

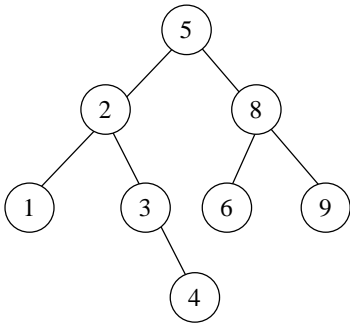
# 《資料處理》

一、請將下列 8 個數字 2, 1, 5, 9, 8, 3, 4, 6 依序加入一棵空的平衡二元樹 (AVL Tree)，請畫出此二元樹並寫出其「中序」之追蹤順序？(25 分)

## 試題評析

本題結合了資料結構中 tree 兩種常見題型，也就是 AVL Tree 的插入建樹和中序遍歷，對於考生來說 AVL tree 可能會稍難一點，因為必須考慮到插入時的旋轉平衡問題，跟一般的 Binary Search Tree 有點不太一樣，而後半的中序必須依照前面的建樹結果才能答出來，所以前面錯了，後面也會錯，做題時必須謹慎檢查。

答：



中序順序為：1 2 3 4 5 6 8 9

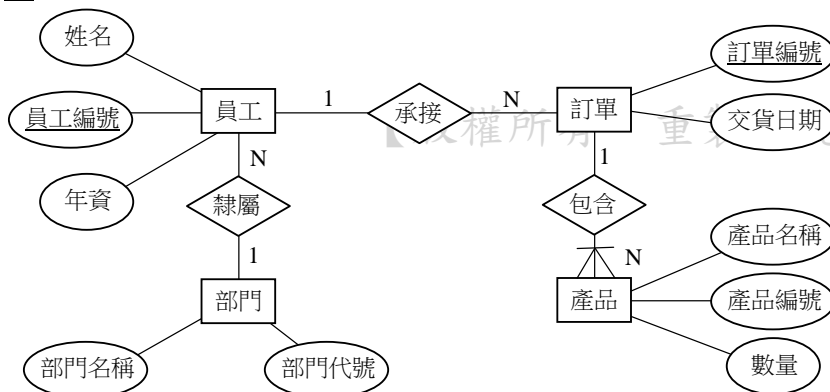
二、請依據下列資料需求，畫出實體-關係模型 (E-R model)。(25 分)

- (一) 員工 (Employee)：包括姓名 (Employee\_name)、員工編號 (Employee\_no)、年資 (Employee\_seniority)，其中員工編號為唯一。
- (二) 訂單 (Order)：訂單編號 (Order\_no)、交貨日期 (Order\_date)，每一筆訂單包含多筆產品，其中訂單編號為唯一，且訂單必定由一個員工負責承接。
- (三) 部門 (Department)：部門名稱 (Department\_name)、部門代號 (Department\_no)，每個部門有多個員工且每個員工必要隸屬於一個部門。
- (四) 產品 (Product)：產品名稱 (Product\_name)、產品編號 (Product\_no)、數量 (Product\_quantity)，一個訂單中必須包含至少一項產品。

## 試題評析

畫 E-R model 算是常考題型之一，照著題目要求小心畫，把相對關係畫清楚，一般都不會有太大問題。

答：



三、試利用氣泡排序法將 27, 10, 9, 68, 56, 40, 23 進行排序，請寫出流程，並說明最壞情況下之時間複雜度為何？（25 分）

**試題評析**

氣泡排序在排序演算法中，雖然不是最有效率的方法，但卻是所有人入門必學的一個經典算法，因為他的概念非常簡單，所以對於考生來說應該是要拿到分數的一題。

**答：**

(一) 泡沫排序每一回合會把最大的數字，經過兩兩交換，放到最後面（泡沫浮起來的概念）。

第一回合

27, 10, 9, 68, 56, 40, 23  
 10, 27, 9, 68, 56, 40, 23  
 10, 9, 27, 68, 56, 40, 23  
 10, 9, 27, 56, 68, 40, 23  
 10, 9, 27, 56, 40, 68, 23  
 10, 9, 27, 56, 40, 23, 68  
 68的位置確定

第二回合

10, 9, 27, 56, 40, 23, 68  
 9, 10, 27, 56, 40, 23, 68  
 9, 10, 27, 40, 56, 23, 68  
 9, 10, 27, 40, 23, 56, 68  
 56的位置也確定

第三回合

9, 10, 27, 23, 40, 56, 68  
 40的位置也確定

第四回合

27和23交換

第五回合

經檢查沒有任何交換(每對數字左邊都小於右邊)  
 代表已經排序完成  
 算法結束

(二) 最壞情況的複雜度  $O(n^2)$

出現在數字反轉時

四、CPU 排程演算法（CPU Scheduling）的目的為何？常見的排程方式有先到先執行排程（First Come First Served, FCFS）、最短工作優先排程（Shortest Job First, SJF）、優先權排程（Priority）、循環分時排程（Round Robin, RR），請分別解釋此四種排程方式。（25 分）

**試題評析**

這題算是作業系統的基本觀念，排班在作業系統中是很重要的問題，因為他能決定系統排程的效率，幾種經典的CPU排班算法，學生考前應該要掌握其中的基本原則以及差異。

**答：**

(一) CPU排程的目的，最主要就是找到一種排班方式，使得工作能夠在期望的時間、效率下完成，讓作業系統能有效完成工作。

(二) 1.FCFS：

依照進入Ready Queue的先後順序，來決定使用CPU的順序。  
 優點：實作非常簡單。

缺點：效能一般較差，平均等待時間很差，且可能造成convoy effect，不適合Time-sharing system。

#### 2.SJF：

會在CPU空閒時，選擇CPU Burst Time最小的process。

其中又依照能不能被插隊，分為Preemptive和Non-preemptive。

優點：效益高。

缺點：各個process的下一個Burst Time較難預測，很難用在Process scheduler。

#### 3.Priority：

會按照Priority高低分配CPU給process，同時間有多個相同Priority的process，則採FIFO。

缺點是容易有Starvation的現象、低優先權可能永遠等不到(一般會用Aging解決這個問題)。

#### 4.RR：

這種排程會訂一個Time slice(Time quantum)，當process使用超過這個time slice時，CPU會中斷此process，並切換使用權。

常用在Time sharing system上，算法效率和Time slice大小有很大關係。

Time slice趨近無窮大時，基本上就相當於FIFO。

【版權所有，重製必究！】