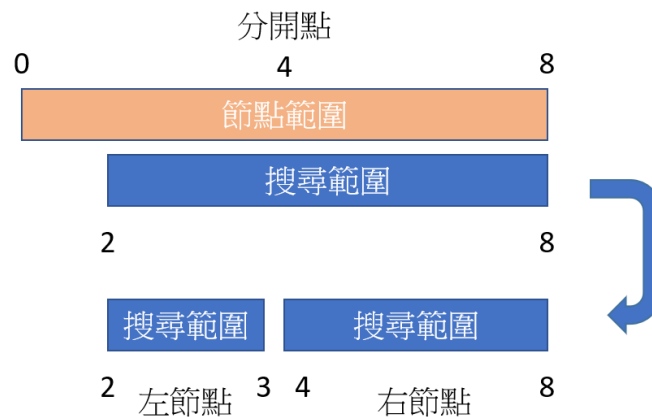
2. 二元樹找最大值

分開點 (**split_points**)：e.g. 5-31 可以分成 5-15/16-31，16 即是分開點

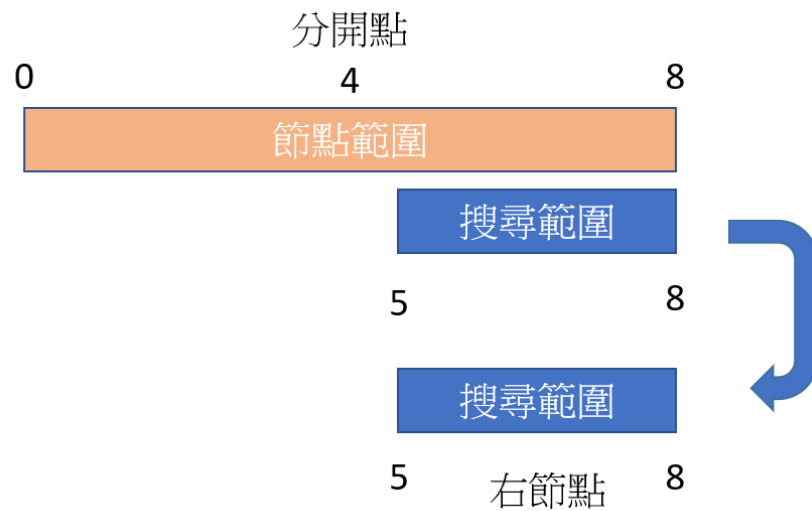
- i. 狀況一：如果搜尋範圍大於 **node** 整體範圍
成功找到一點，該節點為我們要的候選節點



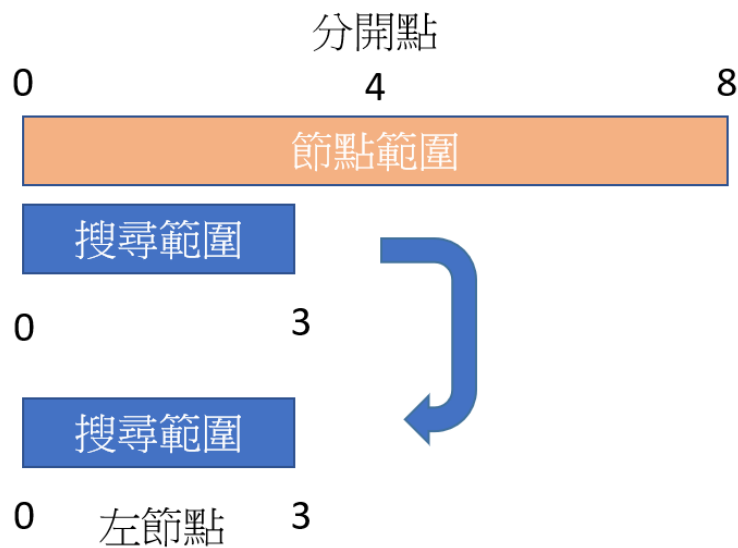
- ii. 如果搜尋範圍橫跨分開點
搜尋：**lchild**(下限～分開點)、**&rchild**(分開點～上限)



- iii. 如果整個搜尋範圍都大於分開點
rchild 搜尋搜尋範圍



- iv. 如果整個搜尋範圍都小於分開點
lchild 搜尋搜尋範圍



如此遞迴就可以找到所有的候選節點了

三、心得與討論：這次的作業整體來說沒有上次的那麼複雜，只要想好了要怎麼實作很快就可以寫出來了，一開始想了很多辦法都沒有辦法完成「建立樹」或是即便可行，後續的搜尋也會卡住（從下往上建立，搜尋時就沒辦法從 root 連到 leaf.....），想了一陣子決定用現在這個方法雖然當總數不是 2^n 時會生成很多 0 的點，但是這應該算是比較好寫跟能處理各種狀況的方法了。