

## 八、高等樹

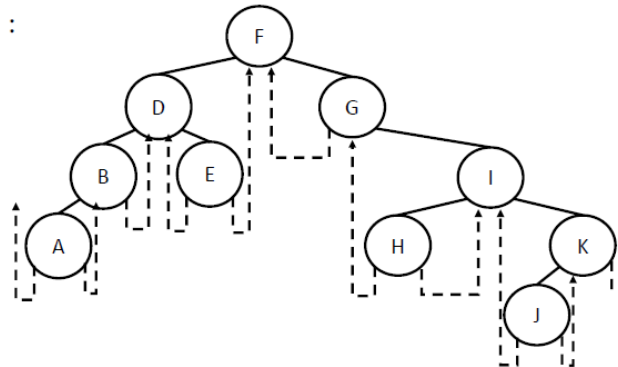
### 一、引線二元樹 (Threaded Binary Tree)

#### 1. 引線二元樹簡介

- ①利用沒用到的鏈結欄位，來指向樹中的其他節點
- ②這些鏈結欄位內存的值，就稱為引線(Thread)
- ③引線是用來指向中序走訪的前一個或後一個走訪節點

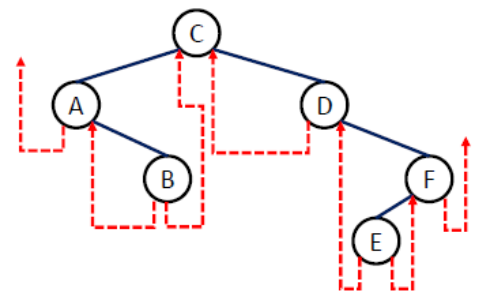
#### ※學生演練

#### ◎試利用中序走訪下圖：



2. 引線二元樹節點結構：為了能夠區分 Child 欄位內的值是指向小孩的鏈結還是引線，結果須多加了兩個欄位來標示是否為引線。(此欄位只需 1 位元，代表是或否，0 表否，1 表是)

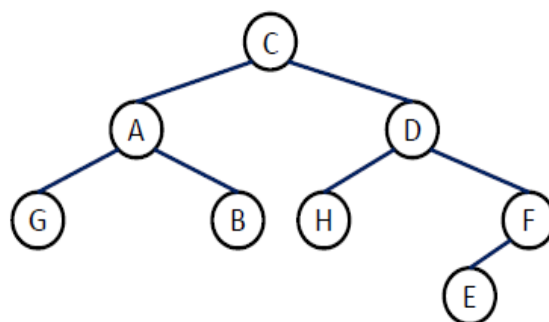
Left Thread	Left_Child	Data	Right_Child	Right Thread
-------------	------------	------	-------------	--------------



#### ※範例演練

#### ◎請 (1) 利用中序走訪下圖

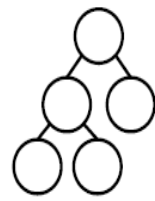
(2) 新增引線 (使用有向虛線表示) 到下列二元樹，使其成為引線二元樹：



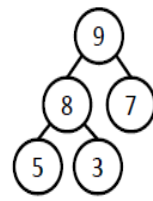
## 二、堆積樹 (Heap Tree)

### 1. 堆積樹簡介

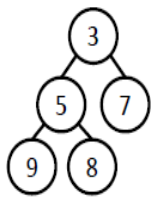
- ① 是一種完整二元樹
- ② 若父節點大於子節點，稱最大堆積樹(Max Heap Tree)；反之，若父節點小於子節點，稱為最小堆積樹(Min Heap Tree)



完整二元樹



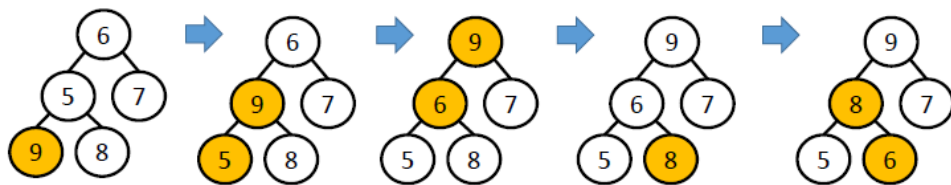
最大堆積樹



最小堆積樹

### 2. 堆積數運算

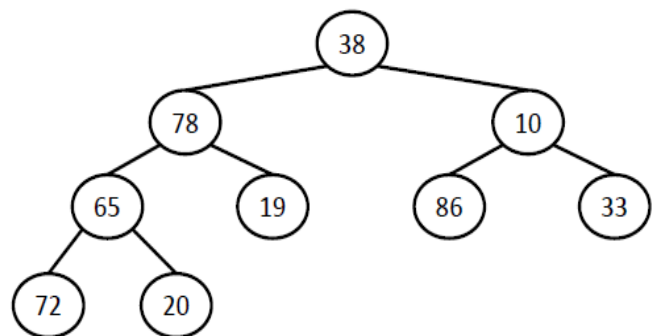
- ① 找出完整二元樹內鍵值最大者，與其根節點交換，依序移到樹根
- ② 找出次大者，依序與其樹根節點交換，直到不能交換為止
- ③ 重複步驟 3，直到全部節點的鍵值大於它的左子樹與右子樹的鍵值為止



#### (一) 完整二元樹轉換成最大堆積樹

##### ※範例講解

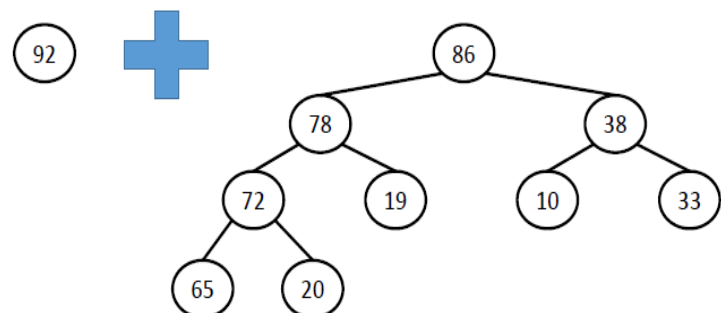
##### ⊙請將右圖二元樹轉成最大堆積樹



#### (二) 將具有新鍵值的節點加入最大堆積樹

##### ※範例講解

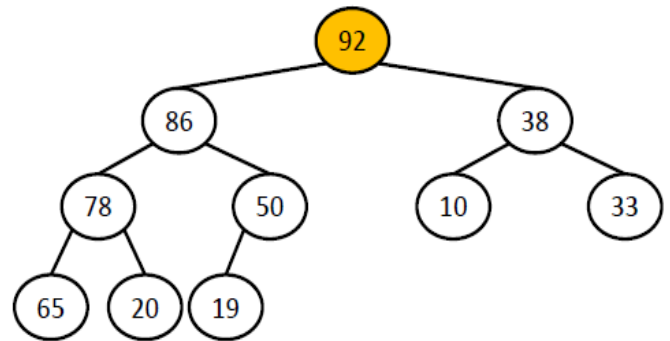
##### ⊙請將右圖新增鍵值後重新排列成最大堆積樹



### (三)刪除根節點

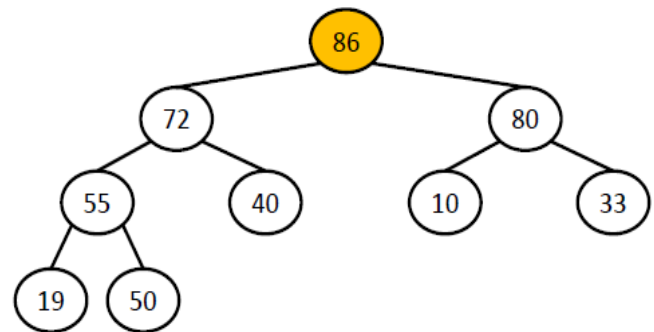
#### ※範例講解

⊙請將下圖刪除根節點後重新排列成最大堆積樹



#### ※學生練習

⊙請將下圖刪除根節點後重新排列成最大堆積樹



## 三、最小-最大堆積樹 (Min-max Heap Tree)

### 1. 最小-最大堆積樹定義

①是一個完整二元樹

②二元樹是交替的階層方式呈現，分別為最小階層(min level)和最大階層(max level)，其中樹根為最小鍵值

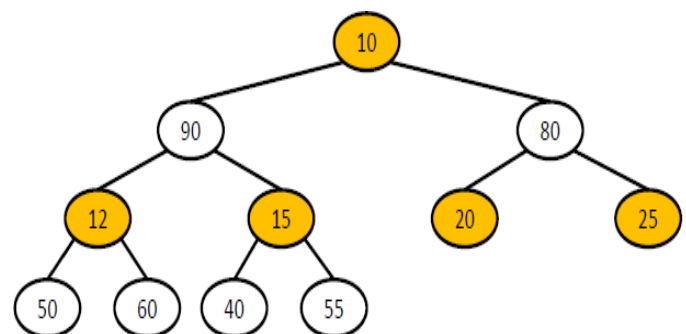
### 2. 最小-最大堆積樹運算

(一)判斷是否為最小-最大堆積樹

(二)加入節點

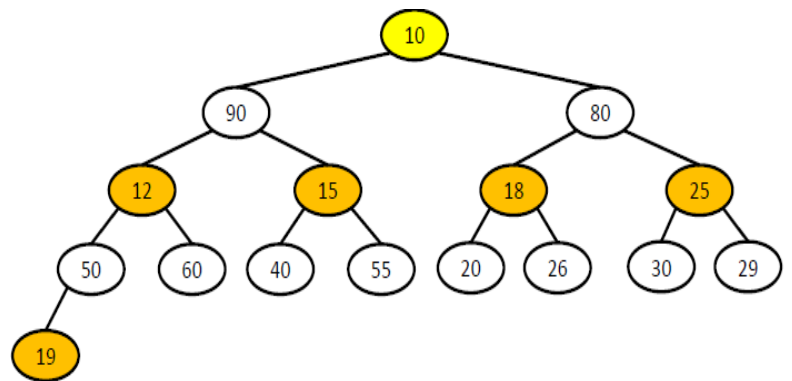
#### ※範例講解

⊙請將右圖加入具有鍵值 2 的新節點



※學生練習

◎請將右圖加入具有鍵值 41 的新節點

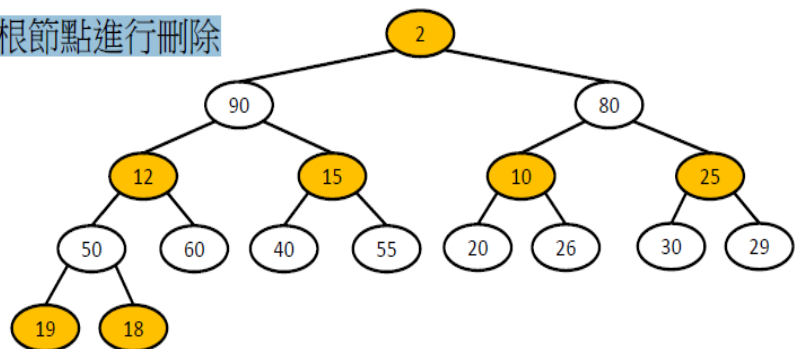


(三)刪除節點

※範例講解

◎從根節點進行刪除

從根節點進行刪除



## 四、雙向堆積樹 (Deaps)

1. 雙向堆積樹定義：是一棵完整二元樹，它可以為空，若不為空，則必須滿足下列特性：

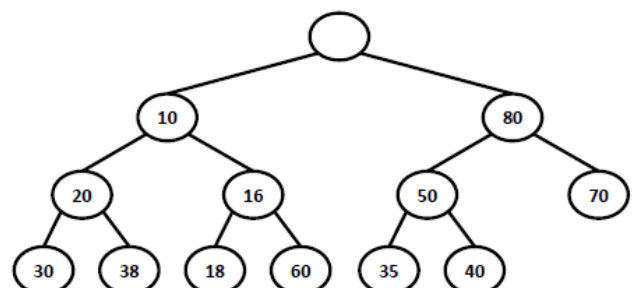
- ①樹根不包含任何的元素
- ②左子樹為最小堆積樹(Min Heap)；右子樹為最大堆積樹(Max Heap)
- ③左子樹節點鍵值相對位子必須要小於右子樹節點鍵值

2. 雙向堆積樹運算

(一)新增節點

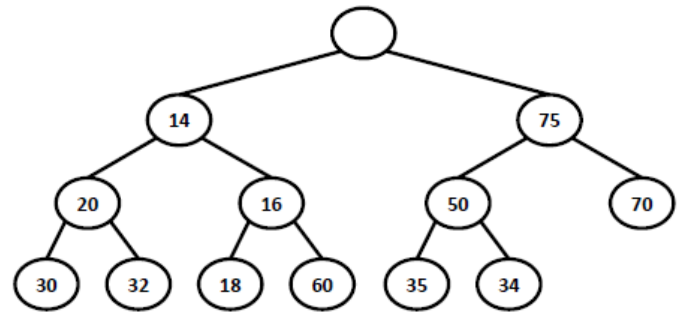
※範例講解

◎請將下圖新增鍵值 5 的節點



※學生演練

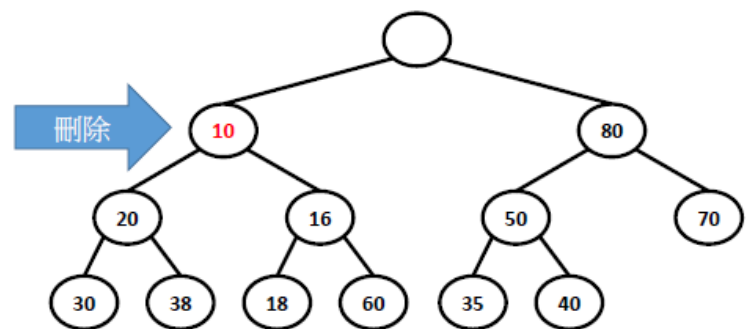
⊙請新增一個鍵值節點為 15 的節點到右邊這棵 DEAP 內



(二)刪除最小鍵值節點

※範例講解

⊙請將下圖刪除鍵值 10 的節點



※學生演練

⊙請刪除一個下圖這棵 DEAP 內的一個最大鍵值節點

