

# 《資料處理概要》

## 試題評析

今年的考題，以電腦網路、資料結構、資料庫與網際網路技術為主。去年沒出現的網路，今年出現不少。其中，資料庫考題的比重仍然最重，資料結構次之。且資料結構其中有一小題，出的題型較偏，若非瞭解整套的資料結構，應該會不知如何作答。

第一題：為電腦網路，若有網路協定的Connection-Oriented與Connection-Less概念，也可朝這方面回答。

第二題：電腦網路與網際網路技術。除了MIME可能較陌生外，其他應該算容易作答。

第三題：為資料結構，第一小題，考的比較偏，其餘應容易回答。

第四題：為資料結構，屬基本考題。

第五題：屬資料庫基本考題。

第六題：屬資料庫基本考題。

一、在廣域網路上使用的兩種資料交換技術中，以線路的使用效率來說，兩者的差異性為何？以資料傳輸的延遲來說，兩者有何差別？網路建置成本上，兩者又有何差別？（12分）

**答：**

(一)目前網路傳輸，是以封包(Packet)形式在進行資料交換，在廣域網路上所使用的兩種交換技術分別為：Virtual Circuit Packet Switching（在答題中簡稱VCPS，線路交換）與Datagram Packet Switching（在答題中簡稱DPS，分封交換）。

1.Virtual Circuit Packet Switching：概念同網路協定上的 Connection-Oriented Transmission。此技術源自於傳統電話網路的Circuit Switching技術。顧名思義，此技術會在進行資料交換時，資料發送者會啟動 Call Setup機制，將資料交換時所需要經過的節點，每一站都將資源事先保留下來，成為Virtual Circuit通道，做資料交換，資料接收端收到Call Setup訊息後，會回傳Acknowledgement，表示Virtual Circuit通道建立完成。當Virtual Circuit通道建立完成後，所有的資料都是透過此通道進行傳輸，不受其他流量的干擾。因此所有資料是依序傳遞，且進行資料傳輸錯誤控制，以確定資料正確的被傳送到接收端。

2.Datagram Packet Switching：概念同網路協定上的Connection-Less Transmission。此技術在傳輸時，不會啟動Call Setup機制，所有的資料封包送上網路後，每個封包各自進行資料傳輸與交換，由於沒有完備的資料傳輸控制，因此資料並不會依序到達接收端，故資料可能會出錯，或接收端沒收到。

(二)以線路使用效率來說，DPS優於VCPS。在VCPS下，傳輸通道可能被保留建立起來，但是不常使用，且此傳輸通道也無法提供給其他資料傳輸使用，造成浪費。DPS技術不需要建立資源保留的通道，網路資源可以被有效利用，因此效率較高。

(三)以資料傳輸的延遲來說，VCPS優於DPS。因為VCPS會建立Virtual Circuit通道，因此延遲可以被有效掌握。但DPS，每個封包會各自進行資料交換，路徑選擇亦不同，因此延遲無法有效掌控。

(四)在網路建置的成本上，VCPS高於DPS。原因是VCPS所有的節點都需要提供資源保留功能，且需要進行完整傳輸錯誤控制。但DPS，則只需要提供簡單的傳輸控制即可。

## 【參考書目】

張宏亮老師，資料處理課程講義第一回，P.82與上課筆記。

二、請逐一回答下述問題：傳統的網路應用中，MIME指的是什麼（請說明代表的中文意義或英文全稱）？其作用為何？MIME的規範中，定義七種資料的內容形態（content type）中，除了text、image、video、audio及application外，另兩種內容形態為何？IP網路上的通訊協定中，TCP與UDP是屬於那一層的協定？兩種協定的主要區別為何？目前Web使用的HTTP通訊協定是經由那一種傳輸層協定提供服務？（18分）

**答：**

(一)MIME是Multipurpose Internet Mail Extensions的縮寫。MIME type是一種網際網路上為電子郵件編碼的通訊

協定型態，傳送如聲音、影像等非以文字為主的資料所使用的標準。使用者可利用MIME把包含聲音、影像、圖形、文字的多媒體電子郵件經由傳輸控制協定送出。附帶有字形、格式與呈現資料的文件，也可以容易的在不同系統上展現。只要傳送、接收郵件的雙方都有支援識別 MIME type 的軟體，郵件系統就會將使用者各種型式的檔案還原，在閱讀信件內容時呼叫相對應的應用程式。如果接收兩端沒有支援MIME type，那麼接收MIME type信件時便會出現亂碼。

(二)MIME規範中，除了text、image、video、audio、與application外，尚有：

1.Multipart：此型態代表，用於連接message的多個部分成另為一個message，這些多個部分的資料可以是不同型態。

2.Message：此型態用於包裝一個e-mail型態的message。

(三)IP網路通訊協定中，TCP與UDP協定是屬於Layer 4傳輸層(Transport Layer) 的協定。

1.TCP協定為一Connection-Oriented傳輸協定。TCP協定在資料傳輸之前會先進行Call Setup動作，一般通稱為Three-Way Handshaking，建立Virtual Channel，此外並進行資料流量控制、錯誤控制與重傳機制。因此資料傳輸會依序到達，且端點與端點間的資料不會出錯。雖然傳輸品質可以被保證但效率較UDP協定不佳。

2.UDP協定為一Connection-Less傳輸協定，不會進行Call Setup動作，只會進行簡單錯誤控制。因此資料傳輸時不一定依序到達，且資料會有出錯，或沒收到的可能。

(四)目前Web使用的HTTP通訊協定是基於TCP/IP傳輸協定提供服務的。

#### 【參考書目】

張宏亮老師，資料處理課程講義第一回，P. 52、82與上課筆記。

三、請問如何將一個二維陣列儲存在一維序列式 (sequential) 的電腦記憶體？堆疊 (stack) 資料結構的主要特性為何？舉出一個在程式執行時需要使用到堆疊的應用。在記憶體中，如何實作堆疊？(15分)

答：

(一)在電腦中，有兩種方法用來實作二維陣列在線性一維的記憶體中。分別為：Row-major order與Column-major order兩種。假設欲將二維陣列A[5][5]實作於記憶體中，按照上述兩種方法，其排列順序分別為：

1.Row-major order：先以每一個Row走完，再換。所以存放順序為 A[0,0] A[0,1] A[0,2] A[0,3] A[0,4] 接著從 A[1,0] A[1,1] A[1,2] ...以此類推。如下圖所示：

陣列編號	記憶體位址
A[0,0]	0
A[0,1]	1
A[0,2]	2
A[0,3]	3
A[0,4]	4
A[1,0]	5
A[1,1]	6
A[1,2]	7
A[1,3]	8

2.Column-major order：先以Column走完，再換。所以存放順序為 A[0,0] A[1,0] A[2,0] A[3,0] A[4,0] 接著從 A[0,1] A[1,1] A[2,1] ...以此類推。如下圖所示：

陣列編號	記憶體位址
A[0,0]	0
A[1,0]	1
A[2,0]	2
A[3,0]	3
A[4,0]	4
A[0,1]	5
A[1,1]	6
A[2,1]	7
A[3,1]	8

(二)Stack資料結構特性為先進後出(First In Last Out, FILO)，意即先放入堆疊中的資料會較晚移出堆疊。

(三)通常程式在撰寫時若使用遞迴的方式，就會使用到堆疊資料結構，將上一層的資料先Push到堆疊中儲存下來。待下層都執行完畢後，再將資料Pop回來繼續未完成的程式。或在各式圖形追蹤演算法中，進行深度優先追蹤(Depth-First Search, DFS)時，會需要使用堆疊資料結構，將未拜訪過的節點先用堆疊儲存下來，待接下來的追蹤。

(四)實作堆疊，通常可使用陣列(Array)來實作，或使用鍊結(Linked List)來實作。

- 1.使用陣列實做：通常寫程式時，會宣告一陣列，與一Top變數，用來記錄目前資料存放於堆疊的陣列空間編號。假設從陣列編號0的位置開始存放，每Push一筆資料，此Top變數就加一。此方法，優點是實作容易，但缺點是堆疊大小固定，沒有彈性。
- 2.使用鍊結實作：使用動態記憶體配置，將一個個節點，根據資料Push到堆疊中，串連起來形成堆疊。並有一指標變數Top，記錄目前堆疊最上層存放資料的位址。每Push一筆資料，就分配一記憶體空間來儲存，並將此節點鍊結起來，且將此指標變數，指向此新加入節點。優點為堆疊的大小彈性，但實作較複雜。

#### 【參考書目】

1. 一般資料結構書籍。
2. 張宏亮老師，資料處理課程講義第四回，P. 103與上課筆記。

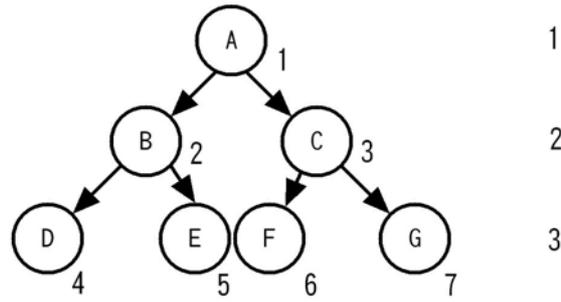
四、考慮一個二元樹的資料，請舉出兩種資料結構並說明如何分別以此資料結構在記憶體實作儲存資料的方法。此外，請分析這兩種儲存方式的優劣點。若要擴充到一般的樹狀結構，兩種方式各需要做什么改進？(15分)

**答：**

二元樹(Binary Trees)表示，樹的節點最多只有兩個子節點。實做二元樹，通常有兩種方法：分別為「用陣列實作」和「用鍊結實作」。

(一)1.以陣列來實作：此方法實作簡單，但樹的大小會受限於陣列的大小。此外，要使用此方式來實作，最好是完滿二元樹，也就是每一個樹節點都要具備兩個子節點。否則將造成陣列中大部分是空值的空間浪費。完滿二元樹是一棵樹高h擁有 $2^h-1$ 個節點的二元樹，這是二元樹在樹高h所能擁有的最大節點數，換句話說，只需配置 $2^h-1$ 個元素，我們就可以儲存樹高h的二元樹，如下圖所示，btree[ ]表示一陣列：

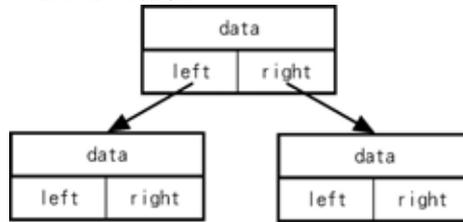
樹高



btree[ ]	A	B	C	D	E	F	G
	[1]	[2]	[3]	[4]	[5]	[6]	[7]

2. 從樹根開始，依序編號，即可使用陣列來實作二元樹。此外，在進行找左子樹與右子樹時，可很方便透過計算可得。例如：btree[i]的左子節點為btree[i\*2]，右子節點為btree[i\*2+1]。btree[i]的父節點為btree[i/2]。
3. 若擴充到一般樹狀結構，根據樹的分枝度，需要調整找尋子節點與父節點的計算公式。

(二) 1. 二元樹鏈結表示法是使用動態記憶體配置來建立二元樹，採用結構表示法，來表示樹的節點結構，只是成員變數，除了要儲存的資料外，並有兩個指向左子樹和右子樹的指標，如下圖所示，data儲存需要存放的資料，left與right分別為指向左右子數的指標：



2. 若需要知道父節點的位置，則需要加入一指向父節點的指標變數。使用此方法實作二元樹，實作較複雜，但樹的大小彈性，且當樹的節點稀疏時也較適合使用此方法。
3. 若擴充到一般樹狀結構，需要彈性的預留指向子樹的指標變數空間。

**【參考書目】**

張宏亮老師，資料處理課程講義第四回，P.116-117與上課筆記。

- 五、(一) 物件式資料庫系統比關連式資料庫優異的功能有那些？請具體提出兩項優點。
- (二) 關連式資料庫的表格裡，一個或多個屬性 (attribute) 成為候選鍵 (candidate key) 的條件是什麼？
- (三) 主鍵 (primary key) 和候選鍵有什麼關係？
- (四) 有關關連式資料庫的設計中，何謂沒有正規化的資料表？(五) 又何謂滿足第一正規化的表格？(20分)

**答：**

- (一) 物件導向式資料庫是將物件導向的概念，導入於資料庫中。物件導向資料庫的優點：
1. 資料庫與程式設計環境使用一致的資料模式：物件導向的技術可以用在資料庫及系統設計及開發上，因此不會造成資料庫的設計以及程式的設計因使用了不同的資料模式而產生錯誤。
  2. 資料模型化能力：由於物件導向技術具有資料模型化的能力，可以處理複雜及龐大的資料結構和物件關係，因此當系統或資料擴充時，並不會造成太大的影響，而且可以很彈性的加以修正及加強系統功能，這一點是關連式資料庫無法做到的。
  3. 長時間交易：關連式資料庫對於長時間的資料處理並無法提供良好的解決方法，但是物件導向資料庫則

可以滿足此種需求，並不會因為長期的鎖住（lock）某筆資料而造成執行上或是使用上的困擾。

4.執行效率較佳：物件導向技術並不利於如關連式資料庫一樣的索引方式作資料處理。它支援了物件識別碼技術，因此可以非常快速的找出所要的物件。

所以，物件導向式資料庫對於資料的處理方式確實具有較大的彈性。另外，其對於需要處理各種大量且複雜的資料，像是多媒體的影像、圖形、聲音等，或是需要表示資料間複雜關係的應用系統，都適合使用物件導向資料庫。

(二)在關連式資料庫的表格裡，要成為候選鍵(Candidate Key)需要滿足兩項條件：

1.唯一性：表示此組屬性，可以決定非在此組屬性內的其他屬性。

2.最小性：若任何一個屬性從此組屬性移出，將不再具備唯一性。

(三)一個關連表中可能不只一個候選鍵，在這些候選鍵中挑一個出來當主鍵，其他未被選到的則為其他鍵(Alternate Key)。

(四)沒有正規化的關連表，通常會有意義相同的Attribute出現，或資料表中的值並非單元值(Atomic Value)。

(五)滿足一階正規化，關連表中不會出現意義相同的Attribute，且資料表中的值均為單元值(Atomic Value)，並且需要有一主鍵(Primary Key)。

#### 【參考書目】

張宏亮老師，資料處理課程講義第二回73頁、第三回7頁與上課筆記。

六、(一)一個資料庫系統的設計大都依循層次架構，請說明怎麼樣的層次架構以及為何要有如此設計？

(二)關連式代數(relational algebra)定義多種處理資料的運算，除了Selection與Projection之外，請列出至少另外二種運算。

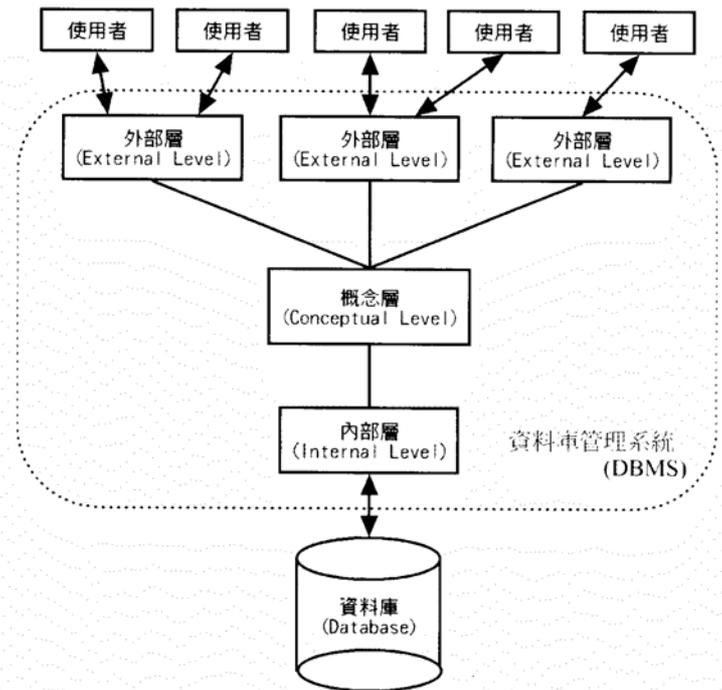
(三)假如以一個關連儲存書籍的資料如下：Book(isbn, title, author, year, company)，其中Book為關連的名稱。請分別以關連式代數以及SQL的敘述回答下述兩個查詢問題：

1.列出所有書名(title)為“database systems”的所有資料

2.列出所有書名(title)與作者(author)之資料。(20分)

**答：**

(一)美國國家標準協會在1980年代就以三階層資料庫的概念來建構資料庫，稱為ANSI/SPARC三階層架構。此架構分別以使用者、資料庫管理師與實際儲存的觀點來檢視資料庫儲存的資料。



1.外部層(external level)：是最接近使用者的層次，描述個別使用者所見資料庫的部分集合或資訊內涵。

2.概念層(conceptual level)：描述整體資料庫的邏輯結構，是概念上完整的資料庫。

3.內部層(internal level)：是最接近資料實體的層次，描述資料庫在輔助儲存體存放方式，是實際儲存資料的結構。

目的是分隔使用者看到的資料庫與實際資料庫儲存的結構，在概念層的完整資料庫，並不會因為實際儲存裝置的改變而影響。利用三階層架構(外層／概念層／內層)製作資料庫，在各層之間的應對乃是協助達成資料獨立(Data Independence)的主要基礎。

(二)以基本與非基本運算區分，可分為：

1.基本(primitive)operator：指不能由其他operator導出，包含Union、Different、Product、Restrict Select、Project共五個。

2.非基本(non-primitive)operation：指可由五個primitive operator來取代者，包含Intersection、Join、Division。

(三)Relation Schema為Book(isbn, title, author, year, company)

(1)以Relational Algebra來查詢：

$$\sigma_{title=="database systems"}(Book)$$

(2)以SQL 來查詢：

```
Select title, author
From Book
```

**【參考書目】**

張宏亮老師，資料處理課程講義第二回，P. 55-56、77、88與上課筆記。