

逢 甲 大 學

資 訊 工 程 學 系 專 題 報 告

資 料 結 構 教 學 - 排 序 法

王 璽 銘 (四 甲)

學 生 : 楊 文 嘉 (四 甲)

黃 柏 睿 (四 甲)

指 導 教 授 : 黃 溪 春

中 華 民 國 九 十 二 年 十 一 月

目 錄

| | |
|--------------------------|----|
| 目錄..... | 1 |
| 圖表之目錄..... | 3 |
| 摘要..... | 6 |
| 第一章 導論..... | 8 |
| 1.1 前言..... | 8 |
| 1.2 系統簡介..... | 8 |
| 1.3 系統目的與動機..... | 9 |
| 第二章 環境介紹與工作規劃..... | 11 |
| 2.1 系統開發工具..... | 11 |
| 2.1.1 PHP..... | 11 |
| 2.1.2 Flash..... | 12 |
| 2.1.3 Apache Server..... | 13 |
| 2.1.4 Dreamweaver..... | 14 |
| 2.1.5 FrontPage..... | 14 |
| 2.1.6 MySQL..... | 15 |
| 2.1.7 phpMyAdmin..... | 16 |
| 2.2 系統執行環境..... | 16 |
| 2.2.1 伺服器端..... | 16 |
| 2.2.2 使用者端..... | 18 |
| 2.3 工作分配與進度..... | 18 |
| 2.3.1 工作分配..... | 18 |
| 2.3.2 甘特圖..... | 19 |
| 第三章 系統分析..... | 21 |
| 3.1 系統架構..... | 21 |
| 3.2 使用者觀點..... | 22 |
| 3.3 系統功能..... | 24 |
| 3.3.1 線上教學..... | 24 |
| 3.3.2 線上測驗..... | 25 |
| 3.3.3 成績查詢..... | 26 |
| 3.3.4 學生資料修改..... | 26 |
| 3.3.5 系統管理..... | 27 |
| 3.3.6 討論版..... | 29 |

| | | |
|--------|---------------------|----|
| 3.4 | 系統流程..... | 30 |
| 3.4.1 | 線上測驗流程..... | 30 |
| 3.4.2 | 成績查詢流程..... | 31 |
| 3.4.3 | 學生基本資料修改流程..... | 31 |
| 3.4.4 | 系統管理流程..... | 32 |
| 3.4.5 | 帳號管理流程..... | 33 |
| 3.4.6 | 題庫管理流程..... | 34 |
| 3.4.7 | 考卷管理流程..... | 35 |
| 3.4.8 | 考試管理流程..... | 36 |
| 3.4.9 | 討論版管理流程..... | 37 |
| 3.4.10 | 新增帳號流程..... | 37 |
| 3.4.11 | 查詢密碼流程..... | 38 |
| 3.4.12 | 討論版流程..... | 38 |
| 第四章 | 系統實作..... | 40 |
| 4.1 | 教學單元介紹..... | 40 |
| 4.2 | 教學頁面製作..... | 50 |
| 4.3 | PHP 網頁製作..... | 56 |
| 4.3.1 | PHP 跟 MYSQL 連結..... | 56 |
| 4.3.2 | 登入介面撰寫..... | 58 |
| 4.3.3 | 考試介面撰寫..... | 59 |
| 4.3.4 | 計算成績撰寫..... | 60 |
| 4.3.5 | 題庫出題撰寫..... | 62 |
| 4.3.6 | 考卷出題撰寫..... | 64 |
| 4.3.7 | 考卷開啟、關閉程式撰寫..... | 68 |
| 4.4 | 資料庫規劃、設計..... | 70 |
| 4.4.1 | 資料庫 E-R 圖..... | 70 |
| 4.4.2 | 資料表總覽..... | 70 |
| 4.4.3 | 資料表詳細說明..... | 71 |
| 4.4.4 | 資料庫建置..... | 78 |
| 第五章 | 系統評估與心得..... | 80 |
| 5.1 | 系統評估..... | 80 |
| 5.2 | 未來發展..... | 81 |
| 5.3 | 專題製作心得..... | 81 |
| 參考資料 | | 83 |
| 附錄 A | 檔案清單..... | 84 |
| 附錄 B | 題目清單..... | 88 |

圖表之目錄

| | | |
|--------|-------------------------|----|
| 圖 2-1 | 系統預計甘特圖..... | 19 |
| 圖 2-2 | 系統實際甘特圖..... | 20 |
| 圖 3-1 | 系統架構圖..... | 21 |
| 圖 3-2 | 使用者觀點圖..... | 22 |
| 圖 3-3 | 老師觀點圖..... | 22 |
| 圖 3-4 | 學生觀點圖..... | 23 |
| 圖 3-5 | 訪客觀點圖..... | 23 |
| 圖 3-6 | 系統管理者觀點圖..... | 24 |
| 圖 3-7 | 線上教學架構圖..... | 25 |
| 圖 3-8 | 線上測驗架構圖..... | 25 |
| 圖 3-9 | 成績查詢架構圖..... | 26 |
| 圖 3-10 | 學生資料修改架構圖..... | 26 |
| 圖 3-11 | 系統管理架構圖..... | 27 |
| 圖 3-12 | 帳號管理架構圖..... | 27 |
| 圖 3-13 | 題庫管理架構圖..... | 28 |
| 圖 3-14 | 考試管理架構圖..... | 28 |
| 圖 3-15 | 考卷管理架構圖..... | 29 |
| 圖 3-16 | 討論版架構圖..... | 30 |
| 圖 3-17 | 線上測驗流程圖..... | 31 |
| 圖 3-18 | 成績查詢流程圖..... | 32 |
| 圖 3-19 | 學生基本資料修改流程圖..... | 32 |
| 圖 3-20 | 系統管理流程圖..... | 33 |
| 圖 3-21 | 帳號管理流程圖..... | 34 |
| 圖 3-22 | 題庫管理流程圖..... | 35 |
| 圖 3-23 | 考卷管理流程圖..... | 36 |
| 圖 3-24 | 考試管理流程圖..... | 37 |
| 圖 3-25 | 討論版管理流程圖..... | 38 |
| 圖 3-26 | 新增帳號流程圖..... | 38 |
| 圖 3-27 | 查詢密碼流程圖..... | 39 |
| 圖 3-28 | 討論版流程圖..... | 39 |
| 圖 4-1 | BUBBLE-SORT 程式碼..... | 40 |
| 圖 4-2 | HEAP-SORT 程式碼..... | 42 |
| 圖 4-3 | INSERTION-SORT 程式碼..... | 43 |
| 圖 4-4 | MERGE-SORT 程式碼..... | 44 |
| 圖 4-5 | QUICK-SORT 程式碼..... | 46 |

| | | |
|-----------|-------------------------|----|
| 圖 4-6 | RADIX-SORT 程式碼..... | 47 |
| 圖 4-7 | SELECTION-SORT 程式碼..... | 48 |
| 圖 4-8 | SHELL-SORT 程式碼..... | 50 |
| 圖 4-9 | FLASH 開發環境圖..... | 50 |
| 圖 4-10(a) | 按鈕實作圖..... | 51 |
| 圖 4-10(b) | 按鈕實作圖..... | 51 |
| 圖 4-10(c) | 按鈕實作圖..... | 52 |
| 圖 4-10(d) | 按鈕實作圖..... | 52 |
| 圖 4-11 | FLASH 實作圖..... | 53 |
| 圖 4-12 | 引導線製作圖..... | 54 |
| 圖 4-13(a) | Quick-Sort 製作..... | 54 |
| 圖 4-13(b) | Quick-Sort 製作..... | 55 |
| 圖 4-13(c) | Quick-Sort 製作..... | 55 |
| 圖 4-14 | PHP 與 MYSQL 連結圖..... | 56 |
| 圖 4-15 | 登入介面程式碼圖..... | 59 |
| 圖 4-16 | 考試介面程式碼圖..... | 60 |
| 圖 4-17 | 計算成績程式碼圖..... | 62 |
| 圖 4-18 | 題庫出題程式碼圖..... | 63 |
| 圖 4-19 | 出題介面圖..... | 64 |
| 圖 4-20 | 考卷出題程式碼圖..... | 66 |
| 圖 4-21 | 考卷出題介面圖..... | 66 |
| 圖 4-22 | 亂數出題程式碼圖..... | 67 |
| 圖 4-23 | 亂數出題介面..... | 68 |
| 圖 4-24 | 更新學生排名程式碼圖..... | 69 |
| 圖 4-25 | 更新難易度程式碼圖..... | 69 |
| 圖 4-26 | 考卷開啟、關閉介面圖..... | 69 |
| 圖 4-27 | 資料庫 E-R 圖..... | 70 |
| 圖 4-28 | 資料表總覽圖..... | 71 |
| 圖 4-29 | 學生資料表圖..... | 72 |
| 圖 4-30 | 題庫資料表圖..... | 73 |
| 圖 4-31 | 考卷資料表圖..... | 74 |
| 圖 4-32 | 題目章節資料表圖..... | 75 |
| 圖 4-33 | 管理者資料表圖..... | 76 |
| 圖 4-34 | 學生答案資料表圖..... | 76 |
| 圖 4-35 | 討論版文章、回覆文章資料表圖..... | 78 |
| 圖 4-36 | phpMyAdmin 介面圖..... | 79 |

| | | |
|-------|----------------|----|
| 表 2-1 | 伺服器使用軟體..... | 17 |
| 表 2-2 | 工作分配表..... | 18 |
| 表 4-1 | 學生資料表欄位..... | 72 |
| 表 4-2 | 題庫資料表欄位..... | 73 |
| 表 4-3 | 考卷資料表欄位..... | 74 |
| 表 4-4 | 題目章節資料表欄位..... | 75 |
| 表 4-5 | 管理者資料表欄位..... | 76 |
| 表 4-6 | 學生答案資料表欄位..... | 77 |
| 表 4-7 | 討論版資料表欄位..... | 77 |
| 表 4-8 | 回覆文章資料表欄位..... | 78 |



摘 要

本專題是以 Windows XP 加上 Apache 架設線上教學網站，使用常見的Web 介面讓使用者使用。線上教學內容共包含了九個部份。分別是：教學頁面、線上測驗、成績查詢、學生基本資料修改、新增帳號、查詢密碼、教學頁面、討論版、系統管理。選擇Windows XP 來架設網站是因為它的介面、環境對我們較為熟悉；其次是它的安全性也夠，一般人要入侵也不容易，讓管理者也更容易管理。在網頁部份，一部份為Flash教學頁面，另一部份由 PHP 語法及 MySQL 這一個資料庫來架構，透過 PHP 和 MySQL 的搭配而完成了這一個網站。

網站在架設時，發現透過程式的使用可以使網站和網頁更加容易管理；例如PHP，可以將網頁的部分內容和資料庫相結合，與資料庫的結合增加了網頁的彈性，也使得網站可透過資料庫來作管理；由於本網站著大部分都是採用 PHP 的語法，也結合了資料庫，更是將此優點表現出來。

線上測驗主要內容是要讓學生，輸入個人的班級、學號、密碼，進入考場考試。在這裡有一個重點，學生要開始考試之前，需要由老師先行開放考試時間。查閱成績主要是讓學生可以知道他在每一個章節、每一次的考試成績，學生只要在網頁上選擇他所想知道的考試成績，就可以知道該次考試的成績。

在教學方面我們主要採用Fundamentals of Data Structures in C 這一本書籍，並且採取比較著名的八種排序，分別為BUBBLE-SORT，INSERTION-SORT，QUICK-SORT，SELECTION-SORT，HEAP-SORT，MERGE-SORT，RADIX-SORT，SHELL-SORT，主要針對這幾個排序做介紹。而教學頁面主要功能是讓學生能在課堂上之外的時間，可以透過網路進入到教學網站中學習各排序的排序技巧，而我們就是以動畫配合程式碼的方式讓學生更加瞭解各排序程式的技巧。讓學生不至於單單面對枯燥乏味的文字敘述而已。

當初在規劃系統時，我們的目標便在於讓老師可以輕易地使用考試出題系統與製作考卷。利用簡單的操作介面系統便可於短時間內製作一份考卷且可對考卷作局部修改，相信這項功

能對於要製作考卷的使用者而言絕對是一大福音。這也是我們由衷希望這份系統能帶給使用者更多的便利性。往後我們希望可以做出更完整、更方便的考試系統，讓大家在使用上更容易上手！



第一章 導論

1.1 前言

近年來台灣，網際網路的蓬勃發展，幾乎達到家家有網路，人人可上網的境界，網路咖啡廳各處林立。無線網路也逐漸大眾化，相信在不久的未來，只要隨身攜帶筆記型電腦，PDA，甚至行動電話，到處都可以”行動上網”，人類的溝通再也不受限制。

基於此理念，傳統的教學方式也能有重大改變。目前已有遠距視訊教學，知識的傳達已無地域上的限制，雖然教學品質，師生互動的成效有待觀察，但我相信，隨著虛擬實境技術的成熟，這些問題也能輕易克服。

但另一方面，人類發明了文字也不是偶然的！影像，聲音是最直接的表達方式，卻會因個人表達的方式，風格而有所不同。文字雖然呆板，卻最單純，反而能呈現最真實的表達，這也是文字永遠不會消失的原因。課堂上還是離不開黑板，而網路教學及測驗，還是需要文字網頁，討論版及文字測驗來輔助。

1.2 系統簡介

這個系統屬於教學系統，該系統主要以「資料結構」的教學為主軸，製作目標在於使學生可以輕而易舉地進入資料結構的世界裡並愉快的學習。該系統包括FLASH教學、線上測驗、出題管理、成績查詢、討論版。利用大眾所熟悉的Web 介面，提供學習者一個便利的學習環境，並提供一套最生動活潑、最易懂的教材，讓您即使待在家中，也能夠輕鬆的學習本系統的使用對象，主要是以想在資訊領域中對於程式設計想更深入了解的使用者。

因為在資訊領域中，學程式設計的人，程式設計的底子十分重要，底子要打穩，就得對程式設計中常用到的資料結構有

所了解，因此，資料結構在資訊領域中便扮演著舉足輕重的角色。可謂「人手一套，不可或缺」。生動、易懂是我們製作此教學系統的兩大特色，而「簡單，更簡單，再簡單」是我們的製作方針。

網站中的教學內容完全針對初學者而設計，我們堅持把操作過程有如V8 全程記錄般完整重現在教學上，並且不厭其煩修訂；因為我們堅信「唯有我們用心，學習者才能獲得最大的利益」。我們希望提供不受時空限制的學習方式，不再侷限於固定的時間固定的地點才能學習，提供便利化的學習環境及課後複習的最佳管道。

很高興我們踏出了第一步，能讓我們做的教學系統提供給學弟妹使用；未來我們希望可以發展出更完善、更便利的教學系統，也希望可以製作出其他學科的教學內容來提供給學弟妹使用，讓大家在學習的道路上走的更輕鬆！

1.3 系統目的與動機

線上教學系統主要目的是，透過網際網路讓學習的時間以及空件變得更自由，不必再受到一般制式教學方法的限制，減低學習的困難度。

設定的帳號密碼登入，在系統中除了文字的說明之外，更加入了Flash的動畫教學，讓學習的過程更輕鬆，更可以透過評量系統的題目來評估自己的學習成效。

一個網站的運作，不在於漂亮的網頁美工，而是功能及內容為主。我們的理念是，學生隨時隨地，透過網路就能學習知識，再也不用背著厚厚的書。而我們也藉此機會學習了網站的建構，維護，以及資料庫的規劃，動畫的製作。

而且我們知道資料經過這一些排序的方法後可以將資料以有序的方式呈現，以利於之後我們要搜尋資料時可以用各種不同的搜尋技巧去搜尋以增進資料存取的效率。例如：二元搜尋法只能使用在已排序過的資料，所以在使用二元搜尋法之前要先將資料經過排序。而我們的測驗題目，選擇了資訊系相關

的教材 - “資料結構的排序法”，而主要的排序方法，參考 Horowitz 所著 Fundamentals of Data Structure in C 為主。

經由組員的合作，以及專題老師對系統的疑問，使我們瞭解開發一個專案需要周全的考慮，很多的問題都在實際處理時才會發現，組員們的合作溝通也很重要。雖然此系統並非完美，但從發現問題，尋求方法，到解決問題，自我對於系統的要求等等的過程才是重點。



第二章 環境介紹與工作規劃

2.1 系統開發工具

此章說明了我們開發此系統所使用的程式語言、開發軟體、所使用的軟體等等。

2.1.1 PHP

PHP 是一種伺服端的嵌入式 Scripting 語言，它運作在 HTML 文件裡提供 (ON DEMAND) 的內容產生能力，因此可以把網站變成 WEB 應用程式，再也不是單純的靜態網頁倉庫。而我們這組為何要使用 PHP 呢？其他的選擇如：ASP, Cold Fusion, Perl, Java 也都非常強大，也擁有很多使用者。我們看中的是，PHP 是開放式原始碼，而且單純，語法結構跟大家所熟悉的 C 語言非常類似，卻又去除了容易出錯的指標用法，將指標視為可變變數的用法之一。

PHP 有跟 Apache 伺服器緊密結合的特性；加上它不斷的更新及加入新的功能；並且它幾乎支援所有主流與非主流資料庫；再以它能高速的執行效率，因為不斷地有新的函式庫加入，以及不停地更新的活力，使得 PHP 無論在 UNIX 或是 Win32 的平台上都可以有更多新的功能。它提供豐富的函式，使得在程式設計方面有著更好的支援。

目前最新測試版本：PHP 4.3.4RC1 、PHP 5.0.0b1

穩定版本：PHP 4.4.3

PHP的官方網站 <http://tw2.php.net/>

Zend引擎 <http://www.zend.com/>

2.1.2 Flash

Flash 的產生，為網頁設計提供了動態網頁的選擇，擁有結合音效、影像、動畫的多媒體方案，也具備了高品質、體積小、互動性高的優點，我們所看中的就是其簡單，卻又擁有強大功能。其特色大略分述如下列幾點：

(1) 向量式繪圖軟體：

向量式的繪圖方式最大的優點就是檔案小，也不失其高畫質的特色。Flash 的繪圖工具操作簡易，每種工具皆有多種附屬功能工選擇。

(2) 影音整合：

可匯入各種聲音檔，為電影加上背景音樂及特殊音效，因應網路傳輸的需要，也可壓縮音效，再次精簡了檔案體積。

(3) 可匯入、匯出多種檔案格式：

音效檔 (WAV) (MP)、點陣、向量格式圖形檔 (WMF、AI、EPS、DXF、BMP、JPG、PNG、GIF)、數位影片格式 (MPEG、MOV、AVI)。

(4) 多種互動性的指令 (ACTION 動作指令)：

不用學習複雜的程式語言，直接使用 Actions 中的各種互動性指令來編輯、設定動畫。

(5) 動畫檔案體積小：

Flash 利用 Tweening 的原理，模擬出物件移動變化的動畫，不需像 GIF 需放置多張圖，檔案體積再次縮小。

(6) 可模擬網頁動畫下載的情況：

提供電影的網路測試環境，使用者可模擬網路上電影的下載播放情形，可得知電影檔案是否過大。

目前最新版本：Macromedia Flash MX 2004

MACROMEDIA公司網址：<http://www.macromedia.com/>

2.1.3 Apache Server

皆下來談到的就是該如何建立一個 Web 網站呢？我們需要一套能把網路上的網頁供其它人來存取的軟體，也就是所謂的 HTTP (Web) Server 協定服務軟體，HTTP Server 比較有名的包括：Apache HTTP Server、MS 的 Internet Information Server、Netscape 的 Enterprise Server 等。

而 Apache HTTP Server 是一個免費且開放源碼的 HTTP 伺服器，主要目的在於提供一個安全並且有效率的 HTTP Server 環境。Apache HTTP Server 目前有 Unix-Based 和 Windows 的版本。

根據統計，目前 Apache HTTP Server 已成為佔有率最高的 HTTP Server 軟體，而且它的功能與執行效率並不會比一般商業的 Web Server 軟體來的差，甚至領先許多。

Apache HTTP Server 目前的版本分為 1.3.x 版與 2.0.x 版，兩個版本基本上功能都差不多，不一樣的地方在於 2.0.x 的執行效率會比 1.3.x 來得好，因為 2.0.x 程式結合了執行緒〈thread〉與程序〈process〉，讓熟悉系統的管理者能夠調校各種不同的組合，達到最佳效能，而 1.3.x 版只能以程序〈process〉方式來運作，兩者觀念完全不同，但是目前看來 1.3.x 版程式的運作會比 2.0.x 來的穩定。另外，2.0.x 版也把許多模組內建在自己的系統裡。

最新版本：Apache 2.0.47

Apache 軟體網站：<http://httpd.apache.org/>

2.1.4 Dreamweaver

網頁編輯軟體，同樣是 MACROMEDIA 公司的產品。它結合了視覺化的配置工具、程式開發功能及程式設計支援，令使用者不需瞭解繁雜的 HTML 語言，就可以快速地製作出符合標準的網頁程式。

且提供了整合式環境所需要的工具，可以開發 HTML、XHTML、XML、ASP、微軟 ASP.NET、JSP、PHP 及 Macromedia ColdFusion 網站。另外透過 Macromedia Exchange for Dreamweaver，可以自選超過 800 種擴充元件，自訂及擴充整個開發環境。

最新版本：Macromedia Dreamweaver MX 2004

MACROMEDIA公司網址：<http://www.macromedia.com/>

2.1.5 FrontPage

FrontPage2000 是一個製作網頁的軟體工具，它結合了網頁編輯跟網站管理的功能，使用者可以在它的介面裡直接編輯製作網頁，其中還加入了Excel 的功能，還有VB Script 和Java Script 支援，可以直接加入，直接用預覽介面來看成果，是一個可以省去寫繁雜的html語言的軟體工具，習慣用html來寫網頁的人也可以在介面編輯。

最新版本：FrontPage2002

軟體網址：

<http://www.microsoft.com/taiwan/office/frontpage/>

2.1.6 MySQL

為了達成網站的動態呈現，就想到我們需要資料庫。而眾多資料庫中，我們選擇 MySQL 的原因，MySQL 是一個真正的多使用者、多執行緒、符合 ANSI SQL92 標準 SQL 資料庫伺服器。

跨平台、開放原始碼、可自行編譯，當然要是不打算更改標準原始碼，也有編譯完成的二進位檔。SQL (結構化查詢語言) 是世界上最流行的和標準化的資料庫語言。

MySQL 是以一個客戶機/伺服器結構的實現，它由一個伺服器背景執行程式 `mysqld` 和很多不同的客戶程式和庫組成。SQL 是一種標準化的語言，它使得儲存、更新和存取資訊更容易。

例如，你能用 SQL 語言為一個網站檢索產品資訊及儲存顧客資訊，同時 MySQL 也足夠快和靈活以允許你儲存記錄文件和圖像。MySQL 主要目標是快速、健壯和易用。

最初是因為我們需要這樣一個 SQL 伺服器，它能處理與任何可不昂貴硬件平台上提供資料庫的廠家在一個數量級上的大型資料庫，但速度更快，MySQL 就開發出來。MySQL 建立的基礎事業已用在高要求的生產環境多年的一套實用例程。

儘管 MySQL 仍在開發中，但它已經提供一個豐富和極其有用的功能集。

版本資訊：

Production: [4.0.15a](#)

Alpha: [4.1.0](#)

Recent: [3.23.58](#)

MySQL 網站: <http://www.mysql.com/>

2.1.7 phpMyAdmin

由於使用文字介面來操作 MySQL 有些麻煩，此軟體由 PHP 開發，來幫助我們以網頁的方式來操控 MySQL，所有想的到的資料庫指令都有，著實讓我們操作方便許多。

最新版本：phpMyAdmin 2.5.4

軟體首頁：<http://www.phpmyadmin.net/>

2.2 系統執行環境

我們所採用的軟體幾乎所有的 UNIX 和 Linux 系統上都能使用，就算是封閉性的微軟環境，經由開放源碼社群的努力，也能支援微軟的作業環境。不能在 UNIX LIKE 環境中開發的有 Dreamweaver、FrontPage 以及 Flash。

2.2.1 伺服器端

此章節說明了我們開發此系統所使用的機器配備以及作業環境，第一台機器是我們的網站伺服器、資料庫伺服器，系統就是建置在此台機器上，另外兩台機器則是其他組員開發系統時所使用的機器。以下將詳細列出我們使用的機器配備。

1. 桌上型電腦一(伺服器)

CPU ---PentiumIII 866MHz

硬碟---Maxtor 30GB

記憶體---320MB

顯示卡---NVIDIA GeForce2 MX400 多媒體繪圖/影像加速卡

顯示器---ADI LCD 15吋

作業系統---WinXP

使用軟體---

表2-1 伺服器使用軟體

| | |
|---|--------------------|
| • | Apache 1.3.27 |
| • | Apache Monitor |
| • | PHP 4.3.0 |
| • | MySQL 3.23.54 |
| • | PHP-Nuke 6.0 |
| • | phpMyAdmin 2.3.3pl |

2. 桌上型電腦二

CPU ---AMD Duron 1.6GHz

硬碟---Quantum 9GB

記憶體---256MB

顯示卡---NVIDIA GeForce3 多媒體繪圖/影像加速卡

顯示器---ViewSonic E70 17吋

作業系統---WinXP

3. 桌上型電腦三

CPU ---AMD Athlon 850MHz

硬碟---Seagate 30GB

記憶體---384MB

顯示卡---NVIDIA TNT2 多媒體繪圖/影像加速卡

顯示器---CTX 17吋

作業系統---WinXP

2.2.2 使用者端

一般個人電腦配備有網頁瀏覽器 (IE、Netscape、Mozilla) 皆可使用本系統，另外需下載 Flash Player 才可觀賞動畫，當然需要連接上網際網路，透過網際網路才能連接上我們的網站伺服器，進而使用本系統。

2.3 工作分配與進度

此章節說明了我們開發此系統時三個人分配工作的情形，還有我們開發系統的時程表。

2.3.1 工作分配

表 2-2 列出了我們本組工作分配的情形，工作內容分為資料蒐集、系統規劃、教學資料蒐集、教學以及動畫製作、資料庫規劃、系統測試、報告製作，有參與此項工作的人員在表格裡使用星號“★”來代表有參與此項工作。

表 2-2 工作分配表

| 工作內容 | | 資料蒐集 | 系統規劃 | 教學資料蒐集 | 教學、動畫製作 | 資料庫規劃 | 系統測試 | 報告製作 |
|------|-----|------|------|--------|---------|-------|------|------|
| 參與人員 | 楊文嘉 | ★ | ★ | ★ | ★ | | ★ | ★ |
| | 王璽銘 | ★ | ★ | ★ | | ★ | ★ | ★ |
| | 黃柏睿 | ★ | ★ | ★ | | ★ | ★ | ★ |

2.3.2 甘特圖

甘特圖是以一種表示系統開發過程的條狀圖表，可以顯示工作的活動，以及持續時間可用來排定、計畫和管理開發系統的行程。

圖 2-1 展示了一開始我們所預定開發此系統的甘特圖，第一部份是討論、規劃本系統的甘特圖，第二部分是我們實作系統的甘特圖，第三部分是系統修改的甘特圖，第四部分是我們作書面報告的甘特圖。



圖 2-1 系統預計甘特圖

圖 2-2 展示了我們開發此系統的實際甘特圖，第一部份討論、規劃，整體而言符合我們所預期的進度，並沒有太大的差異。第二部分實作，我們起步時間延遲了一些，意見也有分歧，因此拖慢了整體進度，程式整合算是較為困難的部分，缺乏溝通以致最後各人實作出來都有差異，花了很多時間在統一程式的部分。第三部分系統修改，此部分在實作的後部分實際上就

開始進行了，因此我們是一邊測試也一邊修改，所以與原本所計畫的時間不同。指導老師給了我們很多改進的意見，組員之間也互相提供很多意見。第四部分書面報告方面，訂定目錄很快就完成了，內容部分如預期花了最多的時間，進度比預期稍晚一些，因此也拖慢了後面的內容修改。

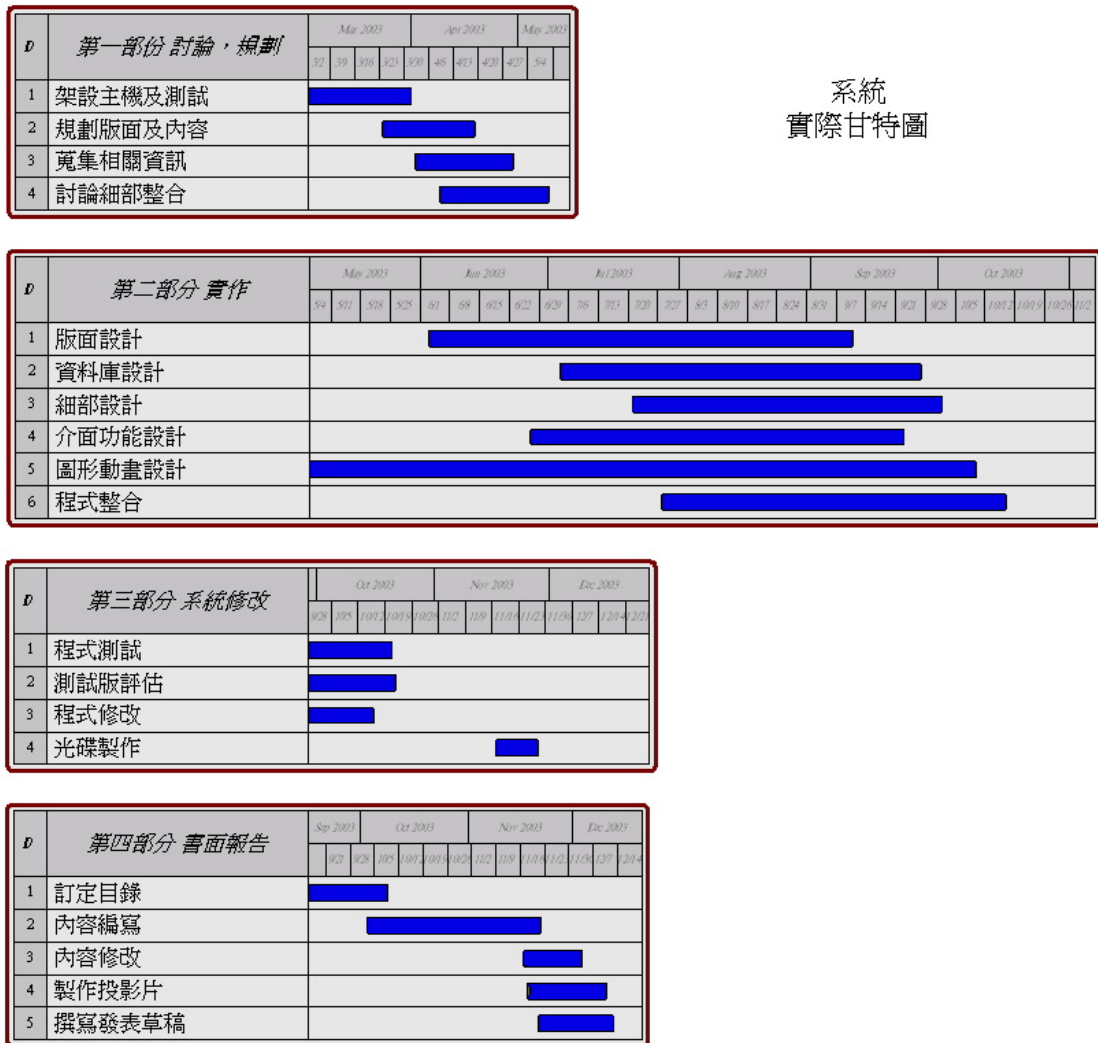


圖 2-2 系統實際甘特圖

第三章 系統分析與設計

此章節介紹了我們的系統架構、組織、功能以及介面，在此將可看到我們對於問題的解決方法。

3.1 系統架構

圖 3-1 為本系統的架構圖，先以使用者的觀點來觀看本系統。使用者分為訪客、老師、學生、系統管理者。

訪客即為所有來使用此系統的人，任何對於搜尋有興趣的使用者都可以來觀看、學習。

系統管理者則為整個系統的主宰，擁有整個系統的控制權，而老師的帳號、密碼，則為系統管理者輸入之後，以 E-MAIL 的方式通知老師，若是帳號發生錯誤，則老師需通知系統管理者修正。而學生的資料則是學生自行輸入，老師及系統管理者有權刪除學生帳號。

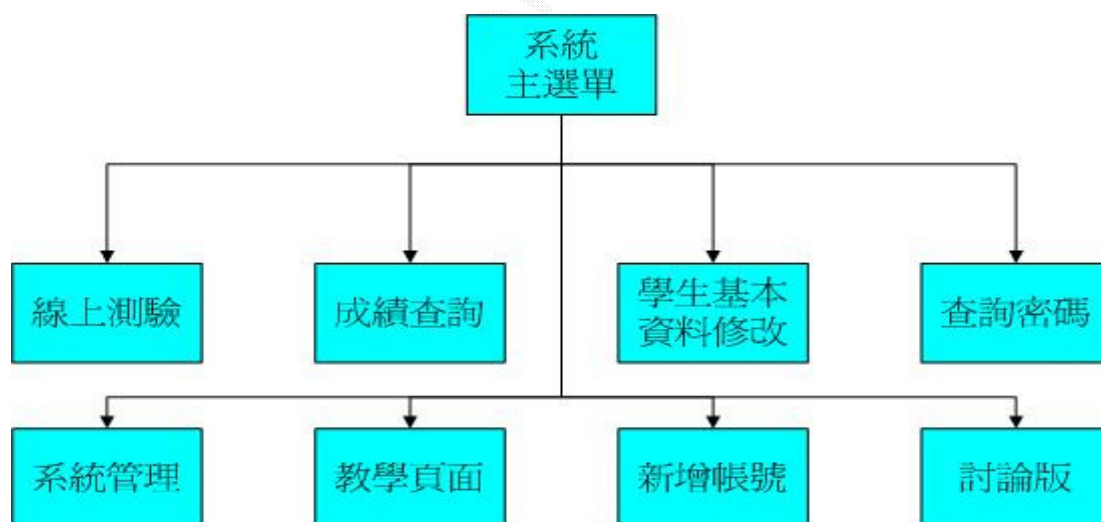


圖 3-1 系統架構圖

3.2 使用者觀點

圖 3-2 是所有的使用者對於本系統的觀點，使用者包含了訪客、老師、學生以及系統管理者。以權限來說，系統管理者能全盤操控整個系統，依次為老師、學生，最後是一般訪客。

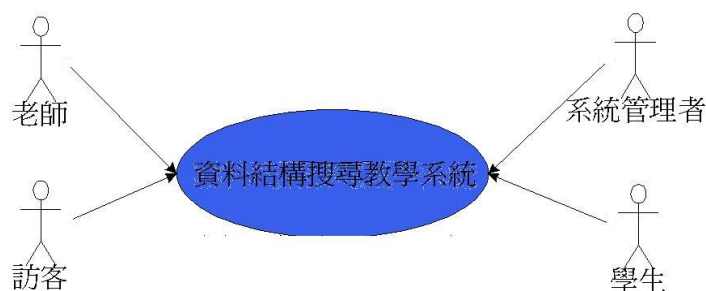


圖 3-2 使用者觀點圖

老師觀點

圖 3-3 是老師對於本系統的觀點，老師能做的管理較多。對於教學頁面，若是有需要修改的地方，則通知系統管理者修改。題庫管理使老師自由修正題目。

考卷管理，能製作考卷、修改考卷。帳號管理能刪除學生帳號。討論版管理，能刪除留言。考試管理，開放考卷、刪除學生錯誤成績。

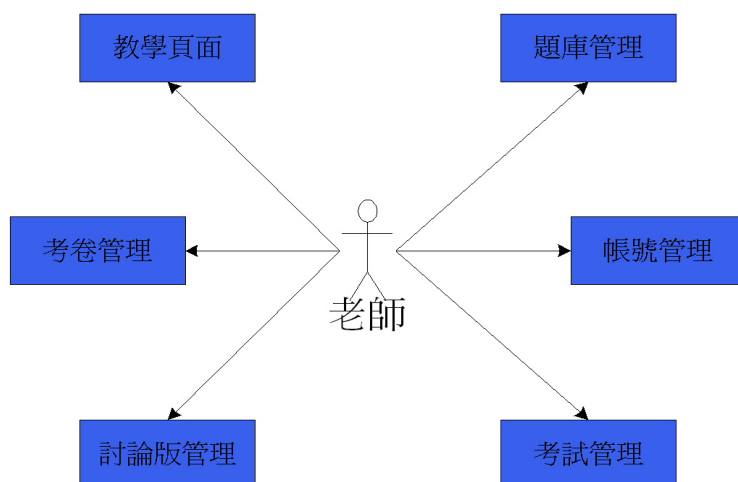


圖 3-3 老師觀點圖

學生觀點

圖 3-4 以學生的觀點來看，能觀看搜尋的教學、線上測驗、查詢成績、修改個人資料、到討論版發言或者回覆文章。

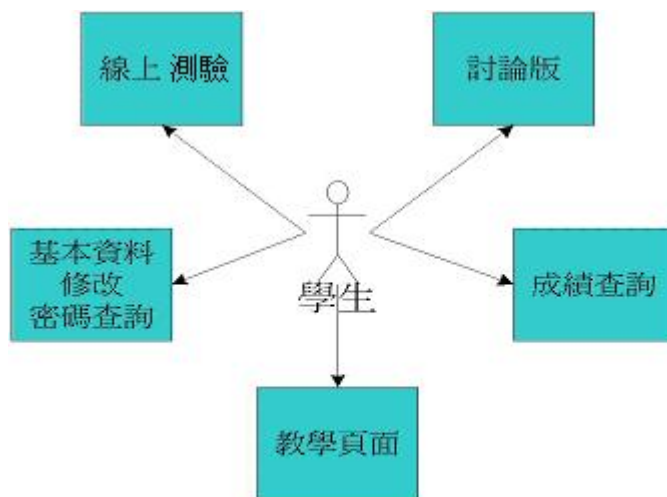


圖 3-4 學生觀點圖

訪客觀點

圖 3-5 顯示了訪客觀點圖，一般訪客能做的，就是瀏覽教學頁面以及到討論版發表問題。

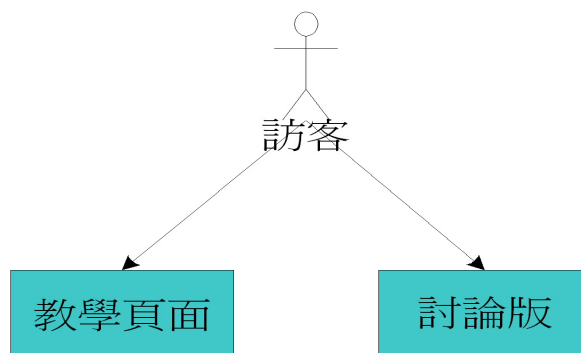


圖 3-5 訪客觀點圖

系統管理者觀點

圖 3-6 顯示了系統管理者觀點圖，系統管理者為系統的主宰，能對系統作任何的修改。

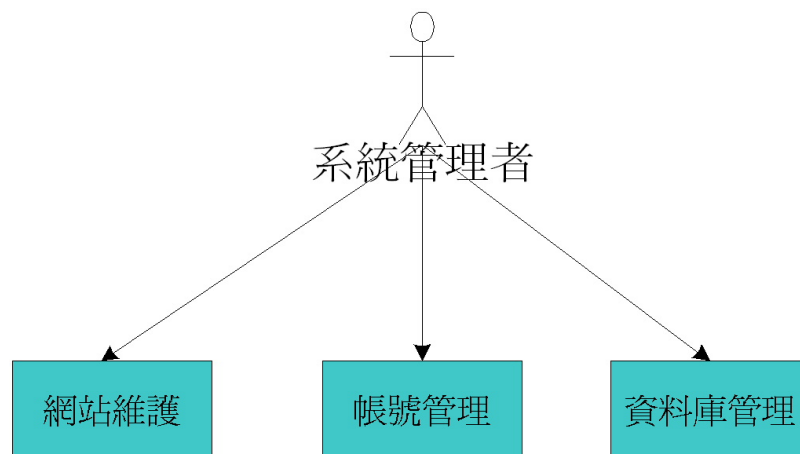


圖 3-6 系統管理者觀點圖

3.3 系統功能

本系統的功能分為以下幾個部分。線上教學、線上測驗、成績查詢、帳號管理、題庫管理、考卷管理、討論版。

3.3.1 線上教學

圖 3-7 展示了線上教學的架構圖，此子系統即為整個教學系統的主體。所有的訪客都能瀏覽頁面，學習資料結構的搜尋方法。倘若需要修改教學內容，則需老師寄 E-mail 給系統管理者以進行修改。

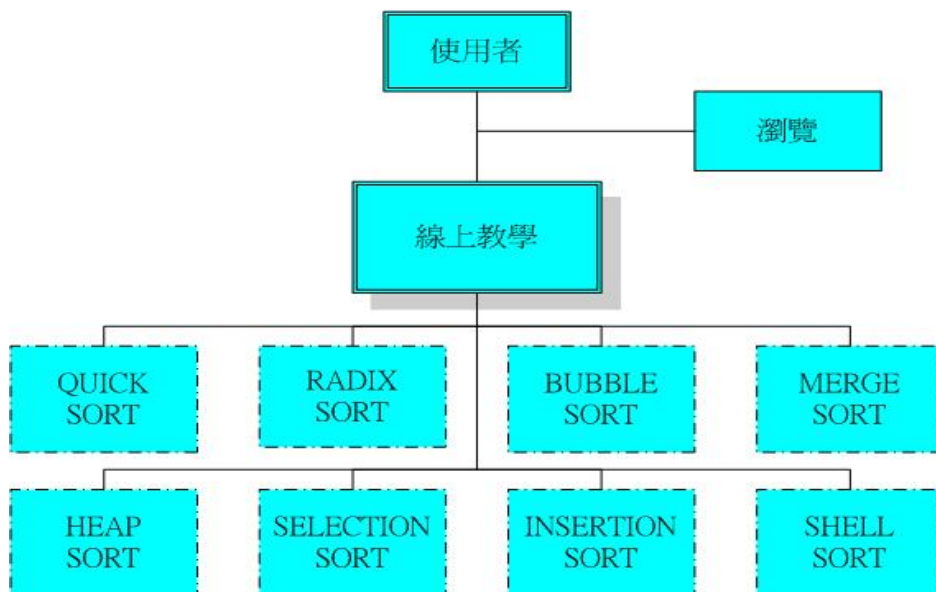


圖 3-7 線上教學架構圖

3.3.2 線上測驗

圖 3-8 為線上測驗的架構圖，此部分即為線上測驗功能，學生登入後，確認密碼無錯誤，無重複交卷後，即可參加考試，考試完成之後便計算、紀錄考試成績，還可觀看解答說明。

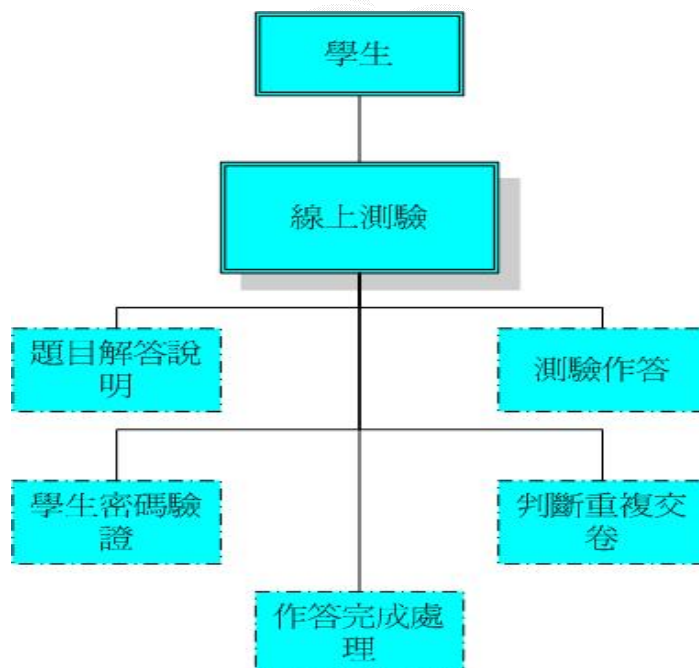


圖 3-8 線上測驗架構圖

3.3.3 成績查詢

圖 3-9 為成績查詢架構圖，此部分的功能，供學生察看每次的考試成績，以及每次的作答情形。可依考卷編號或學生班級來查詢學生成績，還能計算班級平均分數。

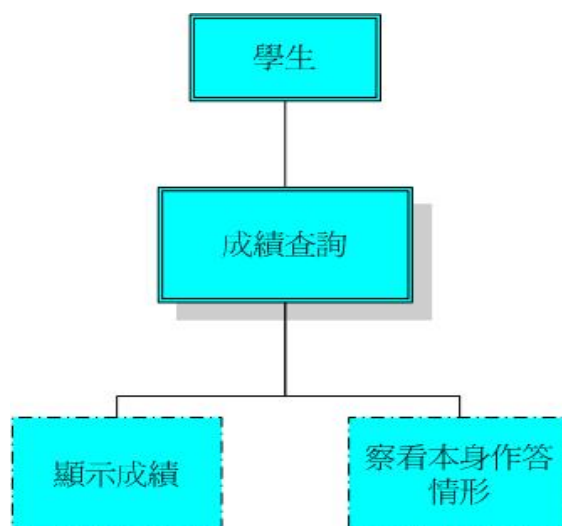


圖 3-9 成績查詢架構圖

3.3.4 學生資料修改

圖 3-10 為學生資料修改架構圖，此功能主要方便學生修改自身資料，修改前需要密碼確認，認可之後即能修改。

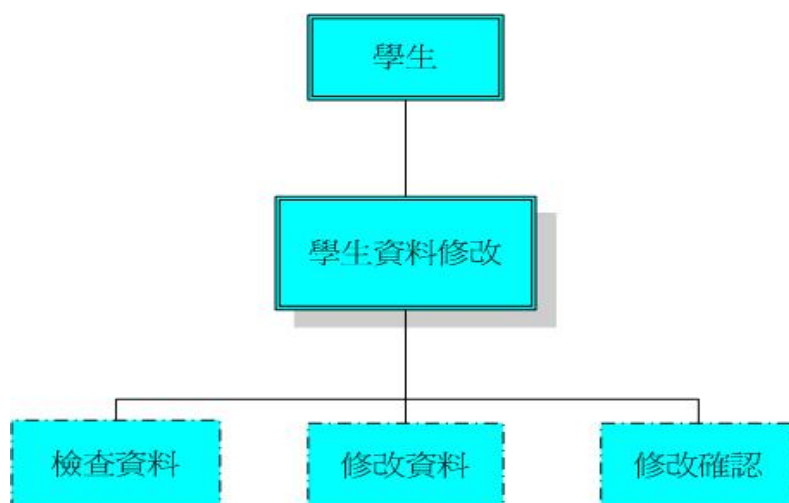


圖 3-10 學生資料修改架構圖

3.3.5 系統管理

圖 3-11 為系統管理架構圖，此系統分為四個子系統，有帳號管理、題庫管理、考卷管理、考試管理，各個子系統的功能由次小節詳述。

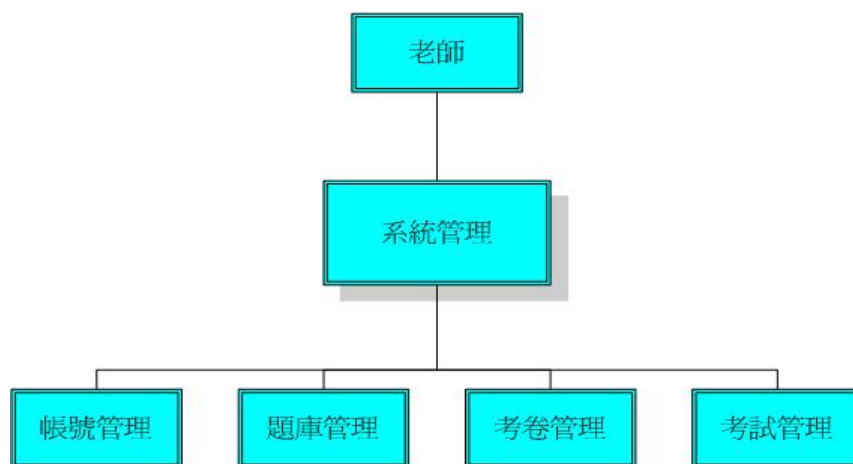


圖 3-11 系統管理架構圖

帳號管理

圖 3-12 為帳號管理架構圖，此功能主要讓老師能夠刪除學生的錯誤帳號。

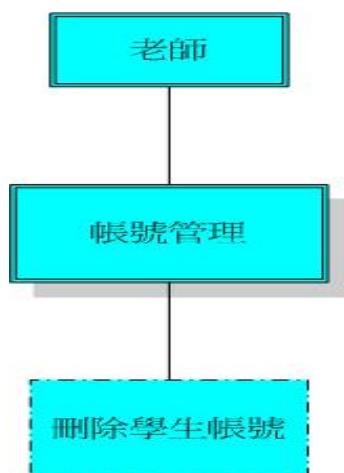


圖 3-12 帳號管理架構圖

題庫管理

圖 3-13 為題庫管理架構圖，線上測驗的題目可由老師直接管理，可以編輯題目、新增題目、瀏覽題目、新增單元、修改單元等。

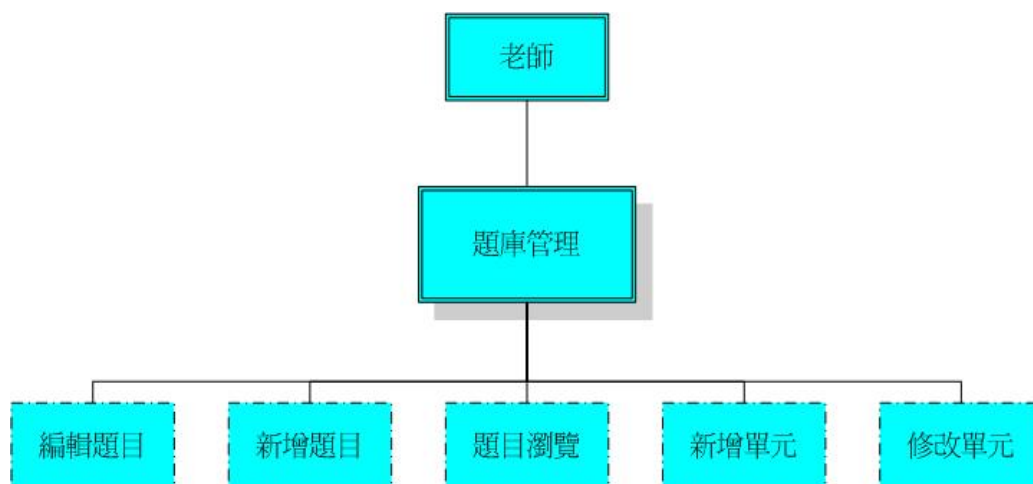


圖 3-13 題庫管理架構圖

考試管理

圖 3-14 為考試管理架構圖，此部分的功能為老師能開放考試，或者查詢或刪除學生的成績。

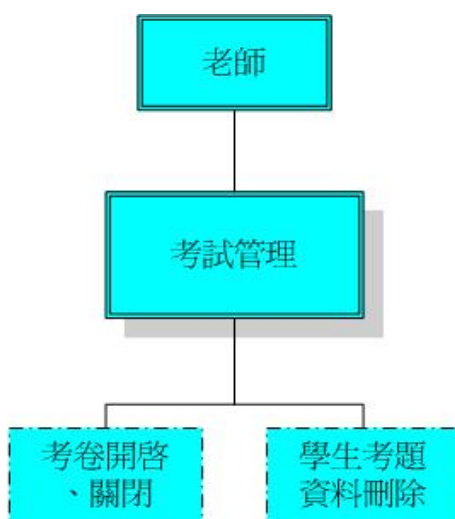


圖 3-14 考試管理架構圖

考卷管理

圖 3-15 為考卷管理架構圖，老師可由此子功能來管理考卷，由題庫選擇題目來建立一份新考卷，也可修改考卷的題目，或修改原有考卷的總題數。

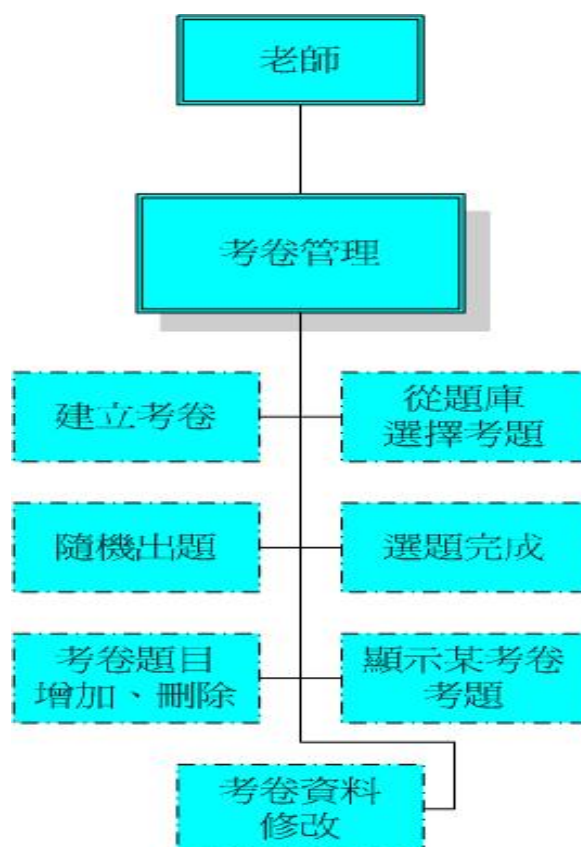


圖 3-15 考卷管理架構圖

3.3.6 討論版

圖 3-16 為討論版架構圖，此討論版提供訪客或學生發表問題，本討論版沒有限制發問人的身份，意即所有人都可以發表問題或回覆問題。若有不當發言，則老師有權刪除該筆文章。

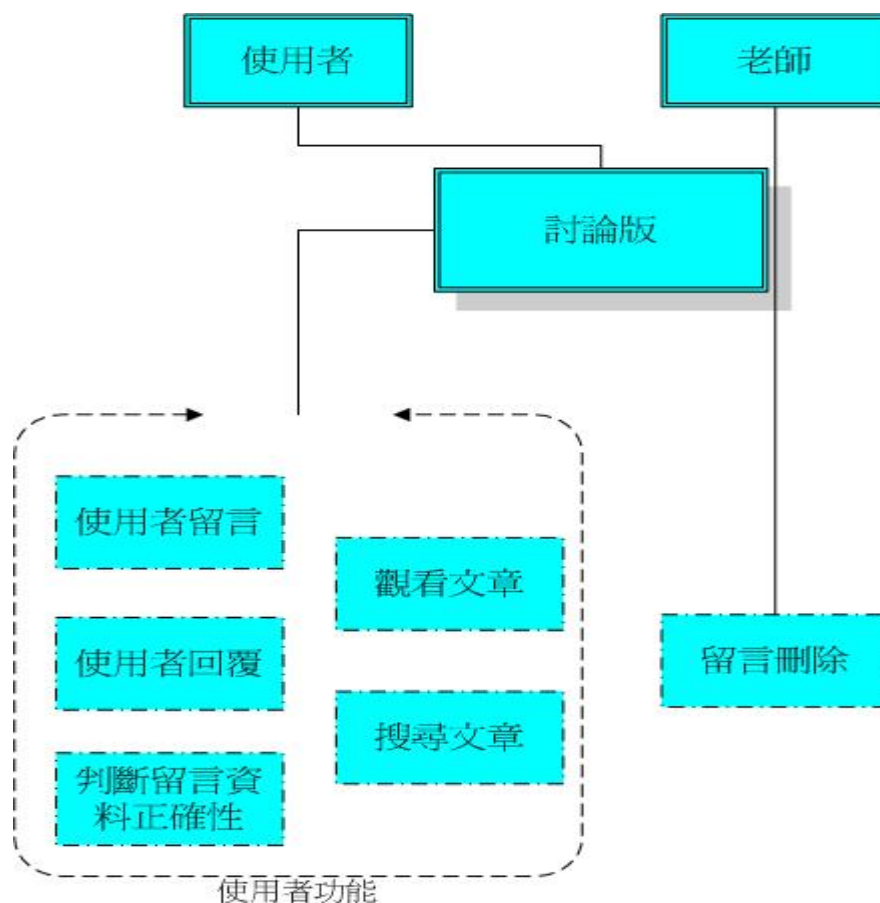


圖 3-16 討論版架構圖

3.4 系統流程

系統的流程圖，顯示了使用者進入系統後所有的動作流程。此章節說明了各類使用者進入本系統的流程。

3.4.1 線上測驗流程

圖 3-17 展示了學生進入考試系統的流程，學生需輸入自己的帳號、密碼以及考卷代號，帳號、密碼經過確認之後，需要在確認考卷代號正確無誤之後，學生才能進入考試。接著就開始選擇答案，學生作答完畢後，按下交卷鈕就完成考試，隨即將此次考試結果寫入考試資料庫。

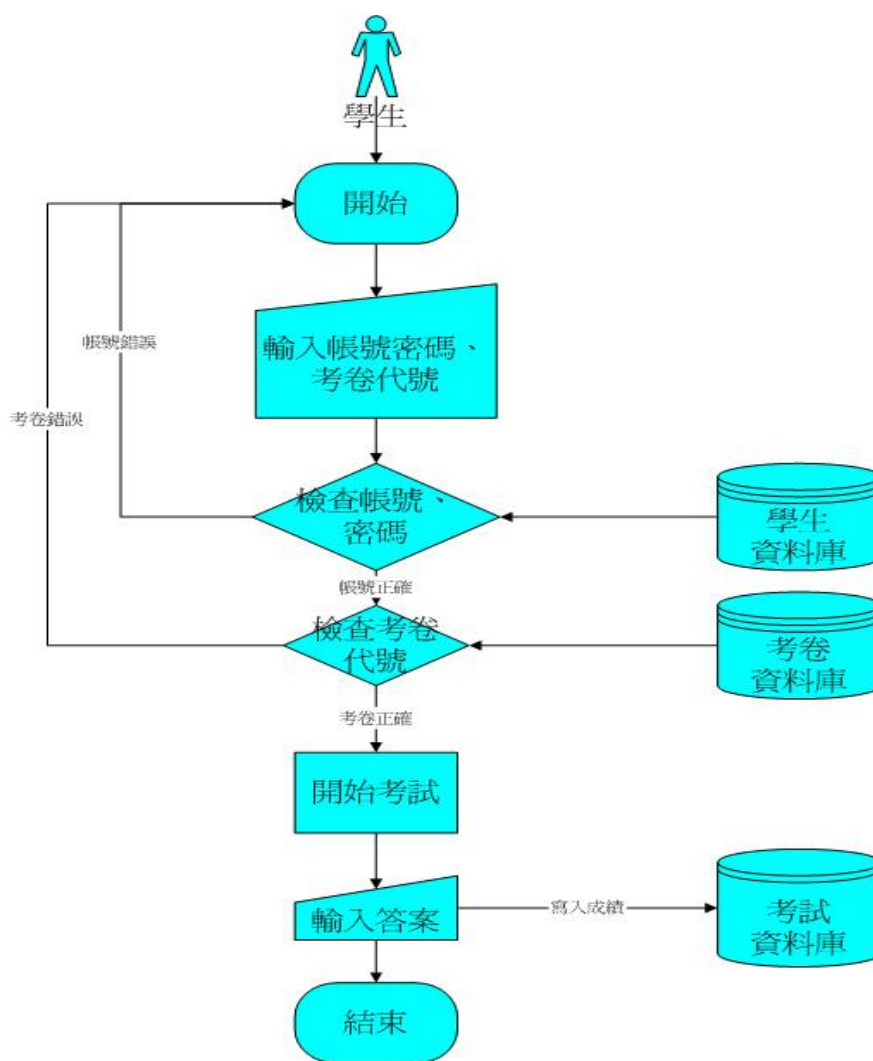


圖 3-17 線上測驗流程圖

3.4.2 成績查詢流程

圖 3-18 展示了學生查詢成績的流程，學生需輸入帳號密碼，經過學生資料庫的比對之後，再選擇考卷編號或班級來觀看成績，也可觀看自己的作答歷史。

3.4.3 學生基本資料修改

圖 3-19 展示了學生修改基本資料的流程，學生進入後需輸入帳號、密碼，經學生資料庫比對資料後，才可進入修改頁面，輸入欲修改的部分後，即可更改學生基本資料，學生資料庫裡的資料進行更新。

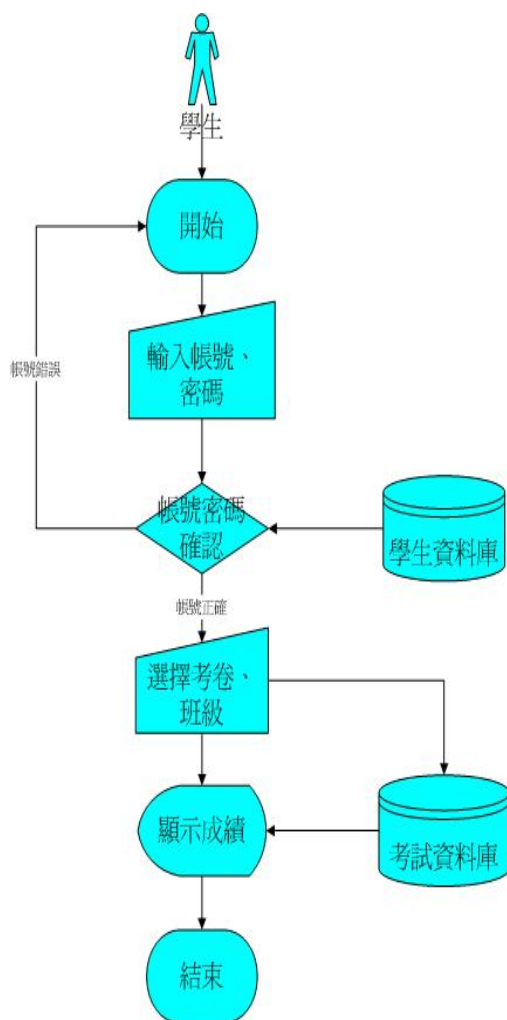


圖 3-18 成績查詢流程圖

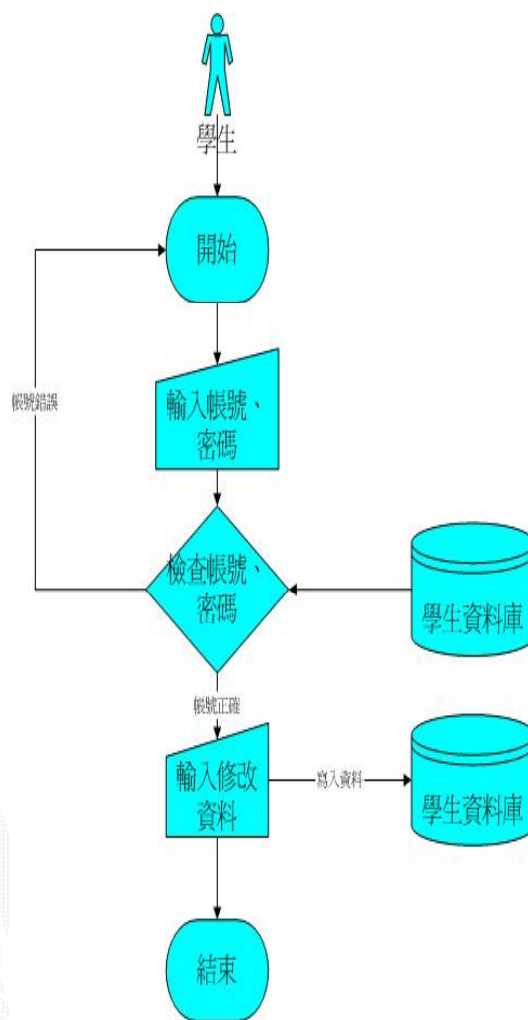


圖 3-19 學生基本資料修改流程圖

3.4.4 系統管理流程

此部分為老師管理此系統之流程圖，由於還分支出五個部分，整個圖會過於龐大，因此再細分為五個子流程圖。

圖 3-20 入管理頁面前，需先作帳號、密碼的確認，才能擁有系統管理的權限。

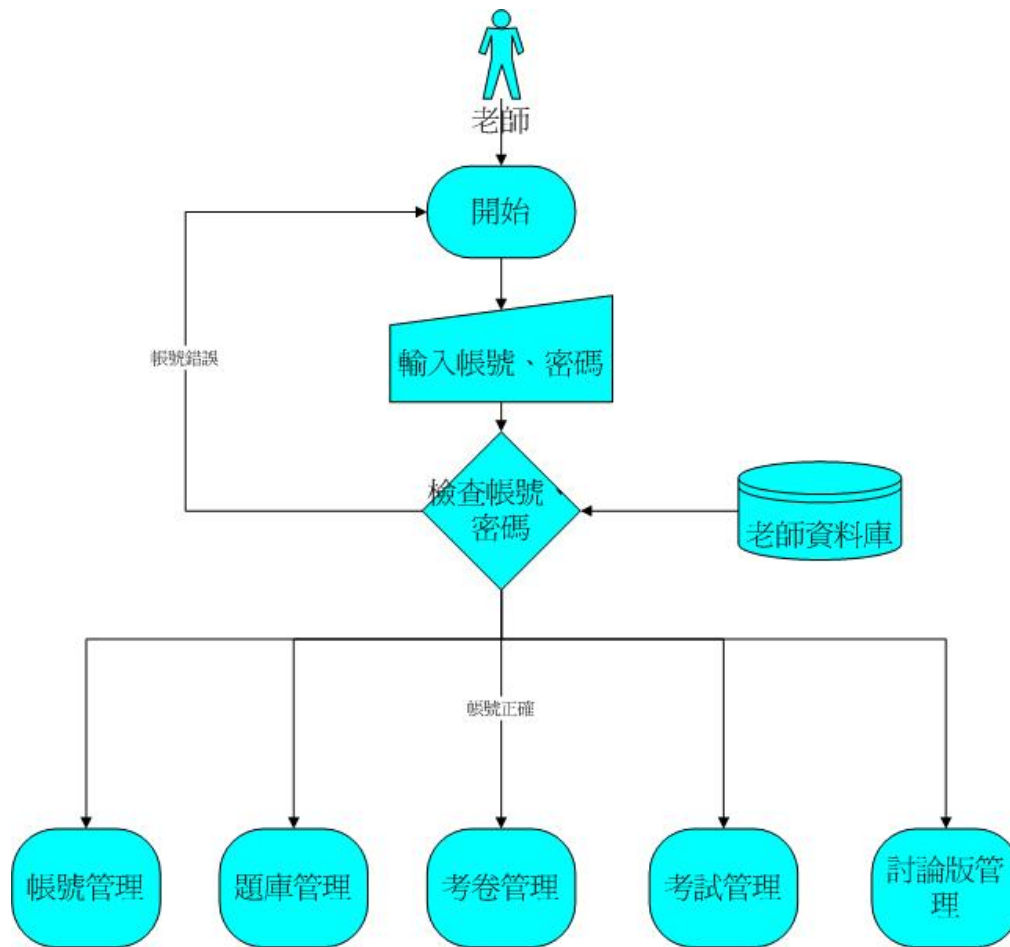


圖 3-20 系統管理流程圖

3.4.5 帳號管理流程

圖 3-21 說明了老師刪除學生帳號的流程圖，老師輸入欲刪除的學生帳號範圍後，系統即搜尋學生資料庫裡符合的學生帳號，並將之刪除，倘若搜尋無任何資料，便不做更新資料庫的動作。

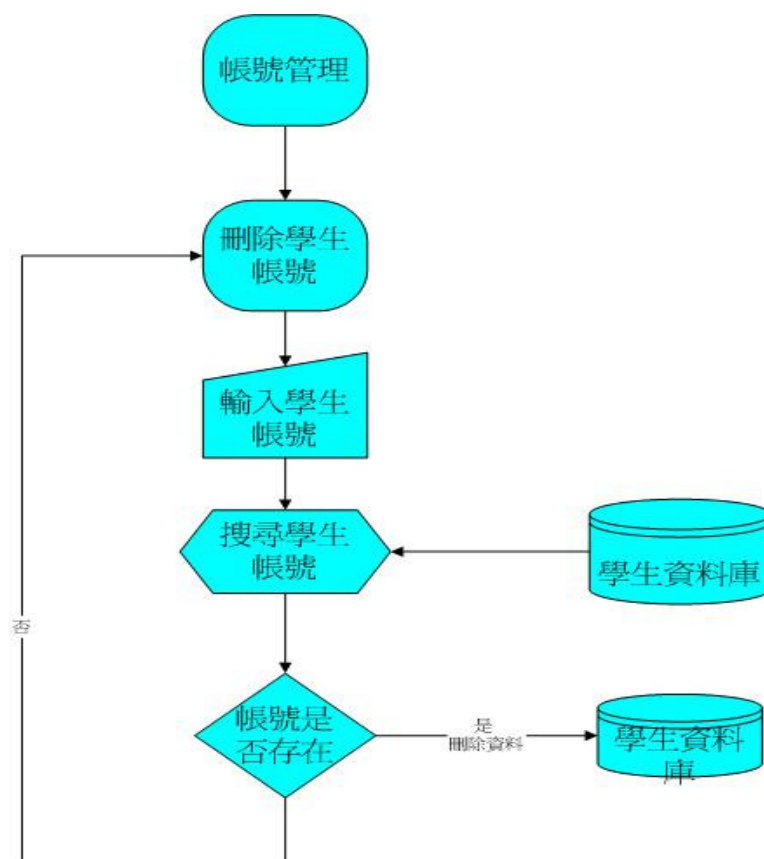


圖 3-21 帳號管理流程圖

3.4.6 題庫管理流程

圖 3-22 進入題庫管理之後，便可瀏覽所有題目，或依章節來顯示題目。其他兩個功能分別是新增題目，新增、修改、刪除章節。

進入新增題目後，首先選擇此題目所屬章節，接著便可輸入題目，確認後即寫入題目資料庫。新增、修改、刪除章節，新增的功能一經輸入後隨即寫入資料庫。

而刪除章節的功能，輸入欲刪除的章節名稱後，系統即搜尋符合的章節名稱，若有符合的就刪除。

最後是修改章節名稱，同樣的，搜尋到符合的章節名稱後才會更改名稱。

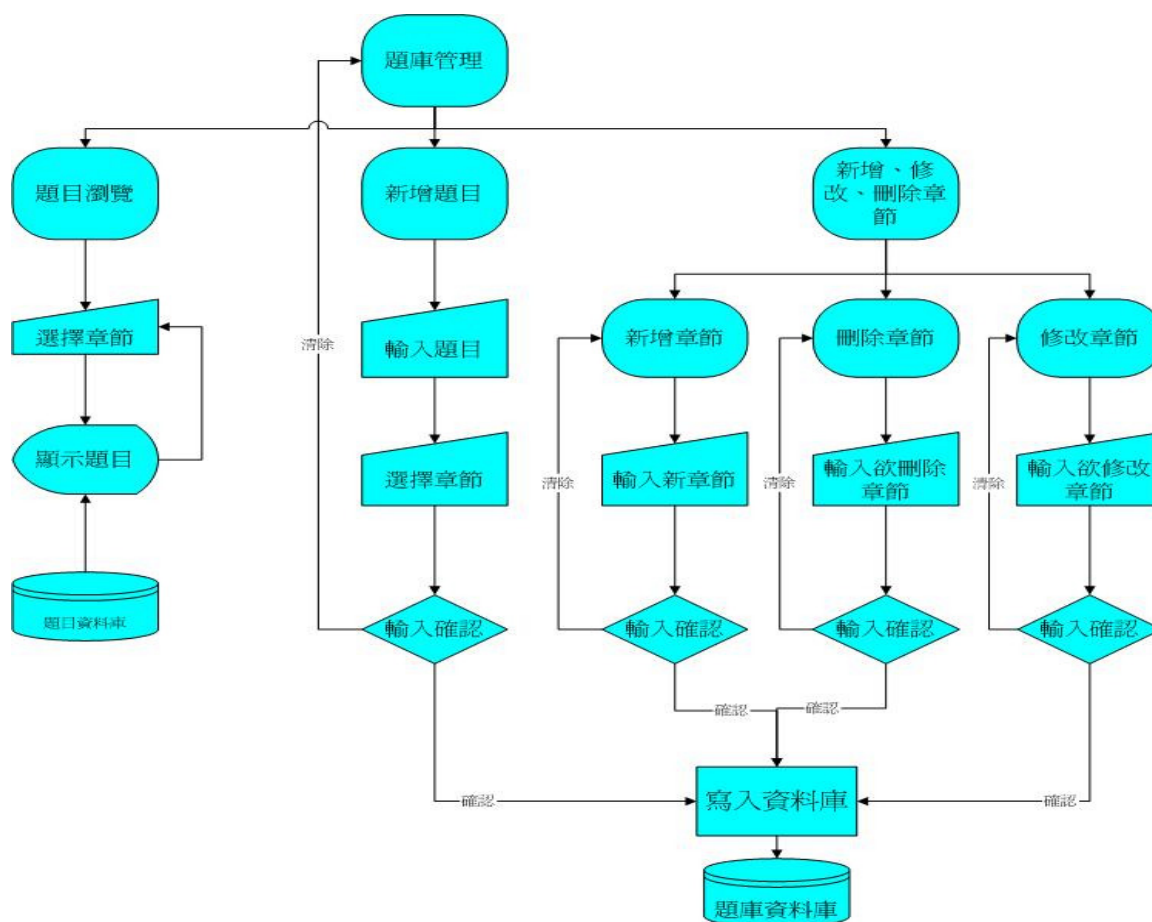


圖 3-22 題庫管理流程圖

3.4.7 考卷管理流程

如圖 3-23，考卷管理分為兩個主流程，一個是建立新考卷，可由老師自行選擇題目，或交由系統亂數出題，而題目足夠後就完成一份考卷，並寫入考卷資料庫。另一個功能是考卷的瀏覽、修改及刪除考卷，可以修改此份考卷的題目、題目總數、或是直接刪除一份考卷，確認後即更新資料庫。

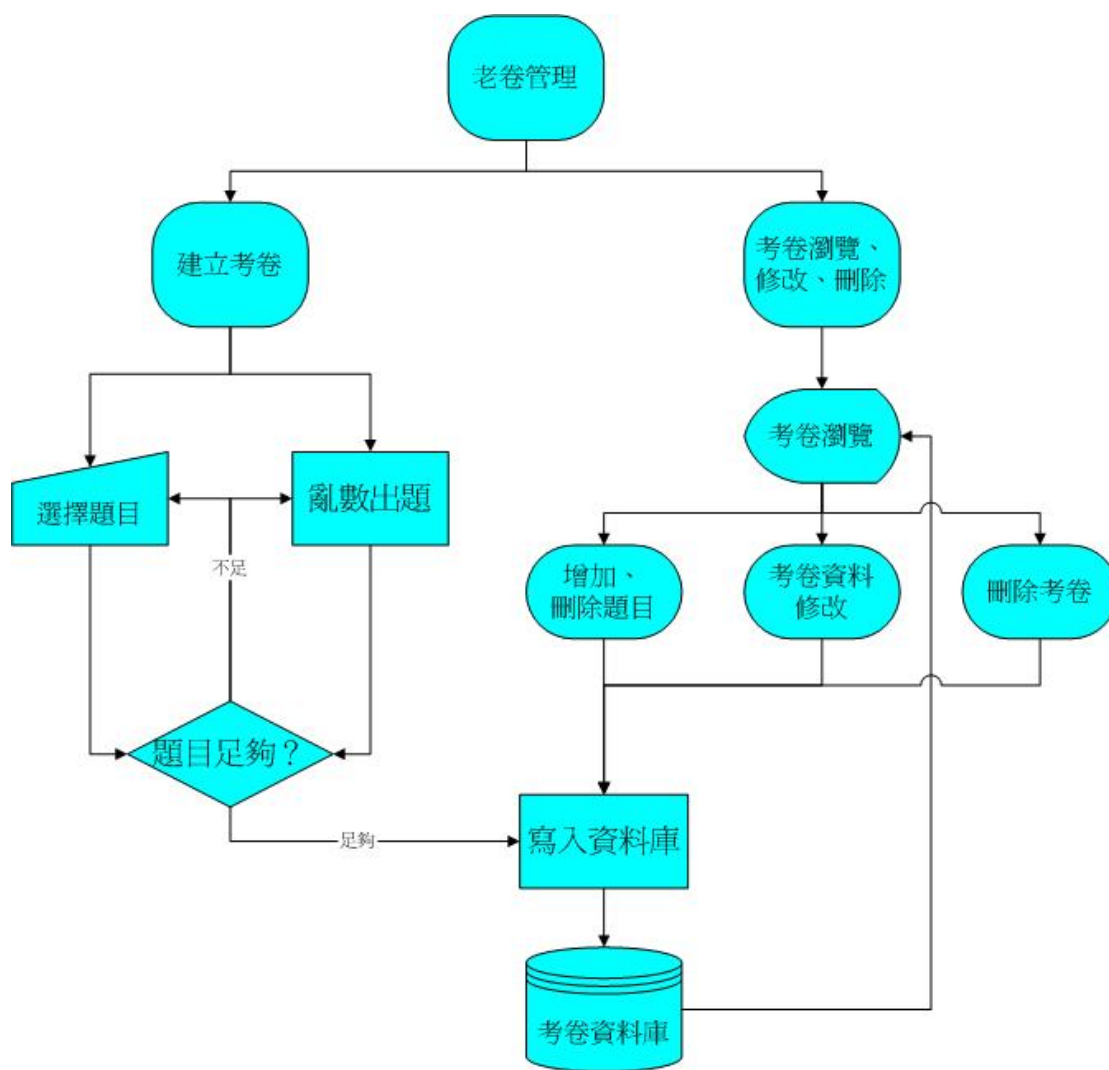


圖 3-23 考卷管理流程圖

3.4.8 考試管理流程

如圖 3-24，考試管理也分為兩個流程，老師可以決定考卷是否開啟，也可隨時關閉，但是要開啟一份考卷，則需要老師將此份題目出完，才能開啟考卷。

而刪除學生錯誤成績，老師需選擇考卷編號，才能顯示出所有參加此考試的學生成績，而選擇刪除成績之後，學生成績就從資料庫裡移除。

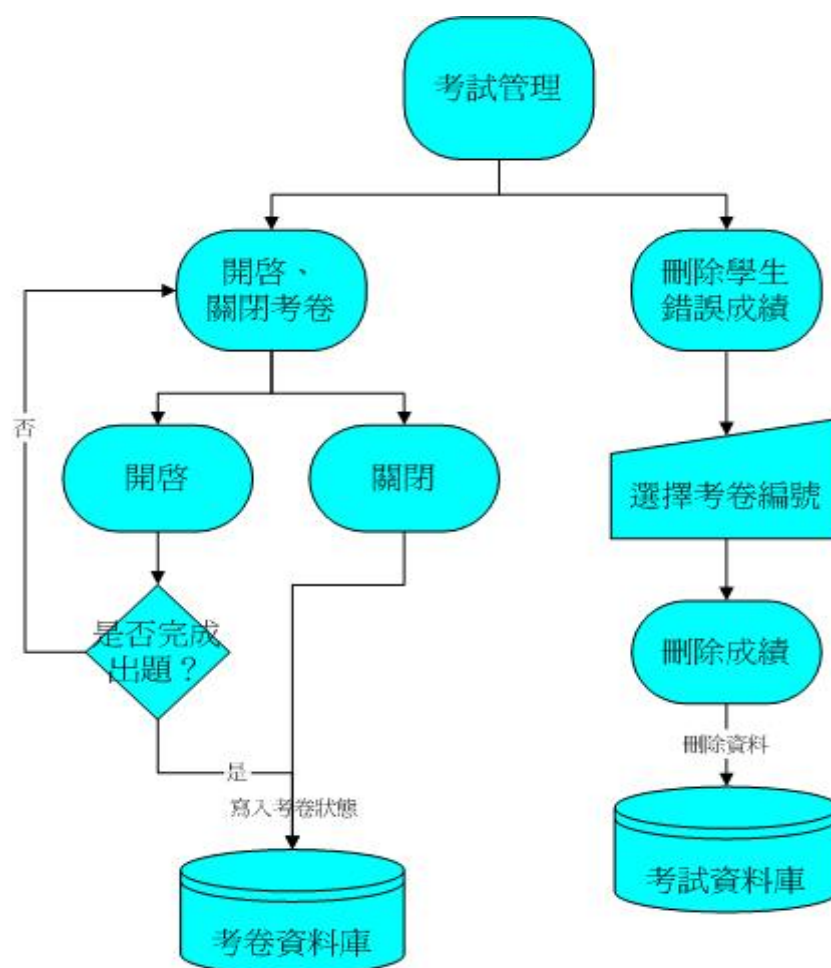


圖 3-24 考試管理流程圖

3.4.9 討論版管理流程

圖 3-25 展示了老師進入後，即可觀看所有的文章，看到想刪除的文章後，便可直接刪除。

3.4.10 新增帳號流程

圖 3-26 說明了學生新增帳號的流程，學生進入後就填入基本資料，而在送出資料後，系統會判斷兩次輸入的密碼是否相同、帳號是否重複，若出現密碼求同或帳號相同則需重新輸入，反之即寫入資料庫。

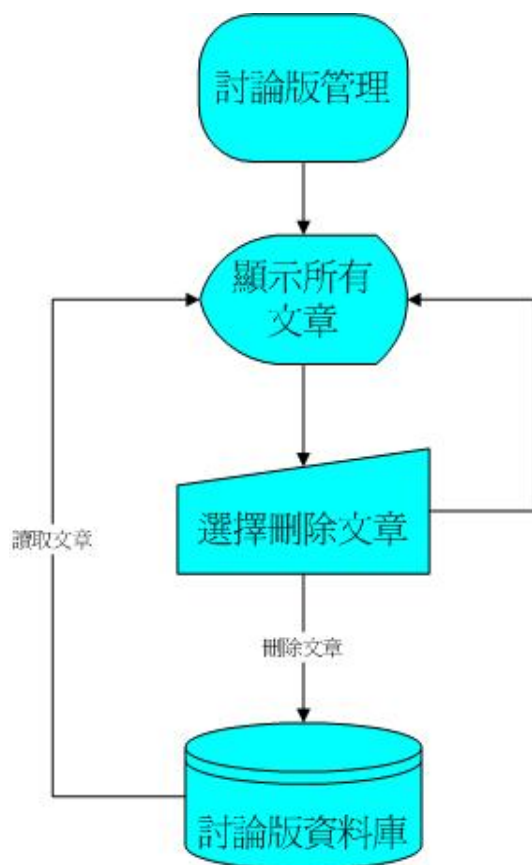


圖 3-25 討論版管理流程圖

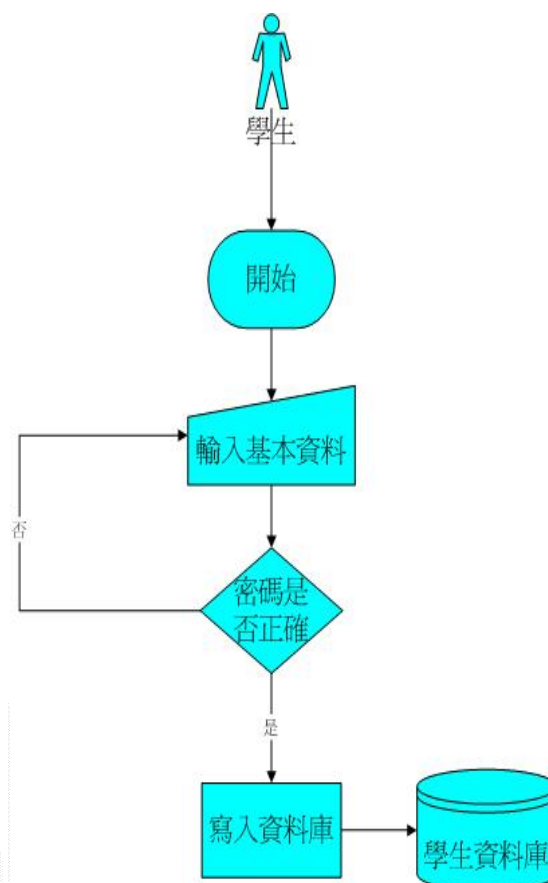


圖 3-26 新增帳號流程圖

3.4.11 查詢密碼流程

圖 3-27 展示了學生查詢密碼的流程，系統會要求使用者輸入學號、帳號、姓名以及 E-mail，經過學生資料庫的比對正確之後，就會在網頁上直接顯現出密碼。

3.4.12 討論版流程

圖 3-28 顯示了進入討論版的流程，這邊屬於開放式的討論版，不需任何密碼確認，因此輸入文章之後，隨即寫入討論版資料庫。

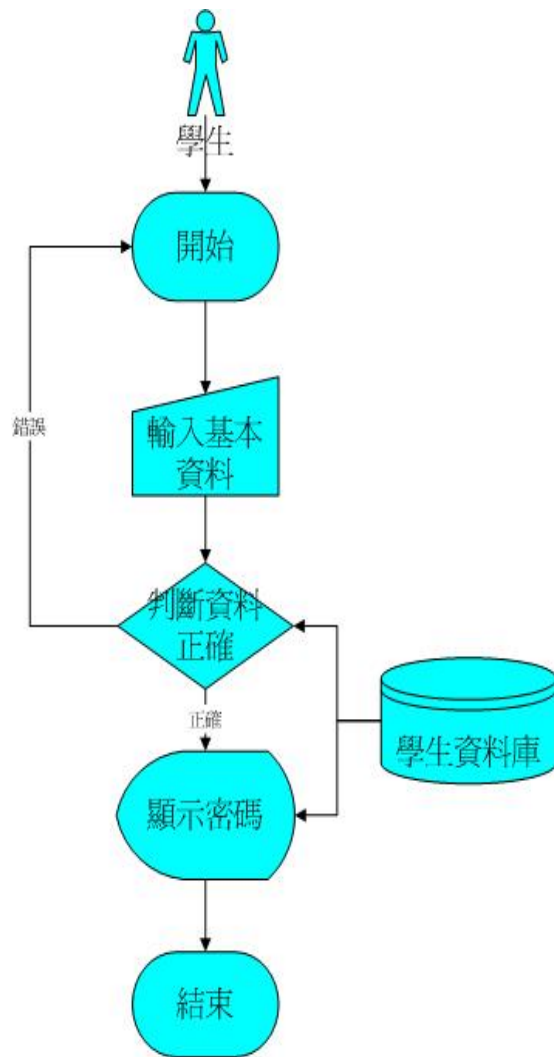


圖 3-27 查詢密碼流程圖

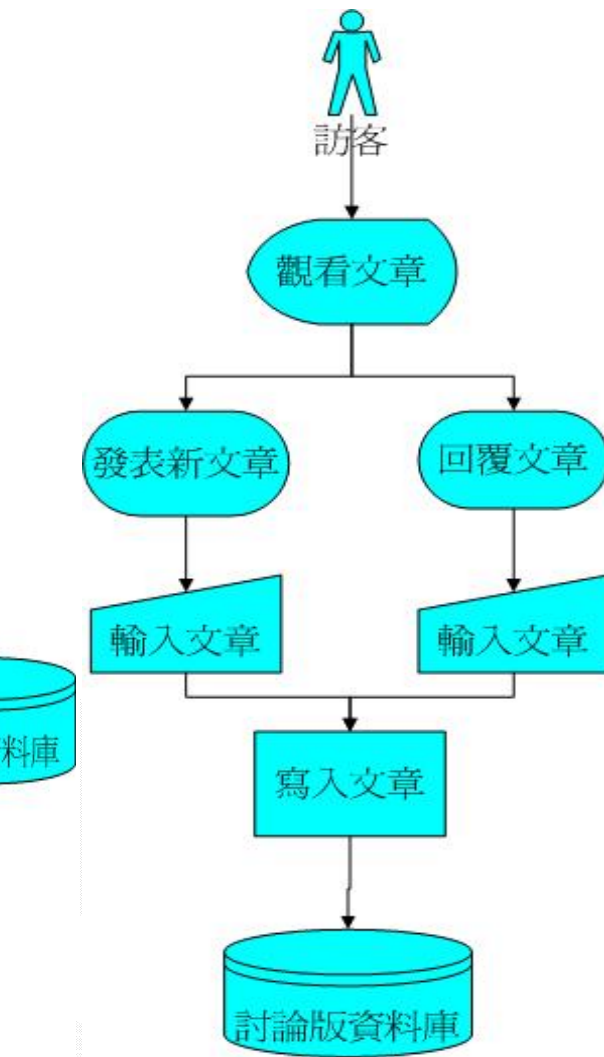


圖 3-28 討論版流程圖

第四章 系統實作

此章節介紹了我們系統實作的方法，教學單元介紹、教學頁面製作、PHP 網頁製作、資料庫規劃、設計。

4.1 教學單元介紹

本系統主要在說明排序的方法，而我們實作了八種最著名的排序法，先對排序做個基本的原理介紹和範例。

BUBBLE-SORT 說明

泡沫排序法可以說是最簡單且最直觀的排序法，一開始所有的資料都放在同一個陣列中，其排序的策略就是由資料的最左邊開始往右比較，只要資料值比右邊一個的資料值大，則兩筆資料相互交換，直到最右邊的資料為止，這時候，最右邊的資料即是最大值了，下一次還是由最左邊的元素開始往右邊比較和交換，但最右邊已經是最大值，所以只比到倒數第二筆資料就停止，這時候第二大的資料就會出現在陣列的右邊第二個，依此類推，直到陣列中只剩下一筆資料時，所有的資料排序已經完成了。主要程式碼如圖 4-1 圖所示。

```
void bubblesort(int a[], int n)
{
    int i, j, t;
    for(i=n-1; i>0; i--)
        for(j=0; j<i; j++)
            if(a[j]>a[j+1])
            {
                t=a[j];
                a[j]=a[j+1];
                a[j+1]=t;
            }
}
```

圖 4-1 BUBBLE-SORT 程式碼

例如輸入為：21, 3, 13, 25, 9

第 1 次執行結果：3, 13, 21, 9, 25

第 2 次執行結果：3, 13, 9, 21, 25

第 3 次執行結果：3, 9, 13, 21, 25

第 4 次執行結果：3, 9, 13, 21, 25

HEAP-SORT 說明

堆積排序是利用堆積的特性完成資料的排序，假設檔案有 n 個記錄 R_1, R_2, \dots, R_n ，其鍵值分別為 K_1, K_2, \dots, K_n 。由於堆積是一個完整的二元樹，首先依序將記錄放入以建立一個完整二元樹。為了利用堆積排序將資料由小到大排序，我們可以將堆積排序的步驟大致上分為以下 4 點：

1. 將所有紀錄依序加入建成一個完整二元樹。
2. 將完整二元樹調整成最大堆積樹。
3. 將樹根跟最後一個節點的位置對調，並且扣除原樹根節點以外的完整二元樹並調整為最大堆積樹。

重複本步驟，直到最大堆積樹中已沒有任一節點。

4. 最後依序印出儲存最大堆積樹的陣列的每個元素值，便可以得到由小到大排序後的結果。

1. 副程式 Adjust 的過程是先以 root 所指的樹根和它的兩個子節點中最大者相比，若較子節點大則不必調整，若較子節點小則將較大的子節點移至樹根的位置，並繼續與下一層的子節點比較，重複這個步驟直到子節點比它小為止，此時才將原來的樹根節點放到合適的位置上。

2. 主程式中，剛開始的第一個 for 迴圈即是將完整二元樹調整為最大堆積樹。而調整方式是：由右而左，由下而上逐層調整

每個子樹，最後使整個樹變成最大堆積樹，首先調整以最後一個節點的父節點為根，(在 $\lfloor n/2 \rfloor$ 的位置)的子樹，接著在調整以同層左邊的節點為根的子樹，若同層左邊沒有其他節點，則調整以上一層最右邊的節點為根的子樹。重複這個步驟，直到整個樹都被調整過，即形成最大堆積樹，經過這些步驟後，可以確保最後整個樹可以形成一個最大堆積樹，因為每次調整後，較大的節點會被移到上層，父節點的鍵值大於小節點的鍵值，所以滿足最大堆積樹的條件。主要程式碼如圖 4-2 圖所示。

```

void Heap sort(Element *list, const int n)
{   for(int n/2; i>=1; i--)
    adjust(list, i, n);
    for(i=n-1; i>=1; i--)
    {   Element t=list[i+1];
        list[i+1]=list[i];
        list[i]=t;
        adjust(list, i, n);
    }
}

void adjust(Element *tree, const int root, const int n)
{   Element e=tree[root];
    int k=e.getKey();
    for(int j=2*root; j<=n; j*=2){
        if(j<n)
            if(tree[j].getKey()<tree[j+1].getKey())j++;
            if(k>=tree[j].getKey())break;
            tree[j/2]=tree[j];
        }
        tree[j/2]=e;
    }
}

```

圖 4-2 HEAP-SORT 程式碼

例如輸入為：26, 5, 77, 1, 61, 11, 59, 15, 48, 19

第 1 次執行結果：77

第 2 次執行結果：61, 77

第 3 次執行結果：59, 61, 77

第 4 次執行結果：48, 59, 61, 77

第 5 次執行結果：26, 48, 59, 61, 77

第 6 次執行結果：19, 26, 48, 59, 61, 77

第 7 次執行結果：15, 19, 26, 48, 59, 61, 77

第 8 次執行結果：11, 15, 19, 26, 48, 59, 61, 77

第 9 次執行結果：5, 11, 15, 19, 26, 48, 59, 61, 77

第 10 次執行結果：1, 5, 11, 15, 19, 26, 48, 59, 61, 77

INSERTION-SORT 說明

所謂的插入排序簡單的說就像我們在玩撲克牌時，整理手中的牌時就是使用插入排序。每當拿到一張新的牌時，就會依據它的大小插在正確的位置，到最後手上的牌就會按小到大排好了。

在一串已按鍵值由小到大排序好的 i 個記錄 R_1, R_2, \dots, R_i 中(其鍵值為 K_1, K_2, \dots, K_i ，且 $K_1 \leq K_2 \leq \dots \leq K_i$)，插入一個記錄 R ，使得這 $i+1$ 個記錄形成一個新的有序串列 R_1, R_2, \dots, R_{i+1} ，且 $K_1 \leq K_2 \leq \dots \leq K_{i+1}$ 。插入排序的每個階段皆產生一個已排序好的一組資料。所謂"插入"的過程是在前面 i 個記錄中插入第 $i+1$ 個記錄，其鍵值為 K 。首先將第 i 個記錄的鍵值 K_i 和 K 相比較，若 $K_i > K$ 則將 R_i 往後移動一個位置，接著在比較 K 和 R_{i-1} 的鍵值 K_{i-1}, \dots 。重複這個過程直到遇到第 j 個記錄，其鍵值 K_j 小於或等於 K ，這時才將 K 放在 $j+1$ 的位置。因為有可能這 i 個記錄的鍵值都大於或等於新插入的鍵值 K ，這樣上述的過程會無法終止，所以必須事先在最前端加入一個虛擬的紀錄 R_0 ，並將它的鍵值設為極小的數，令 $K_0 = -\infty$ 。此法的時間複雜度為 $O(N^2)$ 。主要程式碼如圖 4-3 圖所示。

```
void insertion-sort(element list[], int n)
{
    int i, j;
    element next;
    for(i=1; i<n; i++){
        next=list[i];
        for(j=i-1; j>=0&&next.key<list[j].key; j--;)
            list[j+1]=list[j];
        list[j+1]=next;
    }
}
```

圖 4-3 INSERTION-SORT 程式碼

例如輸入為：23, 12, 6, 43, 31

第 1 次執行結果：6, 23, 12, 43, 31

第 2 次執行結果：6, 12, 23, 43, 31

第 3 次執行結果：6, 12, 23, 31, 43

第 4 次執行結果：6, 12, 23, 31, 43

MERGE-SORT 說明

合併排序簡單的說就是將兩個已經排序好檔案 (X_1, X_2, \dots, X_m) 和 ($X_{m+1}, X_{m+2}, \dots, X_n$) 合併成另一個檔案 (Y_1, Y_2, \dots, Y_n)。合併排序中最常用的就是 2-way merge sort，每次將相鄰的兩個檔案合併成一個排序好的檔案。首先將檔案中的 n 個資料視為長度為 1 的小檔案，然後將相鄰的小檔案兩兩合併成 $n/2$ 個排序好的檔案，長度為 2，依此類推，再逐漸將檔案合併成長度為 4, 8, 16, \dots 的檔案，在合併已排序的兩個檔案時是按照前面所述的方式，最後當合併成一個排序好的檔案時即完成合併排序。主要程式碼如圖 4-4 圖所示。

```
void merge(element list[],element sorted[],int i,int n,int m)
{
    int t,j,k; j=m+1; k=i;
    while(i<=m&& j<=n){
        if(list[i].key<=list[j].key)
            sorted[k++]=list[i++];
        else
            sorted[k++]=list[j++]; }
    if(i>m)
        for(t=j; t<=n; t++)
            sorted[k+t-j]=list[t];
    else
        for(t=i ;t<=m; t++)
            sorted[k+t-i]=list[t];
}
```

圖 4-4 MERGE-SORT 程式碼

例如輸入為：26, 5, 77, 1, 61, 11, 59, 15, 48, 19

第 1 次執行結果：5, 26, 1, 77, 11, 61, 15, 59, 19, 48

第 2 次執行結果：1, 5, 26, 77, 11, 15, 59, 61, 19, 48

第 3 次執行結果：1, 5, 11, 15, 26, 59, 61, 77, 19, 48

第 4 次執行結果：1, 5, 11, 15, 19, 26, 48, 59, 61, 77

QUICK-SORT 說明

快速排序是具有最佳平均執行時間的排序方法。在 n 個記錄 R_1, R_2, \dots, R_n 中，將 R_i 放在 $s(i)$ 的位置，使得若 $j < s(i)$ ，則 $K_j \leq K_{s(i)}$ ；若 $j > s(i)$ ，則 $K_j \geq K_{s(i)}$ ，也就是將 R_i 左邊的鍵值都小於 K_i ，而在 R_i 右邊的鍵值大於 K_i ，這就是一次的分隔 (partition) 的過程。快速排序就是經過許多次的分隔的過程將整個檔案排序。分隔後的整個檔案分成兩個子檔案，一個包含記錄 $R_1, R_2, \dots, R_{s(i)-1}$ ，另一個包含記錄 $R_{s(i)+1}, \dots, R_n$ ，而 $R_{s(i)}$ 則是在兩者之中：

$(R_1, R_2, \dots, R_{s(i)-1}) \quad R_{s(i)} \quad (R_{s(i)+1}, \dots, R_n)$

分隔後 $R_{s(i)}$ 的位置是相對於整個檔案來說是正確的，不需在改變它的位置，而左右的兩個檔案在分別進行快速排序。若所有的子檔案都處理完，則快速排序的過程才算完成。快速排序後每個記錄都在正確的位置上，並且是按鍵值由小到大排序好。快速排序的主要精神是在分隔的過程，分隔指利用其中的某個鍵值 (通常是第一個記錄的鍵值) 作為分隔元素 (Partition Element) 或軸 (Pivot Key)，並將其所有紀錄分為兩組，其中一組的鍵值比分隔元素小，另一組的鍵值則比分隔元素大。分隔元素置於兩者之間，因為比它小的都在它的左邊，比它大的都在右邊，所以分隔後分隔元素的所在位置是正確的不需再更動。因此，至多也經過 n 次回合的分隔過程後，快速排序也就到此結束)。主要程式碼如圖 4-5 圖所示。

```
void quicksort(element list[], int left, int right, )
{ int pivot, i, j;
  element temp;
  if(left<right){
    i=left; j=right+1;
    pivot=list[left].key;
    do{
```

```

do    i++;
while(list[i].key<pivot);
do    j--;
    while(list[j].key>pivot);
    if(i<j)
        SWAP(list[i], list[j], temp);
    }while(i<j);
SWAP(list[left], list[j], temp);
quicksort(list, left, j-1);
quicksort(list, j+1, right);
    }
}

```

圖 4-5 QUICK-SORT 程式碼

例如輸入為：26, 5, 37, 1, 61, 11, 59, 15, 48, 19

第 1 次執行結果：11, 5, 19, 1, 15, 26, 59, 61, 48, 37

第 2 次執行結果：1, 5, 11, 19, 15, 26, 59, 61, 48, 37

第 3 次執行結果：1, 5, 11, 19, 15, 26, 59, 61, 48, 37

第 4 次執行結果：1, 5, 11, 15, 19, 26, 59, 61, 48, 37

第 5 次執行結果：1, 5, 11, 15, 19, 26, 48, 37, 59, 61

第 6 次執行結果：1, 5, 11, 15, 19, 26, 37, 48, 59, 61

第 7 次執行結果：1,5,11,15,19,26,37,48,59,61

RADIX-SORT 說明

多鍵排序就是把數個鍵的紀錄排序起來，這些鍵是 K_1 ， K_2 ，…… K_r (其中 K_1 為第一優先的鍵，而 K_r 是最後的鍵)。一個記錄的串列為 R_1, R_2, \dots, R_n 被稱為根據鍵 K_1, K_2, \dots, K_r 排列結果。而多鍵排序又可分為最高有效數字排序 (MSD)，最低有效數字排序 (LSD)，而這兩個方法只是定義各鍵值的排序不同而已，LSD 和 MSD 排序可以用來排序只有一個鍵的紀錄。對於這點，就是將該鍵分成數個鍵，舉例來說：如果原來鍵是一個

數字，那們鍵中的每一個位數可以視為一個鍵。所以當鍵值在 0 到 999 之間的話，我們就可以利用 LSD 和 MSD 對三個鍵 K1，K2，K3 做排序，其中 K1 是百位數，K2 是十位數，K3 是個位數。而每個鍵的範圍是由 0 到 9，所以每個鍵都可以用十個櫃子的櫃子排序。在基數排序(radix sort)中，我們根據一個基底 RADIX_SIZE 將鍵值分解成數個數字。當 RADIX_SIZE=10 時，我們可以得到上述的十進位分解。在一個基底 RADIX_SIZE 排序中，所需的櫃子數為 RADIX_SIZE 個。假設要排序的紀錄是 R1，....，Rn 而它們的鍵值被一個基底 RADIX_SIZE 所分解，對每個第 MAX_DIGIT 為數的鍵，其範圍從 0 到 RADIX_SIZE-1，所以我們需要 RADIX_SIZE 個櫃子。假設每個記錄包含一個 link 欄位，每個櫃子中的紀錄會被連結起來成一條鏈且存在 front[i]，其中 $0 \leq i < r$ ，是指向每條櫃子中的鏈之第一個記錄的指標，以及 rear[i] 是指向櫃子中最後一個記錄的指標，這些鏈會佇列來處理。而範例的這個動畫是以 LSD 的方式處理。主要程式碼如圖 4-6 圖所示。

```
list_pointer radix_sort(list_pointer ptr)
{ list_pointer front[RADIX_SIZE]
  , rear[RADIX_SIZE];
  int i, j, digit;
  for(i=MAX_DIGIT-1; i>=0; i--){
    for(j=0; j<RADIX_SIZE; j++)
      front[j]=rear[j]=NULL;
    while(ptr){
      digit=ptr->key[i];
      if(!front[digit])
        front[digit]=ptr;
      else
        rear[digit]->link=ptr;
      rear[digit]=ptr;
      ptr=ptr->link;  }
    ptr=NULL;
    for(j=RADIX_SIZE-1; j>=0; j--){
      if(front[j]){
        rear[j]->link=ptr; ptr=front[j];
      }
    }
    return prt;
  }
}
```

圖 4-6 RADIX-SORT 程式碼

例如輸入為：179->208->306->93->859->984->55->9->271->33

第 1 次執行結果：

271->93->33->984->55->306->208->179->859->9

第 2 次執行結果：

306->208->9->33->55->859->271->179->984->93

第 3 次執行結果：

9->33->55->93->179->208->271->306->859->984

SELECTION-SORT 說明

選擇排序簡單的說就是找出剩餘記錄中鍵值最小的紀錄。並將它放在適當的位置上。第一次會找出所有 N 個記錄中鍵值最小的紀錄，並將它與在第 1 個位置的紀錄對調，使鍵值最小的紀錄放在第一個位置。第二次則將剩餘的 N-1 個記錄中，找出鍵值最小的紀錄，並將它與第 2 個位置對調...，依此類推。總之，第 i 次的過程會找出相對於所有紀錄中具第 i 小鍵值的紀錄。這個過程必須重複 N-1 次，當前面的 N-1 個記錄都已在正確的位置時，剩餘的最後一個位置也已正確的位置上，因為此時它的鍵值最大，而它也在第 N 個位置。主要程式碼如圖 4-7 圖所示。

```
void selectionsort (int list[], int last)
{
    int current, smallest, holddata, walker;
    for(current=0; current<last; current++)
    {
        smallest=current;
        for(walker=current+1; walker<=last; walker++)
            if(list[walker]<list[smallest])
                smallest=walker;
        holddata=list[current];
        list[current]=list[smallest];
        list[smallest]=holddata;
    }
}
```

圖 4-7 SELECTION-SORT 程式碼

例如輸入為：23, 78, 45, 8, 32, 56

第 1 次執行結果：8, 78, 45, 23, 32, 56

第 2 次執行結果：8, 23, 45, 78, 32, 56

第 3 次執行結果：8, 23, 32, 78, 45, 56

第 4 次執行結果：8, 23, 32, 45, 78, 56

第 5 次執行結果：8, 23, 32, 45, 56, 78

SHELL-SORT 說明

謝爾排序法的主要精神是首先將整個陣列分成若干個陣列，分別以插入排序法來排序每一個小陣列，再將排序的陣列逐漸變大，而且陣列中的資料也越來越具有某種程度的排序關係，所以排序的速度會得到改善。我們先將陣列 a 中的 N 個元素分別為 dn 個組別(資料量不要太大)，第一組包含 $a[0]$ 、 $a[dn]$ 、 $a[2dn]$，第二組包含 $a[1]$ 、 $a[dn+1]$ 、 $a[2dn+1]$，依此類推。接著單獨對每一組元素進行直接插入排序，完成後再以 $dn-1$ 的組距重新排序($dn-1 < dn$)，使得每一組的資料變多一點，而且組別變少，重覆分組及排序，直到 $d_0=1$ ，使得所有元素屬於同一組為止。主要程式碼如圖 4-8 圖所示。

```
void shellsort(int list[], int last)
{
    int hold, incre, curr, walker;
    incre=last/2;
    while(incre!=0)
    {
        for(curr=incre;curr<=last;curr++)
        {
            hold=list[curr];
            walker=curr-incre;
            while(walker>=0&&hold<list[walker])
            {
                list[walker+incre]=list[walker];
                walker=(walker-incre);
            }
            list[walker+incre]=hold;
        }
        incre=incre/2;
    }
}
```

```
        return;  
    }
```

圖 4-8 SHELL-SORT 程式碼

例如輸入為：77, 62, 14, 9, 30, 21, 80, 25

第 1 次執行結果：30, 21, 14, 9, 77, 62, 80, 25

第 2 次執行結果：14, 9, 30, 21, 77, 25, 80, 62

第 3 次執行結果：9, 14, 21, 25, 30, 62, 77, 80

4.2 教學頁面製作

FLASH 的開發環境：

圖 4-9 為 Flash 的開發介面。

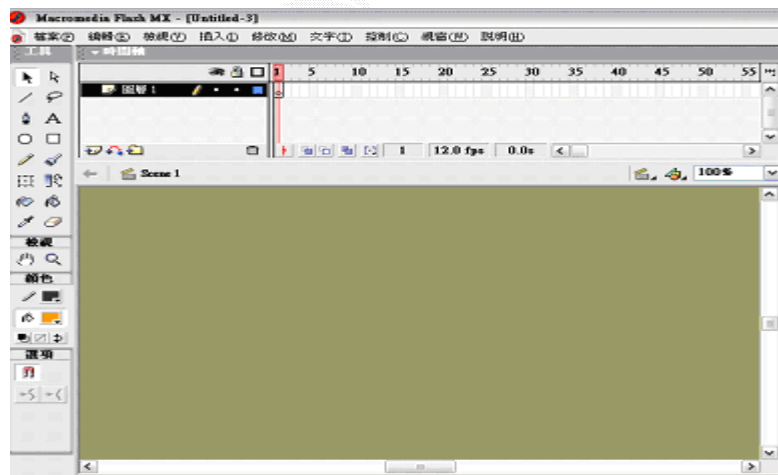


圖 4-9 FLASH 開發環境圖

Flash 按鈕實作：

我們以圖4-10中的play的按鈕為例，首先先我們將按鈕這個元件單獨編輯，而在play的按鈕元件中有分成一般、滑鼠經過、按下以及執行四個影格，以下我們將單獨的介紹這4個影格的作用。

1. 一般正常的情况滑鼠不會去移動到按鈕時，此時按鈕所呈現的狀態，如圖4-10 (a)。

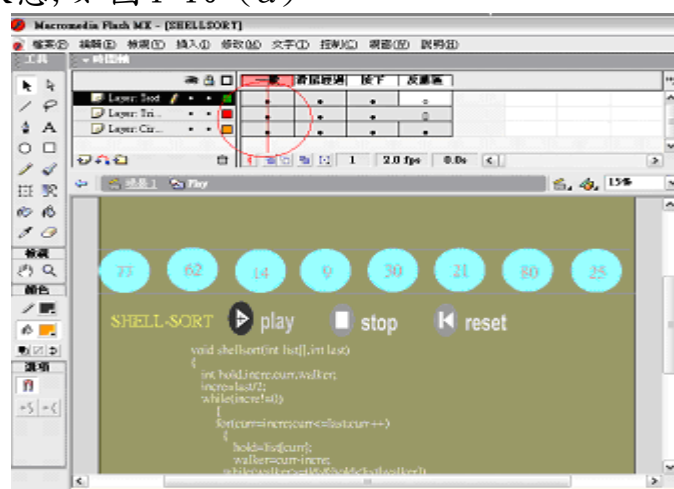


圖4-10 (a) 按鈕實作圖

2. 當滑鼠移動到執行的範圍時候的影格時，此時按鈕呈現出來給使用者看到的狀態，如圖4-10 (b)。

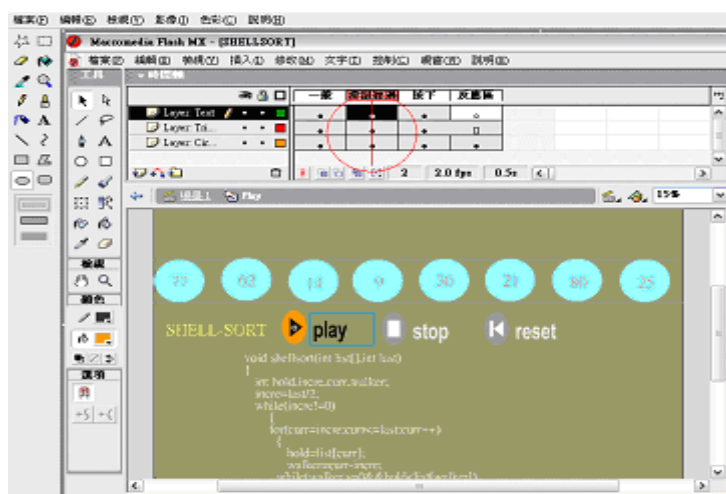


圖4-10 (b) 按鈕實作圖

3. 按下:當滑鼠一直按著執行區域時候的影格,此時按鈕呈現的狀態,如圖4-10(c)。

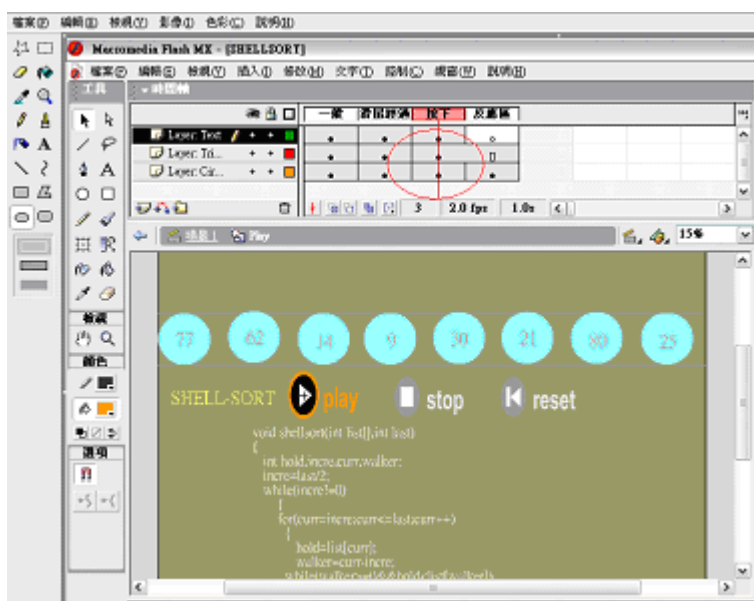


圖4-10 (c) 按鈕實作圖

4. 執行:按鈕所感應到的部分,當滑鼠移動到執行區域時就會跳到執行的影格,如圖4-10(d)。

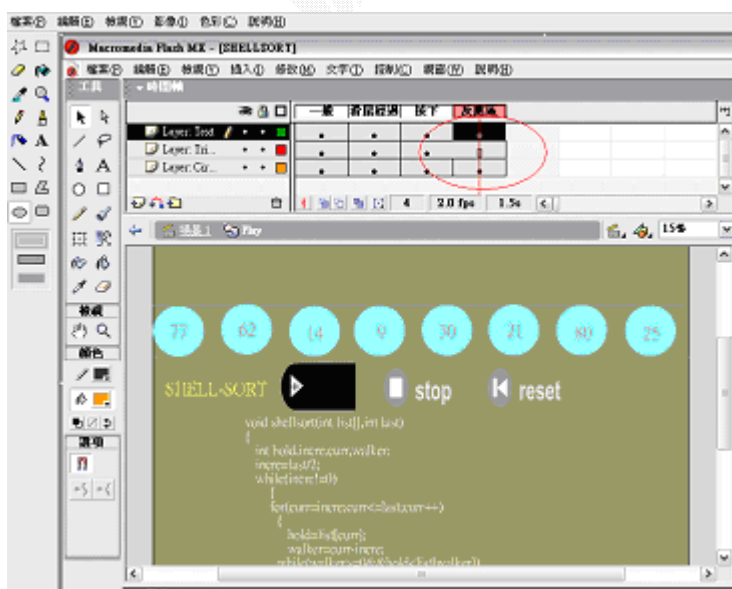


圖4-10 (d) 按鈕實作圖

這樣的按鈕元件編輯完成後,加到FLASH動畫中,可以讓使用者很容易的看到表達的重點,並且方便控制流程。

Flash 的實作部分：

要呈現動畫的部分主要會用到兩個技巧：

1. 移動漸變：此功能主要用在物件的移動、大小旋轉變化。我們在工作區建立一個圖形時，時間軸上的影格會出現一個實心的小黑點，將所要移動的物件放在開始以及結束的影格並且要設定相對的置，然後在開始的影格上按下滑鼠右鍵，選取產生移動漸變，很明顯的時間軸上的箭號出現了。這項功能不僅僅能製作移動的動畫，更可產生變大變小以及旋轉的效果。如圖4-11中紅色圈圈所示。

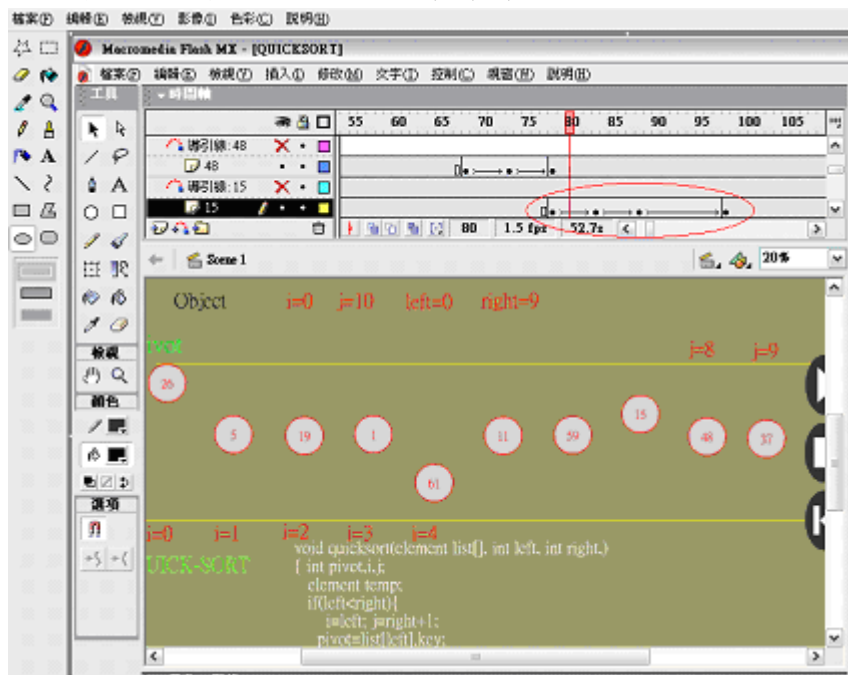


圖4-11 FLASH移動漸變實作圖

2. 引導線：此功能可以引導物件隨著我們規定的路線前進。首先我們先選取要隨規定路線前進物件的所在圖層，接著在圖層上單擊滑鼠右鍵，選取增加引導線，這時候會產生一個導引線的新圖層，這個圖層就是用來決定物件是要隨著哪一條路線前進。接下來在這個新圖層的工作區中插入關鍵影格並且畫出我們希望物件所做的路線，然後將物件的開始位置對齊此引導線的開端，並將物件的結束位置對齊此引導線的末端，如同移動漸變的產生方法，在此物件的開始影格單擊滑鼠右鍵，選取產生移動漸變，即可以產生出一個依引導線軌跡運動的物件動畫。圖4-12中紅色圈圈即為導引線。

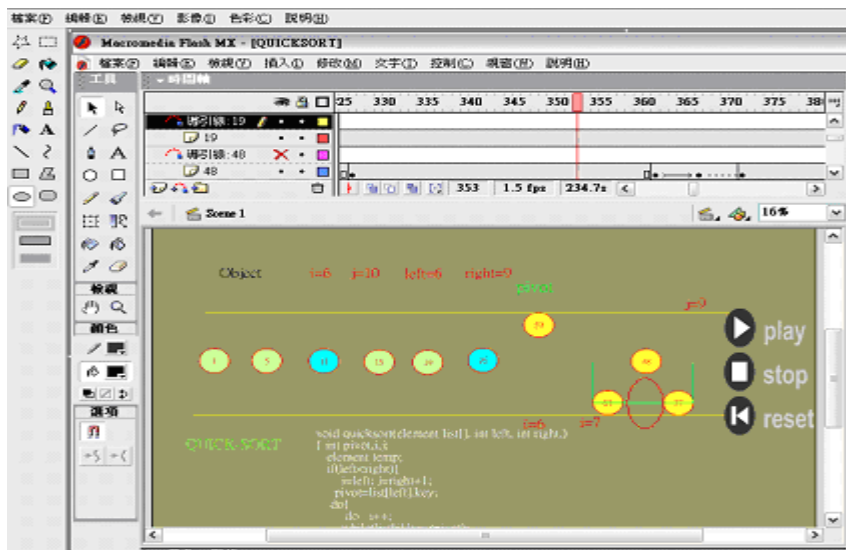


圖4-12 引導線製作圖

我們以Quick-Sort為例，我們可以看到在圖4-13(a)中左邊紅色圈圈的部分，就是我們這個動畫用到的主要部分，裡面主要是以導引線和物件為主，並且加上一些適當的程式碼去控制動畫的流程。

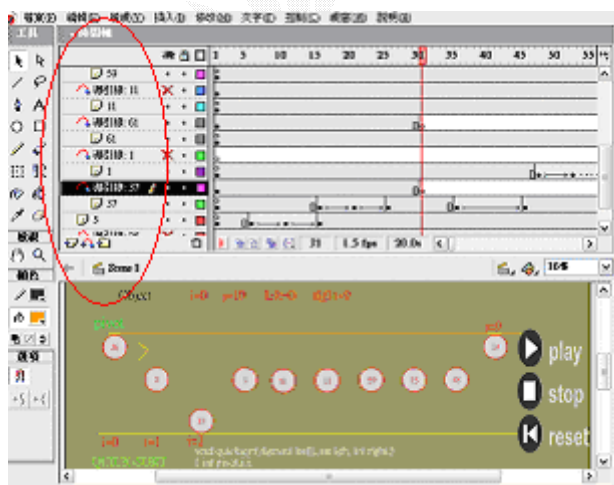


圖4-13(a) Quick-Sort製作

以圖4-13(b)中19這一個元件為例，我們可以藉由產生移動漸變去配合動畫的表現，並且使用導引線去控制此元件所要經過的路線。

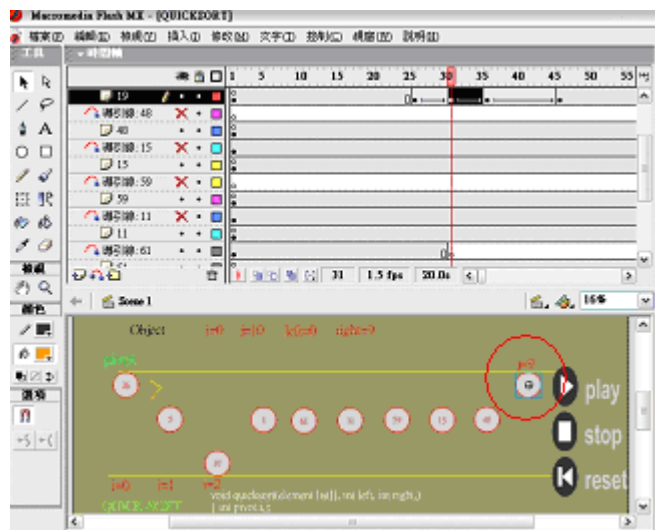


圖4-13(b) Quick-Sort製作

以圖 4-13(c)Quick-Sort 中的程式碼為例，我們要依此程式碼的行為配合動畫目前所呈現的部分，因此我們將程式碼跟動畫作在一起，以便利使用者觀看程式的變化，並且將程式馬目前執行到哪一部分用紅色的文字表現出來，使得使用者更加容易學習。經過 FLASH 動畫配合程式碼的變化表示，相信會讓使用者更加容易觀看跟比較，並且更容易理解動畫的行為。

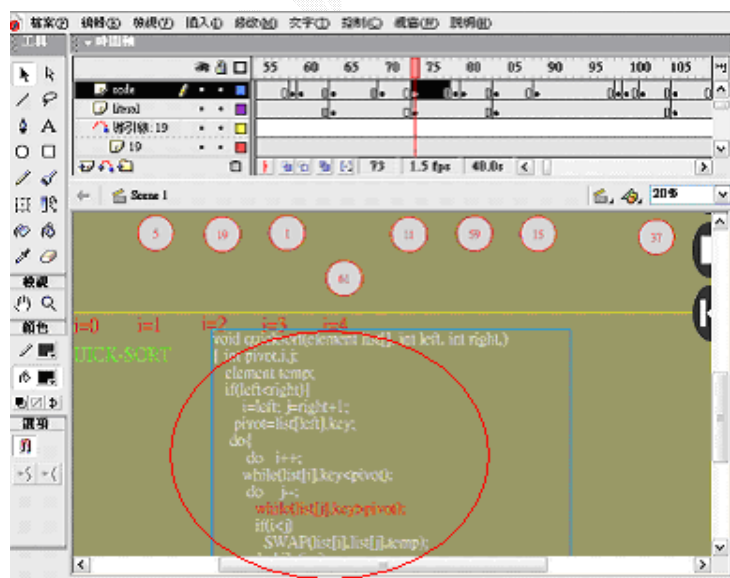


圖4-13(c) Quick-Sort製作

4.3 PHP 網頁製作

此節將說明我們製作 PHP 互動式網頁的方法。

4.3.1 PHP 跟 MYSQL 連結

圖4-14是表示使用者、伺服器、資料庫與PHP之間的關係。網路使用者經由網路，觀看網頁以及存取資料庫。

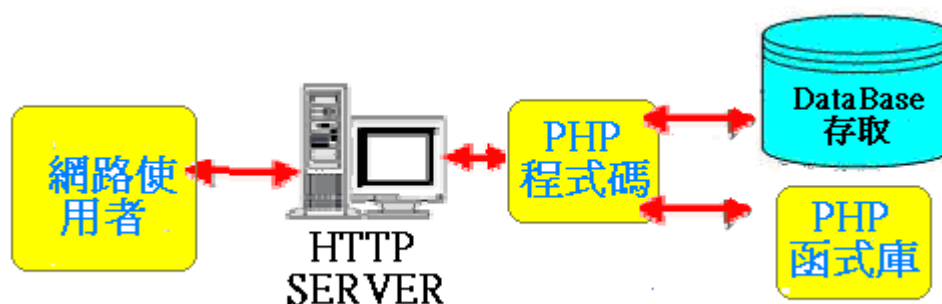


圖 4-14 PHP 與 MYSQL 連結圖

以PHP 連結資料庫作查詢可以分為以下四個階段：

1. 建立與資料庫的連結

我們透過mysql_connect()這個函式函資料庫作連結。

函式的原形

如下：

```
int mysql_connect ([string hostname [:port]
[:/path/to/socket]
[, string username [, string password]]]);
```

當連結成功時，則會傳回一正整數值之link identifier，連結失敗時則傳回錯誤訊息。

2. 選擇要做查詢之資料庫，並輸入SQL指令作查詢後將其結果存至變數中：

我們利用mysql_select_db()與mysql_query()兩個函式對資料庫作查詢。

函式的原形如下：

```
int mysql_select_db(string database_name,  
int[link_identifier]);
```

本函式選擇MySQL 伺服器中的資料庫以供之後的資料查詢作業(query) 處理。成功傳回true，失敗則傳回false。

```
int mysql_query(string query, int [link_identifier]);
```

本函式送出query 字串供MySQL 做相關的處理或者執行。若沒有指定link_identifier 參數，則程式會自動尋找最近開啟的ID。當query 查詢字串是UPDATE、INSERT 及DELETE 時，傳回的可能是true 或者false；查詢的字串是SELECT 則傳回新的ID 值。當傳回false 時，並不是執行成功但無傳回值，而是查詢的字串有錯誤。

3. 取得查詢的結果：

可以利用mysql_fetch_row()、mysql_fetch_array() 及mysql_fetch_object() 等函式取回查詢之結果，不同點在於其傳回值。

以mysql_fetch_row()為例=> array mysql_fetch_row(int result);

用來將查詢結果result 之單列拆到陣列變數中。陣列的索引是數字索引，第一個的索引值是0。若result 沒有資料，則傳回false值。

4. 關閉與資料庫的連結，以及釋放傳回值占用記憶體：可以使用mysql_close()關閉與資料庫的連結，mysql_free_result() 釋放傳回值所佔用的記憶體。

函式的原形如下：

```
int mysql_close(int [link_identifier]);
```

本函式關閉與MySQL 資料庫伺服器的連線。若無指定參數 link_identifier 則會關閉最後的一筆連線。實際上本函式不是一定需要的，當PHP 整頁程式結束後，將會自動關閉與資料庫的連線。成功傳回true、失敗傳回false 值。

```
boolean mysql_free_result(int result);
```

本函式可以釋放目前MySQL 資料庫query 傳回所占用的記憶體。一般只有在非常擔心在記憶體的使用上可能會不足的情形下才會用本函式。PHP 程式會在結束時自動釋放。

4.3.2 登入介面撰寫

我們在此利用HTML所提供的認證檔頭(header)功能，來對使用者作認證的動作。

程式只要輸出兩個檔頭，當瀏覽器讀到這兩個檔頭時，就會出現一個要求使用者輸入帳號與密碼的視窗。

使用者輸入資料後，瀏覽器就會將使用者帳號與密碼傳回給程式，並由程式與資料庫中資料比對是否一致。圖4-15所表現的程式碼為學生的登入考試系統處理程式。

```
<?
include ("config.php");

if ($work=="check_std") {
    if (empty($account) || empty($passwd) || empty($test_no)){
        echo "帳號、密碼輸入不完全，請回上一頁重新輸入";
        exit();
    }

    $str = "select account,passwd from $studenttable where
            account=' $account' ";
            //check student account
    $result = mysql_query($str,$link);
```

```

$total=mysql_num_rows($result);
if ($total!=1){
    echo "帳號錯誤!";
    exit();
}
else {
    $row=mysql_fetch_row($result);
    if ($row[1]!=$passwd){
        echo "密碼錯誤!";
        exit();
    }
}

header("Location:on_line_test.php?tid=$tid&account=$account");
}
?>

```

圖4-15 登入介面程式碼圖

4.3.3 考試介面撰寫

我們在此的動作是先去讀取test_paper這個Table中的題號，因為題號存放在欄位中是以〈題號：題號：…〉存放著，所以我們使用split這個函式將它給分解出來，之後在以跑迴圈的方式，將每一是給印出來。圖4-16所表現的程式碼為學生考試時系統處理程式。

```

<?
include ("config.php");
$str_test="select * from $testtable where tid=$tid";
$result_test = mysql_query($str_test,$link);
$row_test=mysql_fetch_row($result_test);

list($tid,$test_name,$test_no,$test_status,$create_date,$question_to
tal,
$question_total_now,$question_no,$ans_no)=$row_test;

$prob=split(":",$question_no);           //分解的題目

for($i=0;$i<$question_total;$i++)      //迴圈顯示出題目
{
    $str_question="select
question,option1,option2,option3,option4,option5,cid from

```

```

$problemtable where pid=$prob[$i]";
    $result_question = mysql_query($str_question,$link);
    $row_question=mysql_fetch_row($result_question);

list($question,$option1,$option2,$option3,$option4,$option5,$cid)=$r
ow_question;
    if($cid!=0)
    {
        $str_picture=" select pic from $picturetable where cid=' $cid' ";
        $result_picture = mysql_query($str_picture,$link);
        $row_picture=mysql_fetch_row($result_picture);
        list($pic)=$row_picture;
    }
?>
        .
        .
        .
//若此題有使用到圖片時↓
    <? if($cid!=0) { ?> <a href=pictures\<? echo $pic ?> target=blank
>
        <img src=pictures\<? echo $pic ?> width=200 height=150  ><? } ?>
    </td>
</tr>
<tr>
    <td width="5%" align="center"><input type="radio" name="ans<?echo
$i;?>" value="1" ></td>
    <td width="65%"><?echo $option1;?></td>
</tr>
        .
        .
        .
<?
}
?>

```

圖 4-16 考試介面程式碼圖

4.3.4 計算成績撰寫

我們在此的動作跟之前的考試介面撰寫很相像，是先去讀取 test_paper 這個 Table 中的答案，因為答案存放在欄位中也是以< 答案：答案：...>存放著，所以我們使用 split 這個函式將它給分解出來，之後在以跑迴圈的方式，將學生所考的每一題答案做比對，並且記錄記錄各選項選答比率及答對率，計算完成績之後將學生成績及考試的選答記錄存放在 st_ans 的 table 中。圖 4-17 程式碼為學生考試時系統處理程式。

```

<?
include ("config.php");

$str_test="select * from $testtable where tid=$tid";
$result_test = mysql_query($str_test,$link);
$row_test=mysql_fetch_row($result_test);

list($tid,$test_name,$test_no,$test_status,$create_date,$question_to
tal,
    $question_total_now,$question_no,$ans_no)=$row_test;

$prob=split(":",$question_no);          //分解題目
$right_ans=split(":",$ans_no);         //分解答案

$str_std_ans="select * from std_ans where test_no='$test_no' &&
account='$account' ";    //check 是否重考
$result_std_ans = mysql_query($str_std_ans,$link);
$rows = mysql_num_rows($result_std_ans);

if ($rows == 1) {                      // 重覆
考
    echo "重覆交卷！！若有誤按，請找老師處理";
    exit();
}
else {
    if ($work=="process_score" && !empty($PHP_AUTH_USER)) {
        $yes=0;
        $no=0;
        $total_score=0;
        for ($i=0;$i<$question_total;$i++) {
            //記錄選項及答對率
            .
            .
            .
            $total_std++;
            $answer="ans".$i;
            $ans_tok[$i]=$answer;
            if ($ans_tok[$i]==$right_ans[$i]){ //判斷對錯
                $yes++;
                $correct_std++; //答對人數
            }
            else
                $no++;
            switch($ans_tok[$i]){ //每個選項人數
                case 1: $opt1_std++;
                    break;
                .
                .
                .

```

```

    }
//更新考題資料
$query_update="update $problemtable set
    total_std=' $total_std',correct_std=' $correct_std'
        .
        .
        .
    }
    $std_ans=join(":",$ans_tok);
    $total_score=$yes * (100/$question_total);

    $str_std="select * from $studenttable where
account=' $account' ";
    //找出學生基本資料
    $result_std = mysql_query($str_std,$link);
        .
        .
        .
        //記錄成績

    $query_insert="insert into std_ans
(test_no, account, class, name, seat_no, std_no, std_ans, score) values
(' $test_no', ' $account', ' $class', ' $name', ' $seat_no', ' $std_no',
' $std_ans', ' $total_score')";

```

圖 4-17 計算成績程式碼圖

4.3.5 題庫出題撰寫

我們在此的動作是由使用者選擇此題目所選擇的章節開始，之後依序輸入題目、選項、答案，若要上傳圖片，則會判斷圖片的格式，若不是 JPG 或 GIF 檔則無法上傳。圖 4-18 程式碼為老師出題時系統處理程式。

```

<?
include ("config.php");
if ($work=="question_std" && !empty($PHP_AUTH_USER)) {
    if (empty($question)
||empty($option1)||empty($option2)||empty($ans)) {
        echo "題目輸入不完全，請回上一頁重新輸入";
        exit;
    }

    if
(($uploadfile_type=="image/jpeg")||($uploadfile_type=="image/gif"))
    {
        if (copy($uploadfile,"pictures/".$uploadfile_name))
        {

```

```

        echo "上傳成功!";
        $query_insert="insert into $picturetable (pic) values
(' $uploadfile_name' )";
        $result = mysql_query($query_insert,$link) or die ("Invalid
Insert Query");
    }
    else { echo "上傳失敗!"; }
}
else { echo"限制上傳檔案類型!!請將類型改為 JPG 或 GIF 檔!!"; }
unlink($uploadfile);

$str = "select cid from $picturetable where pic=' $uploadfile_name' ";
.
.
.
?>
<p align="center"><b><font color="#000080" size="4">新增題目
</font></b></p>

<form method="POST" action="question_add.php"
enctype="multipart/form-data" >
    <input type="hidden" name='work' value='question_std'>
.
.
.
//選擇章節
    <td width="452" align="left" colspan="3">
    <select name="chapter">
    <?
        include ("config.php");
        $str = "select chapter from $sectiontable";
        $result = mysql_query($str,$link);
        while ($row = mysql_fetch_row($result)) {
            echo "<option value=\" $row[0]\">$row[0]</option>";
        }
        mysql_close ($link);
    ?>

```

圖 4-18 題庫出題程式碼圖

出題介面：圖 4-19 是進入系統後的出題頁面。

新增題目

| | | |
|-----------------------|----------------------|--------------------------------------|
| 單元 | Bubble_Sort | |
| 題目 | <input type="text"/> | |
| 選項 1 | <input type="text"/> | |
| 選項 2 | <input type="text"/> | |
| 選項 3 | <input type="text"/> | |
| 選項 4 | <input type="text"/> | |
| 選項 5 | <input type="text"/> | |
| 答案 | <input type="text"/> | 難易度選擇 <input type="text" value="難"/> |
| 上傳說明圖片 (若不需要，請輸入0) | <input type="text"/> | <input type="button" value="瀏覽..."/> |
| 說明 | <input type="text"/> | |

圖 4-19 出題介面圖

4.3.6 考卷出題撰寫

我們在此的動作是由老師選擇要從題庫自行選擇題目來完成出題，或由亂數產出題目來完成。圖 4-20 的程式碼為老師出題時系統處理程式。

由題庫自行選擇題目來完成出題：

主要是讀取 problem_base 這個 table，然後將每題給顯示出來，若將題目選入考卷時，則將題號及答案給加入考卷中，此部份最大的問題是，必須去將之前所選入的題目分解出來，之後再使用 join 這函式將它再次結合其來，才會在欄位成為 < 題號：題號：... > 的樣式。而圖 4-21 所表示的是由題庫自行選擇題目的介面。

```

<?
include ("config.php");
$str = "select question_total, question_total_now, question_no, ans_no
      from $testtable where tid=$tid";
$result = mysql_query($str, $link);
$row = mysql_fetch_row($result);
list($question_total, $question_total_now, $question_no, $ans_no)=$row;
$tok=split(":", $question_no);           //分解之前的題號
if ($work=="choose_question" && !empty($PHP_AUTH_USER)) {
    //題目加入試卷
    if ($question_total==$question_total_now)
    {
        echo "題目已足夠，若要加入題目請先刪除";
        exit;
    }

    $question_total_now++;                //選一題
    $tok[$question_total_now-1]=$pid;     //加入題號
    $question_no=join(":", $tok);

    if ($ans_no=="") $ans_no=$ans; else $ans_no=$ans_no." : ". $ans;
    //加入每題之答案
    .
    .
    .
}
}

if ($work=="delete_question" && !empty($PHP_AUTH_USER)) {
    //刪除已加入的題目

    if ($pid_location==$question_total_now-1)           //最後一個
        unset($tok[$pid_location]);
    else {
        for ($i=$pid_location; $i<$question_total_now-1; $i++) //題目往前
搬
            $tok[$i]=$tok[$i+1];
        unset($tok[$i]);
    }
    $question_no=join(":", $tok);

    $ans_tok=split(":", $ans_no);
    if ($pid_location==$question_total_now-1) //最後一個
        unset($ans_tok[$pid_location]);
    else {
        for ($i=$pid_location; $i<$question_total_now-1; $i++)
            //答案往前搬
            $ans_tok[$i]=$ans_tok[$i+1];
    }
}
}

```

```

unset($ans_tok[$i]);
}
$ans_no=join(":", $ans_tok);

$question_total_now--;
.
.
.
?>
    
```

圖 4-20 考卷出題程式碼圖

由題庫出考題

選擇題目章節、單元： 預定題數 2 已選題數 2 亂數出題 回考卷瀏覽

| 編號 | 單元 | 難易度 | 題目 | 答案選項 | 選答比例 | 答案 | 說明 | 圖片 | 選入考卷 |
|----|-------------|-----|---------------------------------------|---|----------------------|----|--|-------|------|
| 1 | Bubble_Sort | 中 | [[19、23、2、4、99、1、8]經由氣泡排序法之後的PASS3為何? | 2、4、1、8、23、19、99 2、4、1、8、19、23、99 2、4、8、1、19、23、99 2、4、19、1、8、23、99 | 0% 0% 0% 0% | 2 | PASS1:19、2、4、23、1、8、99 PASS2:2、4、19、1、8、23、99 PASS3:2、4、1、8、19、23、99 | 無使用圖片 | 選入考卷 |
| 2 | Bubble_Sort | 易 | 討論BUBBLE SORT的average case的時間複雜度? | 0 (n ²) 0 (n log n) 0 (n) 0 (log n) 0 (log n ²) | 0% 0% 0% 0% | 1 | | 無使用圖片 | 選入考卷 |
| 3 | Bubble_Sort | 中 | BubbleSort在worst case中需要比較幾次才能完成排序呢? | n-1 n(n-1)/2 n ² n ² /2 n(n-1) | 0% 0% 0% 0% | 2 | 需比較 0+1+2+...+n-1=n(n-1)/2 | 無使用圖片 | 選入考卷 |

圖 4-21 考卷出題介面圖

由亂數產出題目來完成：

此部份的動作主要是由老師選擇要亂數出題的章節，然後去讀取符合的條件，若符合時將它讀到陣列中，之後使用 srand 函數去下數種子，再使用 array rand 函數來將數列給亂數讀出。此部份較大的困難，是要判斷之前是否有老師自行選擇，卻為完成考卷，若有的話，則要將取出的亂數去做比較。圖 4-22 是亂數出題的程式碼，圖 4-23 是亂數出題的介面。

```

if((!empty($p_ram_choose_no))&&($p_ram_choose_no<=($question_total-$question_total_now)))
{
    $str3 = "select pid from $problemtable";
    $result3 = mysql_query($str3,$link);
    $row3=mysql_num_rows($result3);
}
    
```

```

$i=0;
while ($row3 = mysql_fetch_row($result3))
{
list($pid)=$row3;
$total_pid[$pid]=$i;
$i++;
}
srand ((double) microtime() * 10000000);
$pid_ram3 = array_rand ($total_pid,$p_ram_choose_no);
$i=0;
while($i < $p_ram_choose_no)
{
if($p_ram_choose_no==1)
{
$str4 = "select ans from $problemtable where id=' $pid_ram3' ";
$result4 = mysql_query($str4,$link);
$row4 = mysql_fetch_row($result4);
list($ans)=$row4;
}
else
{
$str4 = "select ans from $problemtable where
pid=' $pid_ram3[$i]' ";
$result4 = mysql_query($str4,$link);
$row4 = mysql_fetch_row($result4);
list($ans)=$row4;
}
if ($ans_no=="") $ans_no=$ans;
else $ans_no=$ans_no." : ". $ans;
$i++;
}
.
.
.
if($p_ram_choose_no==1) $question_no=$pid_ram3;
else $question_no=join(":",$pid_ram3); //加入題號
}
else
{
$tok=split(":",$question_no);
$tok=array_merge_recursive($tok,$pid_ram3);
$question_no=join(":",$tok); //加入題號
}
.
.
.

```

圖 4-22 亂數出題程式碼圖

隨機出題

請選擇章節 顯示所有題目 總題數2 已出題數1 回考卷瀏覽

| |
|--|
| 所選擇章節總題數 11 |
| 請填入題數(整數) <input style="width: 80%;" type="text"/> |
| 新增題目 清除重寫 |

圖 4-23 亂數出題介面

4.3.7 考卷開啟、關閉程式撰寫

這兒的用處是要讓老師能管理考卷，若考卷沒有開啟的話，則學生無法進入該考卷參加考卷，若再次將考卷關閉的話，則會計算參加該考試學生的排名，及計算考卷內考題難易度更新，這是讓題目的難易度，能以學生的程度來加以判斷。

做法：

1. 計算排名：圖 4-24 所表達的程式碼是使用 `arsort` 這個函式，來將所讀出的資料做排序。

```
//更新排名
$str="select std_no,score from std_ans where test_no='$test_no'";
$result=mysql_query($str,$link);
$num=mysql_num_rows($result);

if ($num>=1) {
    while ($row = mysql_fetch_row($result)){
        list($std_no,$score)=$row;
        $all_score[$std_no]=$score;
    }

    arsort($all_score); //排序，大->小，index 隨之更動
    $rank=1;

    while(list($std_no,$score)=each($all_score)){ //each 傳回陣列索引
及資料
        $query_update="update std_ans set rank_class='$rank' where
test_no='$test_no' && std_no='$std_no'";
        $result = mysql_query($query_update,$link) or die ("Invalid Update
```

```
Rank Query");
    $rank++;
```

圖 4-24 更新學生排名程式碼圖

2. 難易度更新：圖 4-25 的程式碼是去讀取資料庫中，作答總人數和答對的總人數相除之後，乘以 100，若範圍在 0~30，則判斷為難，31~70，判斷為中，71~100，判斷為易。

```
$sstr = "select pid,total_std,correct_std from $problemtable";
$result = mysql_query($sstr,$link);
$n=0;
while ($row = mysql_fetch_row($result)) {
    list($pid,$total_std,$correct_std)=$row;
    if ($total_std > 0){
        $id[$n]=$pid;
        $easy_level[$n]=intval($correct_std/$total_std*100);
        if( ($easy_level[$n]>-1)&&($easy_level[$n]<31) )
    { $easy_level[$n]=難; }
        if( ($easy_level[$n]>30)&&($easy_level[$n]<71) )
    { $easy_level[$n]=中; }
        if(( $easy_level[$n]>70)&&($easy_level[$n]<101) )
    { $easy_level[$n]=易; }
        $n++;
    }
}
for ($i=0;$i<$n;$i++) {
    $query_update="update $problemtable set easy_level='$easy_level[$i]'
where pid=$id[$i]";
    $result = mysql_query($query_update,$link) or die ("Invalid Update
Easy_Level Query");
}
```

圖 4-25 更新難易度程式碼圖

3. 圖 4-26 所表示的是老師要將考卷開啟或關閉的介面。

考卷開啟、關閉

| 編號 | 建立日期 | 考卷名稱 | 考卷代號 | 考題數量 | 已出題數 | 考題編號 | 考卷狀態 | 開啟考卷 | 關閉考卷 |
|----|------------|-------|------|------|------|------|------|------|------|
| 1 | 2007-10-27 | test2 | 123 | 2 | 1 | | | 開啟 | 關閉 |

圖 4-26 考卷開啟、關閉介面

4.4 資料庫規劃、設計

4.4.1 資料庫 E-R 圖

圖 4-27 為本系統資料庫的 E-R 圖，此圖展示了整個系統裡所有資料表間的關係。本系統供單一個老師使用，以避免多個老師之間互相修改考卷或試題，因此一個老師能修改題庫裡面的很多題目，一個老師能管理討論版裡許多文章，一個老師能觀看很多學生的成績，一個老師能出很多份考卷。而學生們都可以在討論版上留言，各則文章也可能有很多學生回覆。而一個學生能查詢許多份成績。N 個學生能參加許多不同的考試。

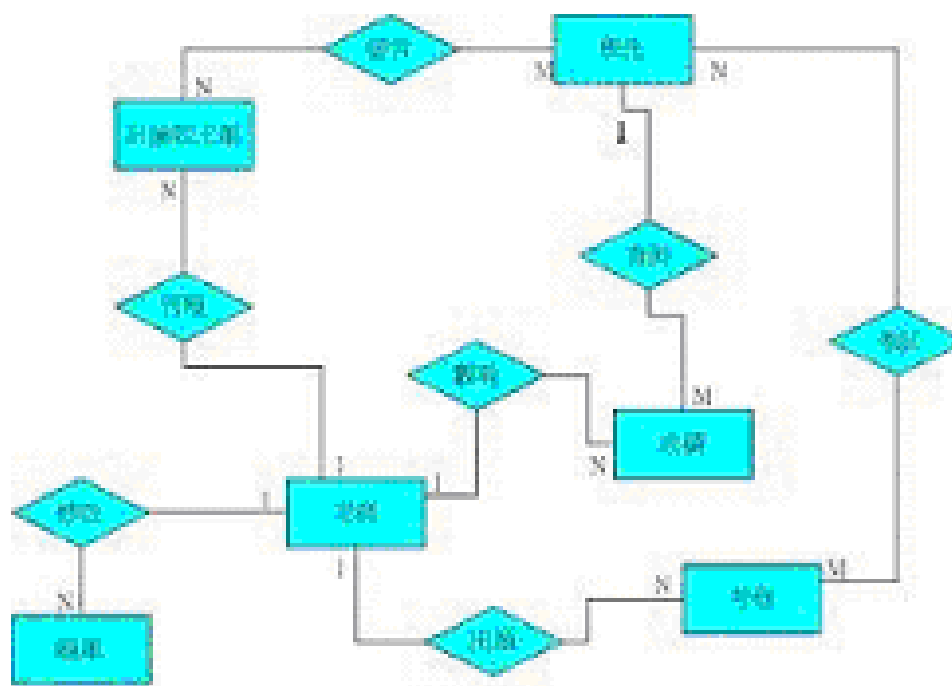


圖 4-27 資料庫 E-R 圖

4.4.2 資料庫總覽

圖 4-28 說明了整個資料庫資料表的總覽，整個資料庫共有九個 table，存放所有我們需要的資料，manager table 存放老師的資料，problem_base table 存放所有題目的資料，test_paper table 存放考卷的資料，std_ans table 存放學生

考試成績，student table 存放學生資料，picture table 存放題目所需圖片，section table 存放題目章節資料，discuss table 存放討論版主題資料，re_discuss table 存放回覆文章資料。每個資料表詳細的說明會在下面的章節說明。

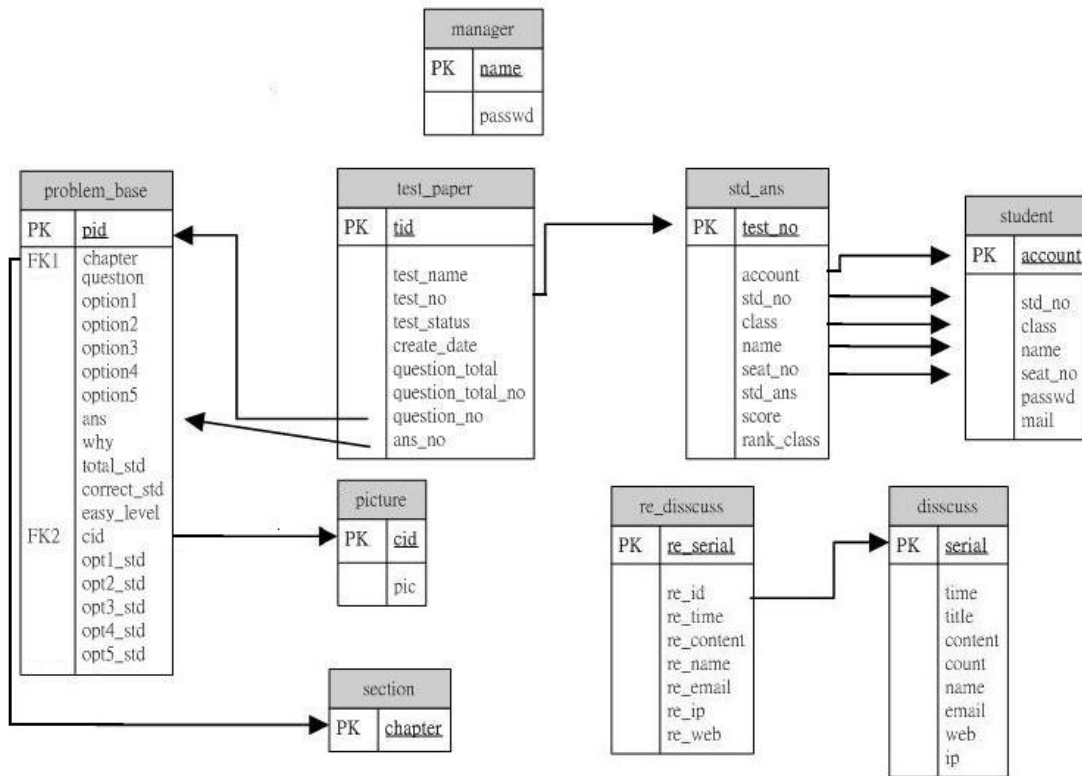


圖 4-28 資料表總覽圖

4.4.3 資料表詳細說明

此章節將會詳細介紹所有資料表的資料欄位，型態，說明。

表4-1為student table 主要存放學生的基本資料，直接想到所需要紀錄的就是學生的姓名，學生的座號，學生的學號，學生所屬的班級，為了方便學生使用系統，我們便學習一般互動式網站使用帳號，密碼的觀念，往後只需輸入兩次資料即可認證，所以我們又增加了 account 以及 passwd 兩個欄位，倘若學生忘記密碼，則可利用學號欄位(std_no)，帳號欄位(account)，姓名欄位(name)，E-mai(mail)欄位來做身份確認，以獲得密碼。

圖4-29表示與學生資料表有關聯性的，就是學生答案資料表，因為每次的測驗成績，也都需要紀錄是哪個學生所做的測驗。

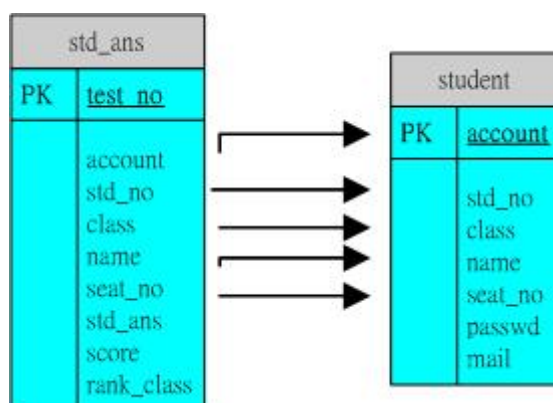


圖4-29 學生資料表圖

表4-1 學生資料表欄位

| 欄位 | 型態 | 說明 | 備註 |
|---------|-------------|------|----|
| account | varchar(12) | 帳號 | 主鍵 |
| std_no | varchar(12) | 學號 | 主鍵 |
| class | varchar(12) | 班級 | |
| name | varchar(12) | 姓名 | |
| seat_no | tinyint(4) | 座號 | |
| passwd | varchar(12) | 密碼 | |
| mail | varchar(30) | 電子郵件 | |

表 4-2 是 problem_base table 存放著所有題目的資料，pid 欄位為主鍵，其值會自動增加，一個題目所需紀錄的資料，題目的敘述(question)，題目的選項(option1, option2, option3, optipn4, option5)，題目的答案(ans)，題目所屬的章節(chapter)，而 pid 之所以作為主鍵，為的是方便我們能馬上找尋到所需要的題目，題目解答說明(why)可以讓答錯的學生，能夠了解做錯的原因，而其他的欄位(total_std, correct_std, easy_level, opt1_std, opt2_std, opt3_std, opt4_std, opt_std)，主要讓我們能分析學生作答的情況，老師可由這些資訊了解學生情形，進而協助學生釐清觀念，圖片編號(cid)，存放題目所需圖形。

圖 4-30 是表示與題庫資料表有關聯性的就是考卷資料表，考卷由許多題目所組成，因此我們需要知道此份考卷究竟有哪些題目，也須知道此題目的答案。

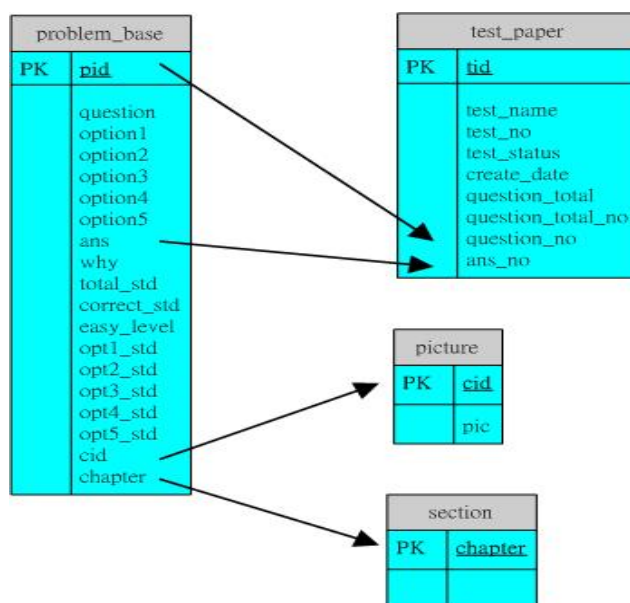


圖 4-30 題庫資料表圖

表 4-2 題庫資料表欄位

| TABLE : problem_base (題庫 TABLE) | | | |
|---------------------------------|-------------|---------|-------------------|
| 欄位 | 型態 | 說明 | 備註 |
| pid | int(10) | 編號 | AutoIncrement, 主鍵 |
| chapter | varchar(32) | 單元 | |
| question | text | 題目 | |
| option1 | text | 選項 1 | |
| option2 | text | 選項 2 | |
| option3 | text | 選項 3 | |
| option4 | text | 選項 4 | |
| option5 | text | 選項 5 | |
| ans | tinyint(4) | 答案 | 預設值 '0' |
| why | text | 說明 | |
| total_std | int(11) | 總作答人數 | |
| correct_std | int(11) | 答對人數 | |
| easy_level | varchar(10) | 難易度 | |
| cid | int(10) | 圖片編號 | 預設值 '0' |
| opt1_std | int(11) | 選答 1 人數 | |
| opt2_std | int(11) | 選答 2 人數 | |

| | | | |
|----------|---------|---------|--|
| opt3_std | int(11) | 選答 3 人數 | |
| opt4_std | int(11) | 選答 4 人數 | |
| opt5_std | int(11) | 選答 5 人數 | |

表 4-3 是表示 test_paper 資料表存放著考卷的資料，需要紀錄的有，此份考卷的名稱(test_name)，考卷代號(test_no)，考卷開啟的狀態(test_status)，此份考卷建立的日期(create_date)，其他的欄位(question_total，question_total_now，question_no，ans_no)，則紀錄著此份考卷裡的題目資訊。

由圖 4-31 可了解考卷資料表除了跟題庫資料表有關連之外，也跟學生答案資料表有關連，因為我們也須知道學生是參加了哪份考卷的考試。

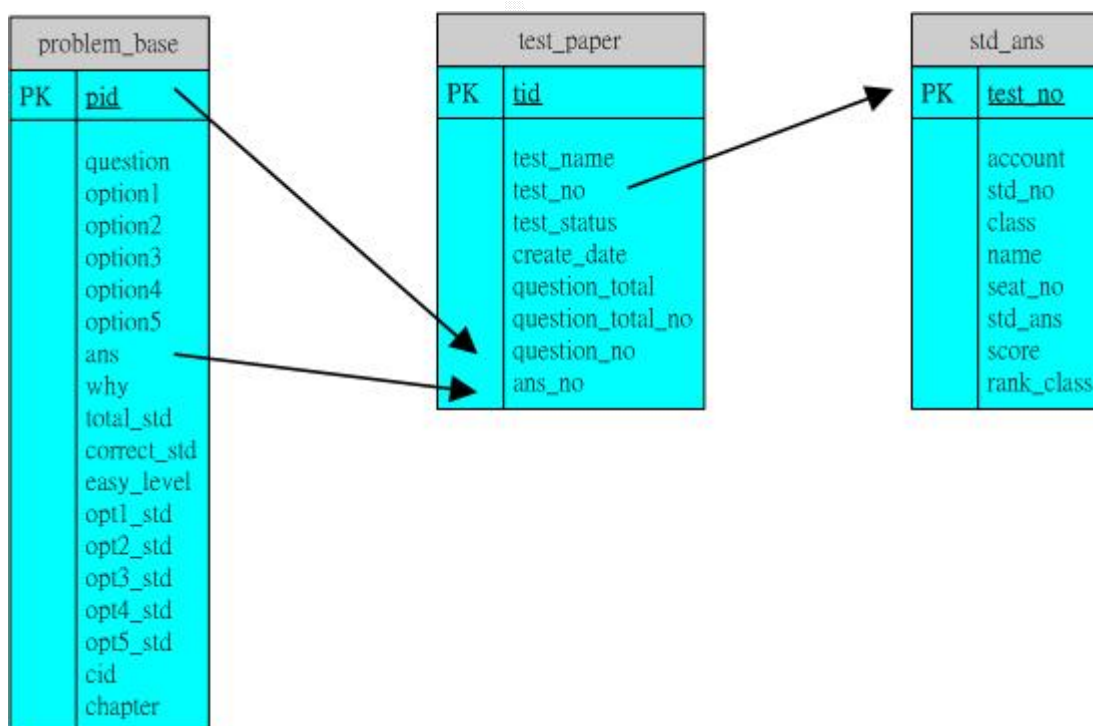


圖 4-31 考卷資料表圖

表 4-3 考卷資料表欄位

| TABLE : test_paper (考卷資料 TABLE) | | | |
|---------------------------------|-------------|--------|-------------------|
| 欄位 | 型態 | 說明 | 備註 |
| tid | int(10) | 考卷編號 | AutoIncrement, 主鍵 |
| test_name | varchar(64) | 試卷名稱 | |
| test_no | varchar(64) | 試卷代號 | |
| test_status | tinyint(4) | 試卷狀態 | 預設值 '0' |
| create_date | date | 日期 | 預設值 '0000-00-00' |
| question_total | int(11) | 題數 | 預設值 '0' |
| question_total_now | int(11) | 已選的題數 | |
| question_no | text | 所選題目題號 | |
| ans_no | text | 所選題目答案 | |

表 4-4 表示 section 資料表存放著章節名稱，chapter 紀錄了章節的名稱。

圖 4-32 可知題目章節與題目資料表存在著關連，由此我們可以知道題庫資料表內的題目是屬於哪個章節。

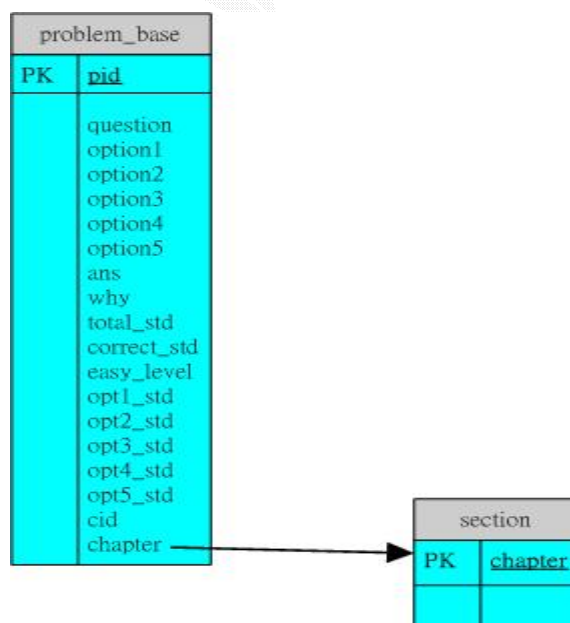


圖 4-32 題目章節資料表圖

表 4-4 題目章節資料表欄位

| TABLE : section (題目章節 TABLE) | | | |
|------------------------------|-------------|------|----|
| 欄位 | 型態 | 說明 | 備註 |
| chapter | varchar(32) | 題目章節 | 主鍵 |

表 4-5 是 manager 資料表存放著老師的帳號密碼資料，以便使用管理權限時的確認，管理者名稱(name)紀錄老師的名稱，passwd 紀錄老師的密碼。

由圖 4-33 可知道管理者資料表獨立於資料庫內，並無跟其它 table 有所關聯。

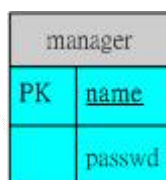


圖 4-33 管理者資料表圖

表 4-5 管理者資料表欄位

| TABLE : manager (管理者 TABLE) | | | |
|-----------------------------|-------------|-------|-------------------|
| 欄位 | 型態 | 說明 | 備註 |
| name | varchar(32) | 管理者名稱 | AutoIncrement, 主鍵 |
| passwd | varchar(32) | 密碼 | |

表 4-6 是 std_ans 資料表主要紀錄著學生參加考試的資料，因為學生參加每次考試的資訊也都需要記錄起來，test_no 欄位可得知學生所參加的是哪次考試。

圖 4-34 表示學生答案資料表與考卷資料表有關連，而 (acct, class, name, seat_no, std_no) 紀錄所參加學生的資料，所以也與學生資料表存在著關連性。學生此次考試的成績(score)，班級的排名(rank_class)。

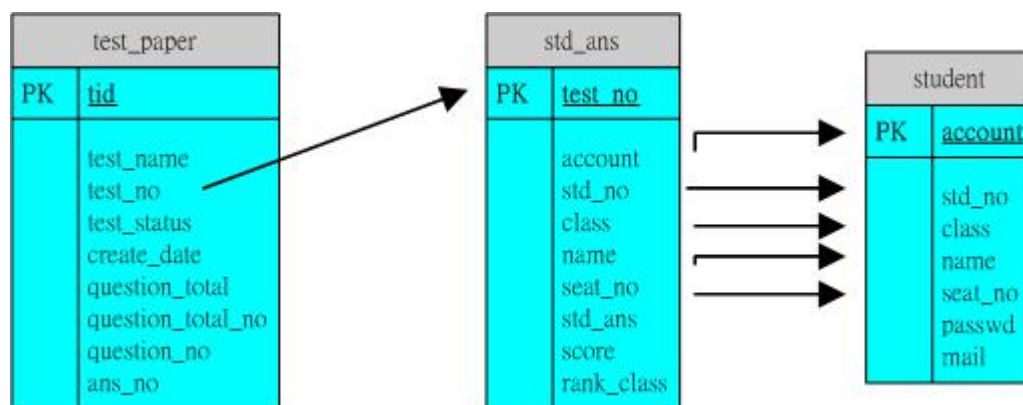


圖 4-34 學生答案資料表圖

表 4-6 學生答案資料表欄位

| TABLE : std_ans(學生答案資料 TABLE) | | | |
|-------------------------------|-------------|------|----|
| 欄位 | 型態 | 說明 | 備註 |
| test_no | varchar(64) | 考卷代號 | 主鍵 |
| accont | varchar(12) | 帳號 | |
| class | varchar(12) | 班級 | |
| name | varchar(12) | 姓名 | |
| seat_no | tinyint | 座號 | |
| std_no | varchar(12) | 學號 | |
| std_ans | text | 學生答案 | |
| score | tinyint | 總分 | |
| rank_class | integer | 班級排名 | |

表 4-7 是 discuss 資料表存放著討論版文章的資料，文章的編號(serial)為主鍵，文章的內容(content)存放內容，文章的標題(title)記錄此文章的標題，其他欄位(name, email, web, ip)，紀錄發文者的資訊，而 count 可以存放此篇文章被點閱的次數。

表 4-7 討論版資料表欄位

| TABLE : discuss(討論版文章 TABLE) | | | |
|------------------------------|--------------|------|-----------------------|
| 欄位 | 型態 | 說明 | 備註 |
| serial | Mediumint(8) | 文章編號 | 主鍵， auto_increment |
| time | datetime | 發表時間 | |

| | | | |
|---------|--------------|-----------|--|
| title | varchar(255) | 標題 | |
| content | text | 內容 | |
| count | int(10) | 人氣值 | |
| name | varchar(50) | 發文者姓名 | |
| email | varchar(50) | E-MAIL 地址 | |
| web | varchar(100) | 個人網站 | |
| ip | varchar(15) | 發文者 ip 位址 | |

表 4-8 顯示了 re_disscuss 資料表存放回覆文章的資料，欄位使用大致都與 disscuss 資料表相同，兩個資料表的關連為 (re_id) 這個欄位，要得知回覆文章屬於那個主題文章，就是使用此欄位來搜尋，所以我們將此鍵設定為索引鍵。

表 4-8 回覆文章資料表欄位

| TABLE : re_disscuss(回覆文章 TABLE) | | | |
|---------------------------------|--------------|------------|-----------------------|
| 欄位 | 型態 | 說明 | 備註 |
| re_serial | int(10) | 回覆文編號 | 主鍵， auto_increment |
| re_id | int(10) | 文章編號 | 索引鍵，預設 值' 0' |
| re_time | datetime | 回覆時間 | |
| re_content | text | 回覆內容 | |
| re_name | varchar(50) | 回覆者姓名 | 預設值' 訪客' |
| re_email | varchar(50) | 回覆者 E-MAIL | |
| re_ip | varchar(15) | 回覆者 ip 位址 | |
| re_web | varchar(100) | 回覆者網站 | |

圖 4-35 表示了討論版文章資料表與回覆文章資料表之間的關連，當文章發表後，若有人對此文章進行回覆，則我們在 re_disscuss 裡的 re_id 欄位放置原本文章的編號，藉此我們就能搜尋到此篇回覆文章究竟是回覆了哪篇文章。

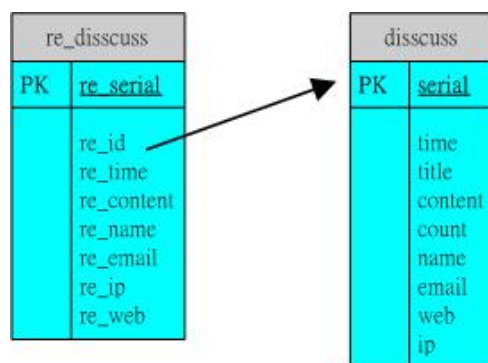


圖 4-35 討論版文章資料表圖

4.4.4 資料庫建置

我們建造資料庫的方法是使用 phpMyAdmin 這套軟體，圖 4-36 左邊顯示了我們的資料庫 exam 裡面所有的 table，可以直接點選資料表的名稱，來進行操作。

而圖 4-36 正中央顯示我們可以對每個資料表執行的操作，可以瀏覽資料表裡的資料，可以觀看結構，直接執行 SQL 語法，也可以匯出資料。

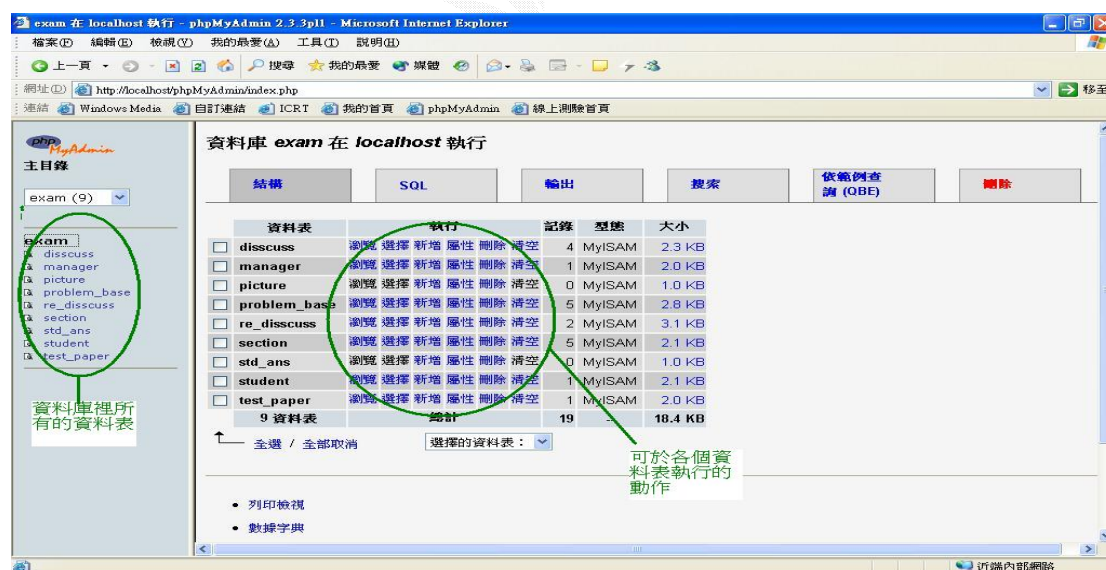


圖 4-36 phpMyAdmin 介面圖

第五章 系統評估與心得

5.1 系統評估

(1) 教學系統評估

本系統主要針對資料結構中的排序章節為教學範圍，使用 Flash 設計、實作。設時，以生動活潑平易近人的方式來呈現，使畫面不至於呆版，讓使用者增加學習的興趣。文字說明方面以書本教學內容，再加上指導老師的解說，盡量簡潔流暢，而此部份的缺點是互動性較為不足，在動畫教學部份無法讓使用者自行輸入所要排序的數列，我們是直接將它設定為固定的型態，使得在教學方面缺少了一些互動性。

(2) 考試系統評估

本系統主要為了評估學生對課程內容及學習情形而製作，我們盡量蒐集各類型的題目，來讓學生避免同一類型題目練習過多。在考試防護方面，我們使用禁止重複考試的方式，來避免學生考試成績缺乏真實性。另外我們也有題目答案說明的設計，使學生瞭解自己出錯的地方，也能知道出錯的原因，此部份的缺點是防護考試不公平的功能不夠強大，我們只有做了禁止重覆考試的功能，例如禁止重新整理或考試時間限制…等功能，因時間不夠未能完成，這是這部份較令人遺憾的事情。

(3) 管理系統評估

我們的管理系統主要供老師使用，在出題方面老師可以選擇題目所屬的章節及難易度，並且可以上傳圖片來讓參加考試的學生能有多元化的考試。在考卷方面，老師可以選擇由題庫自行選擇題目或由系統亂數出題，來完成考卷出題。介面設計，盡量以平易近人及便利性為主，來減少老師在此方面的麻煩，而查詢密碼算是我們比較不足的地方，因為我們沒有使用 Mail-Sever，所以當學生忘記了當時所輸入的密碼時，系統無法直接寄信給此位學生，只能使用線上的查詢系統。

(4) 資料庫系統的評估

本系統是以MySQL與PHP來製作開發的，但是有的動作需要使用PHP加以控制MySQL才能得到結果，因此在程式的撰寫方面需花較多的時間，且PHP的Sever環境必須要有Apache才能使用，所以對於開發環境有一定的要求在。但是在資料搜尋方面，MySQL是使用HASH的方式，因此在搜尋上節省了許多時間。

5.2 未來發展

如同一開始所說的：我們希望提供不受時空限制的學習方式，不再侷限於固定的時間固定的地點才能學習，提供便利化的學習環境及課後複習的最佳管道。我們的範圍絕對不只限於資料結構，大型企業、學校以及補習班利用網路來提供訓練或教學已經不在少數，例如學校與資訊產業合作投資教育網站、逢甲大學的遠距教學、大型補習所謂的數位學堂，這些例子足以顯現無論是學校教育或是社會教育，「網路教學」必將成為未來教學的一股洪流。

很高興我們踏出了第一步，能讓我們做的教學系統提供給學弟妹使用；然而網路教學的最大一個問題：就是學習者如果有疑問或不了解的無法立即的得到解答，我們目前只能在教學內容上盡量做到完整清楚，不要造成學員的困擾及疑問；未來我們希望可以發展出更完善、更便利的教學系統，也希望可以製作出其他學科的教學內容來提供給學弟妹使用，讓大家在學習的道路上走的更輕鬆！

5.3 專題製作心得

經過這次製作專題的過程，使我們學習了許多的事情，從組員的選擇，找尋專題老師，找尋題目，確定題目，設定規劃我們的系統，訂定專題開發的時程表，找尋資料，反覆的討論，老師的意見，到最後終於完成了整個系統。

一開始我們懷抱著許多夢想，想出了很多點子，接著便開始製作，可是直到真正去做之後，我們才突然發現並沒有想像中的容易，光是架設主機我們就花了許多時間，而資料庫的設計規劃更是頭疼，後來網頁更做了許多次的重大改變，終於達到指導教授的要求。很感謝專題老師以及所有專題組員，我們

終於做到了。

黃柏睿

這一次的專題我是負責FLASH的部分，在這之前我都沒有接觸過這一類做動畫的軟體。

一開始FLASH搞的我暈頭轉向，不太知道要從何下手，於是就猛K書，起初剛開始做的時候，讓我感覺到挫折感很大，因為一開始根本都搞不清楚要怎麼弄，之後又去請教同學，最後才逐漸會用FLASH，而我們在教學部分上，我們把白紙黑字枯燥的程式碼轉換成以動畫的方式表現出來，在跟組員討論後覺得將動畫和程式碼做在一起，讓使用者在一邊看動畫一邊可以依程式碼的動作來看到相對應的動作，讓使用者能更清楚的了解程式的行為，而且這一次的專題也讓我深深感受到互助合作的重要性。

當你沒有點子的時候，在一旁的組員卻會適時的給你建議，說不定就讓你有恍然大悟的感覺。所以專題真的是一種大家互助合作的大工程。

楊文嘉

在這次專題中，我跟我的組員負責撰寫資料庫跟PHP方面的功能，且該部分也做出了該有的基本功能，也因為在其他組員負責的部分，如教學模式，能達到當初規劃的效果，才能讓整個專題有了完整性，讓整個系統實際運作起來。

而線上測驗系統做到現在，雖然趨近尾聲，但其實還是有許多新的點子想加進去，且每個功能也都還有可以加強的地方。例如，作一個適性測驗系統(CAT)，其測驗的方式是根據考生在此題的答案來作為下一題選題的依據，藉以更正確的判斷出考生的程度。而其實國際上TOEFL的考試方式近年來的電腦測驗方式也是採用類似這樣的測驗方式。在這專題中，整個系統是以考試為導向的內容，將來也希望能繼續擴充成加強教學的系統，這樣一來，這整個系統的實用性也將大為提高。

這次的專題，不但終於將所學充分運用，更學得了許多以前不瞭解、甚至沒接觸過的部分，實在受益匪淺。未來將持續地為這個作品作維護與擴充，希望能成為真正實用的成品。

王璽銘

參考資料

- [1][維雅譯]，Ellis Horowitz，Sartaj Sahni，Susan Anderson-Freed，Fundamentals Of Data Structure in C，松崗，1999
- [2]胡昭明，最新資料結構，松崗，1998
- [3]趙啟志，實力導引活用 PHP 與 MySQL 建構 Web 世界，博碩，2000
- [4]數位文化，PHP4 互動式網頁資料庫實戰手冊，松崗，2002
- [5]陳澤雄，全方位學習資料結構，基峯，2002
- [6]廖平，資料結構必考題型突破，儒林，1996
- [7]陳俊宏，PHP and MySQL 徹底研究，旗標，2000
- [8]Welling Luke、Laura Thomson，陳長念譯，PHP & MySQL 完全架站攻略，上奇科技，2001
- [9]林政錦，PHP 入門經典，第三波，2001
- [10]施威銘研究室，Flash 5 躍動的網頁，旗標，2000
- [11]吳明哲，Flash MX Action 網頁程式設計贏家，文魁，2003
- [12]何東隆，Flash MX 動感世界. 基礎應用篇，文魁，2002
- [13]數位文化，FLASH 5 初學網頁動畫設計實務，第三波，2000

附錄 A 檔案清單

檔案大致上分為 flash 檔案以及 php 檔案還有 html 檔案，flash 檔案為動畫檔，php 檔案為互動式網頁，而 html 檔案為網頁檔。

| | |
|--------------------------|---------------|
| config.php | 設定檔 |
| images | 存放底圖 |
| exam.htm | 首頁 |
| flash.htm | 教學頁面首頁 |
| flash | 存放 Flash 教學頁面 |
| pictures | 存放題目所需圖解 |
| main.htm | 首頁說明 |
| menu.htm | 左框架主選單 |
| 線上測驗： | |
| on_line_test_check.php | 檢查測驗密碼 |
| on_line_test.php | 測驗作答 |
| on_line_test_process.php | 作答完處理 |
| why.php | 作答完，每題正確選項說明 |
| 成績查詢： | |
| score.php | 顯示成績 |

| | |
|--------------------------|------------|
| score_why.php | 查看自己詳細作答情形 |
| 學生基本資料修改： | |
| check_passwd.php | 檢查密碼 |
| std_edit.php | 修改資料 |
| std_edit_ok.htm | 修改確認 |
| 系統管理： | |
| manager.htm | 系統管理主頁 |
| 帳號管理： | |
| std_del.php | 刪除學生帳號 |
| 題庫管理： | |
| question_show.php | 題目瀏覽 |
| question_add.php | 新增題目 |
| chapter_add.php | 新增單元 |
| chapter_edit.php | 修改單元 |
| question_edit.php | 編輯題目 |
| 考卷管理： | |
| test_create.php | 建立考卷 |
| test_choose_question.php | 從題庫選擇考題 |
| test_choose_ram.php | 隨機出題 |

| | |
|-------------------------|-------------------|
| test_choose_ok.htm | 選題完成 |
| test_paper_show.php | 考卷題目增加、刪除 |
| test_paper_show_one.php | 顯示某考卷考題 |
| test_data_edit.php | 考卷資料修改 |
| 考試管理： | |
| test_paper_open.php | 考卷開啟、關閉 |
| std_score_delete.php | 學生考題資料查詢、刪除 |
| 新增帳號： | |
| std_add.php | 新增單筆帳號 |
| 教學頁面： | |
| menu1.htm | 上框架選單 |
| main1.htm | 排序說明 1 |
| report1~4.htm | 排序說明 2~5 |
| 排序教學： | |
| decbubble-sort.htm | bubble-sort 教學 |
| decinsertion-sort.htm | insertion-sort 教學 |
| decquick-sort.htm | quick-sort 教學 |
| decselection-sort.htm | selection-sort 教學 |
| decmerge-sort.htm | merge-sort 教學 |

| | |
|-------------------|---------------|
| decradix-sort.htm | radix-sort 教學 |
| decshell-sort.htm | shell-sort 教學 |
| decheap-sort.htm | heap-sort 教學 |
| 討論板： | |
| add.php | 新增留言 |
| admin.php | 留言刪除 |
| board.php | 主頁面 |
| chk_data.js | 判斷留言資料 |
| input.php | 使用者留言 |
| re_input.php | 使用者回覆 |
| search.php | 搜尋文章 |
| view.php | 觀看文章 |

附錄 B 題目清單

此附錄包含我們所有的試題，每個排序法各十題，共八十題。

試題

1、請問下列 SELECTION 中哪一行程式碼有誤？

- (1) 1
- (2) 2
- (3) 3

```
void selectionsort (int list[],int last)
{
    int current,samillest,holddata,walker;
    for(current=0;current<=last;current++).....(1)
    {
        smallest=current;
        for(walker=current+1;walker<=last;walker++)
            if(list[walker]<list[smallest]).....(2)
                smallest=walker;
        holddata=list[current];
        list[current]=list[smallest];.....(3)
        list[smallest]=holddata;
    }
}
```

2、若資料為 56, 33, 54, 9, 1, 45 時，使用 SELECTION SORT 演算法，經過第幾次的排序後會呈現 1, 9, 54, 33, 56, 45？

- (1) 第 1 次
- (2) 第 2 次
- (3) 第 3 次
- (4) 第 4 次

3、圖中程式碼，哪一選項的步驟為錯誤？

- (1) A
- (2) B
- (3) C
- (4) D
- (5) E

```
void quicksort(element list[], int left, int right,)
{
    int pivot,i,j;
    element temp;
    if(left<right)
    {
        i=left; j=right+1;
        pivot=list[left].key;.....(A)
        do
        {
            do i++;
            while(list[i].key<pivot);.....(B)
            do j--;
            while(list[j].key>pivot);.....(C)
            if(i<j)
                SWAP(list[i],list[j],temp);.....(D)
        }while(i<j);
        SWAP(list[left],list[j],temp);.....(E)
        quicksort(list,left,j-1);
        quicksort(list,j+1,right);
    }
}
```

4、請問下列數列經由快速排序法後 pivot=12 為何？12, 2, 16, 30, 8, 28, 4, 10, 20, 6, 18

- (1) 12, 2, 6, 8, 4, 10, 28, 30, 20, 16, 18
- (2) 12, 2, 6, 8, 10, 4, 28, 30, 20, 16, 18
- (3) 12, 2, 6, 10, 8, 4, 28, 30, 18, 20, 16
- (4) 12, 2, 6, 10, 8, 4, 28, 30, 20, 16, 18
- (5) 12, 2, 10, 8, 6, 4, 28, 30, 20, 16, 18

排序方式為從 A 陣列中依序取出一個元素，並檢查該元素值與陣列 B 中的資料項作比較，然後將該元素插入陣列 B 中的適當位置，使陣列 B 中的元素維持在已排序的狀態。這是那一種排序法

5、

- (1) 插入排序法
- (2) 合併排序法
- (3) 快速排序法
- (4) 泡沫排序法
- (5) 選擇排序法

設欲排序的資料為：5, 1, 10, 2, 15, 3，若採用堆積排序

6、法(Heap Sort)，則當堆積樹(Heap tree)第三次建成時，其樹根節點資料內容是：

- (1) 3
- (2) 10
- (3) 15
- (4) 5

7、快速排序法所需的平均時間為何？

- (1) $O(\log N)$
- (2) $O(N^2)$
- (3) $O(N \log N)$
- (4) $O(N \log N^2)$
- (5) $O(N^2/2)$

在排序的過程中會在所有資料中挑選最小的一筆資料項，置於資料的第一個位置，然後在剩餘的資料中再找

8、尋最小的一筆資料項，置於資料的第二個位置，如此反覆執行，直到所有資料已由小到大排列。這是那一種排序法？

- (1) 插入排序法
- 2) 合併排序法
- (3) 快速排序法
- (4) 泡沫排序法
- (5) 選擇排序法

9、下列何種排序法之效率(efficiency)最佳？

- (1) 選擇排序法
- (2) 氣泡排序法
- (3) 合併排序法
- (4) 插入排序法
- (5) 選擇排序法

10、 SHELL SORT 是否需要額外的記憶體空間去作排序？

- (1) 是
- (2) 否

11、 判斷程式 InsertionSort(list, n)的內容何步出錯？

- (1) (1)錯了！
- (2) (2)錯了！
- (3) (3)錯了！
- (4) (4)錯了！
- (5) 全部正確無誤

```
void insertion_sort(element list[],int n)
{
    int i,j;
    element next;
    for(i=1;i<n;i++)
    {
        next=list[i];.....(1)
        for(j=i-1;j>0&&next.key<list[j].key;j--).....(2)
            list[j+1]=list[j];.....(3)
        list[j+1]=next;.....(4)
    }
}
```

12、 請使用 Radix_sort 來排序以下資料：11，7，4，123，31，88，349，9，31+，18。

- (1) 11，31，31+，123，4，7，88，18，349，9
- (2) 4，7，9，11，18，31，31+，88，123，349
- (3) 4，7，9，11，18，123，31，31+，349，88
- (4) 349，123，88，31+，31，18，11，9，7，4

若資料為

13、 Tim, Dot, Eva, Roy, Tom, Kim, Guy, Amy, Jon, Amn, Kay, Ron 經過謝爾排序法的第二次處理後結果為何？(設第一次間隔為 5，第二次間隔為 3)

- (1) Tim, , Tom, Kim, Guy, Amy, JonDot, Eva, Roy, Amn, Kay, Ron
- (2) Tim, Tom, Kim, Guy, Dot, Eva, Roy, Amy, Amn, Jon, Kay, Ron
- (3) Kim, Guy, Amy, Jon, Amn, Kay, Ron, Tim, Dot, Eva, Roy, Tom
- (4) Tim, Dot, Eva, Roy, Tom, Kim, Guy, Amy, Jon, Amn, Kay, Ron
- (5) Kay, Dot, Amy, Jon, Amn, Kim, Guy, Eva, Roy, Tom, Tim, Ron

14、 下列有關插入排序的說明，何者錯誤？

- (1) InsertionSort 在最壞情況需要 $O(n^2)$ 的時間完成
- (2) InsertionSort 在最佳情況可在 $O(n)$ 內完成
- (3) InsertionSort 平均需要 $O(n \log n)$ 的時間
- (4) InsertionSort 的效率由輸入資料的 Left-Out-Order 決定

15、 右圖的程式碼為 Heap_sort 的程式碼，所列選項何者有錯？

- (1) (A)部分

- (2) (B)部分
- (3) (C)部分
- (4) (D)部分

```

void Heap_sort(Element *tree, const int n)
{
    for(int i=n/2; i>=1; i--)
        adjust(tree, i, n);
    for(int i=1; i<=n; i++)
        swap(tree[i], tree[n-i+1]);
    for(int i=1; i<=n; i++)
        swap(tree[i], tree[n-i+1]);
}

void adjust(Element *tree, const int root, const int n)
{
    Element *p=tree[root];
    for(int i=2*root; i<=n; i+=1)
        if(tree[i].getKey()>tree[i+1].getKey())
            swap(tree[i], tree[i+1]);
    if(i<=n)
        swap(tree[root], tree[i]);
    tree[root]=p;
}
    
```

16、若資料為 29, 7, 39, 3, 64, 14, 62, 17, 50 而使用謝爾排序法經過一次的比較後會呈現何種情形？(設間隔為 4)

- (1) 3, 7, 14, 17, 29, 39, 50, 62, 64
- (2) 29, 3, 39, 7, 50, 14, 62, 17, 64
- (3) 29, 7, 39, 3, 50, 14, 62, 17, 64

當資料量大時，下列何種排序法所花費的時間最久？

17、(1)快速排序法(Quick Sort) (2)插入排序法 (Insertion Sort) (3)合併排序法(Merge Sort) (4)累堆排序法(Heap Sort)。

- (1) 1
- (2) 2
- (3) 3
- (4) 4

18、下列何者的排序法特性跟插入排序法類似？

- (1) 泡沫排序法
- (2) 選擇排序法
- (3) 快速排序法
- (4) 合併排序法
- (5) 謝爾排序法

在排序的過程中，由於排序的資料量太大，無法一次完

19、全放在主記憶體中，必須藉由輔助記憶體，此種排序方式叫作？

- (1) 快速排序(Quick Sorting)
- (2) 內部排序(Internal Sorting)
- (3) 外部排序(External Sorting)
- (4) 陣列排序(Sort of Array)

20、另 N 代表資料筆數，則 Heap_sort 的時間複雜度為

- (1) $O(\log N)$
- (2) $O(N)$
- (3) $O(N \log N)$
- (4) $O(N^2)$

(5) $O(N \log N^2)$

21、BubbleSort 在 worst case 中需要比較幾次才能完成排序呢？

- (1) $n-1$
- (2) $n(n-1)/2$
- (3) n^2
- (4) $n^2/2$
- (5) $n(n-1)$

22、請問合併排序的時間複雜度為何？

- (1) $O(N^2)$
- (2) $O(N \log N)$
- (3) $O(\log N)$
- (4) $O(N)$
- (5) $O(N \log N)$

23、將 5、4、3、2、1 依序插入空的 min-heap，則 heap 內容依序為：

- (1) 12453
- (2) 12345
- (3) 13254
- (4) 13245

24、插入排序法的平均時間與最壞時間階為？

- (1) $O(\log n)$
- (2) $O(n \log n)$
- (3) $O(\log n^2)$
- (4) $O(n^2)$
- (5) $O(n+1/2)$

25、快速排序法是否為穩定排序？是否為內部排序？

- (1) 是，是
- (2) 是，不是
- (3) 不是，是
- (4) 不是，不是

26、由 SELECTION 的演算法我們知道每次找出來的值會放在串列中的那一個位置？

- (1) 第一個位置
- (2) 第二個位置
- (3) 最後一個位置
- (4) 最後一個位置
- (5) 置於串列的中央

27、請問謝爾排序法最壞的時間複雜度為何？

- (1) $O(N)$
- (2) $O(N^2)$
- (3) $O(\log N)$
- (4) $O(N \log N)$
- (5) $O(N \log N^2)$

若以電腦處理橋牌比賽四位選手比賽者手持牌張的資料，並將個人所持牌花色予以排序（花色大小為：梅花

28、 \langle 方塊 \langle 紅心 \langle 黑桃，牌點大小為

$2<3<4<5<6<\dots<9<10<J<Q<K<A$ ），若使用 RADIX_SORT 來排序，則需使用多少個 QUEUE？

- (1) 1 個
- (2) 17 個
- (3) 4 個
- (4) 13 個

29、利用氣泡排序法排列 N 筆資料的順序，最多做幾次的排序循環？

- (1) $N / 2$ 次
- (2) N 次
- (3) $N - 1$ 次
- (4) $N + 1$ 次
- (5) $(N + 1)/2$ 次

30、請問 SHELL SORT 的平均時間複雜度是？

- (1) $O(N(\log N)^2)$
- (2) $O(N)$
- (3) $O(N \log N)$
- (4) $O(N \log N^2)$
- (5) $O(\log N)$

一個排序的方法稱為 Stable，意指在一個檔案中若存在

31、有幾個相同大小的鍵值，其經排序後相同的鍵值還是保持原來相對次序，對不對？

- (1) 對
- (2) 不對

32、試問 Radix_sort 平均時間複雜度為何？

- (1) $O(\log N)$
- (2) $O(N \log p N)$ ，(p 為基數之值，即 log 以 p 為底)
- (3) $O(N^2)$
- (4) $O(N)$

33、請問 Heap_sort 最壞情形的時間複雜度為何？

- (1) $O(N \log N)$
- (2) $O(N(\log N))$
- (3) $O(N^2)$
- (4) $O(\log N)$
- (5) $O(N^2)$

34、若有 1000 個資料，且使用謝爾排序法則平均所花時間為何？(時間單位為 ns)

- (1) 100000ns
- (2) 10000ns
- (3) 1000000ns
- (4) 100ns
- (5) 1000ns

35、[19、23、2、4、99、1、8]經由氣泡排序法之後的 PASS3 為何？

- (1) 2、4、1、8、23、19、99
- (2) 2、4、1、8、19、23、99
- (3) 2、4、8、1、19、23、99
- (4) 2、4、19、1、8、23、99

36、QuickSort 將 n 筆資料排序，在最差的情況其時間複雜度為何？

- (1) $O(N^2)$
- (2) $O(\log N^2)$

- (3) $O(N)$
- (4) $O(1)$
- (5) $O(\log N)$

37、請問下列何者為不穩定排序法？

- (1) 插入排序法
- (2) 泡沫排序法
- (3) 合併排序法
- (4) 選擇排序法

38、Radix_sort 是否為穩定排序？是否為內部排序？

- (1) 否，是
- (2) 否，否
- (3) 是，是
- (4) 是，否

39、下列為 SHELL 的主要程式碼請問下列哪一行程式碼有誤？

- (1) 1
- (2) 2
- (3) 3
- (4) 4

```
void shellsort(int list[],int last)
{
    int hold,incr,curr,walker;
    incr=last/2;
    while(incr==0).....(1)
    {
        for(curr=incr;curr<=last;curr++) {
            hold=list[curr];
            walker=curr-incr;
            while(walker>=0&&hold<list[walker]).....(2)
            {
                list[walker+incr]=list[walker];
                walker=(walker-incr);
            }
            list[walker+incr]=hold;.....(3)
        }
        incr=incr/2; .....(4)
    }
    return;
}
```

40、在排序的過程中會逐一掃描所有的資料，發現資料所在的位置錯誤，則會立即執行交換資料位置的動作。所以每一次掃描完畢後，至少會有一筆資料被置於正確的位置。這是那一種排序法？

- (1) 插入排序法
- (2) 合併排序法
- (3) 快速排序法
- (4) 泡沫排序法
- (5) 選擇排序法

41、下列各種排序法中，何者的平均排序時間最短？

- (1) 泡沫排序法
- (2) 快速排序法
- (3) 累堆排序法

(4) 插入排序法

下列何種排序法是用在整數資料的排序方法。先按照 42、最後一位數（呼叫某個 stable 排序法）排序，再按照倒數第二位數排序，... 最後按照第一位數排序？

- (1) QuickSort
- (2) BubbleSort
- (3) RadixSort
- (4) InsertionSort
- (5) MergeSort

43、寫出下列資料進行合併排序 PASS2 後的結果：
26, 5, 77, 1, 61, 11, 59, 15, 48, 19

- (1) 5, 26, 1, 77, 11, 61, 15, 59, 19, 48
- (2) 1, 5, 26, 77, 11, 15, 59, 61, 19, 48
- (3) 1, 5, 11, 15, 26, 59, 61, 77, 19, 48
- (4) 1, 5, 11, 15, 19, 26, 48, 59, 61, 77
- (5) 26, 5, 77, 1, 61, 59, 11, 15, 19, 48

44、請問 SELECTION SORT 的平均時間複雜度為何？

- (1) $O(N^2)$
- (2) $O(N\log N)$
- (3) $O(\log N)$
- (4) $O(N)$

45、請問 SELECTION SORT 最多會移動資料的複雜度為多少？

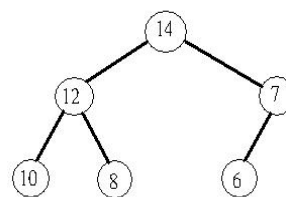
- (1) $O(N^2)$
- (2) $O(N\log N)$
- (3) $O(N)$
- (4) $O(N^3)$

46、下列排序法哪一種是先挑一個 pivot 令它就定位，大小元素分列左右？

- (1) QuickSort
- (2) InsertionSort
- (3) ShellSort
- (4) MergeSort

(5) RadixSort

47、右圖為一最大堆積(max heap)，求插入其值為 13 的節點(node)後之結果？依階度順序(level order)列出：



- (1) 14, 13, 12, 10, 8, 7, 6
- (2) 14, 12, 13, 10, 8, 6, 7
- (3) 14, 13, 12, 10, 8, 6, 7
- (4) 14, 12, 13, 10, 8, 7, 6

48、請問 Merge sort 是否需要另外的資料空間？

- (1) 是
- (2) 否

49、下列哪一個排序法有用到 divide and conquer 的策略？

- (1) bubble sort
- (2) merge sort
- (3) selection sort

50、當排序 n 個不同的資料時，插入排序法最少要將資料搬動幾次？

- (1) n
- (2) $n+1$
- (3) $n/2$
- (4) $n-1$
- (5) $n+1/2$

51、下列何種排序法之效率(efficiency)最佳？

- (1) 合併排序
- (2) 插入排序
- (3) 泡沫排序
- (4) 選擇排序

52、請問 SELECTION SORT 是內部排序法還外部排序法？

- (1) 內部排序法
- (2) 外部排序法

氣泡排序法(Bubble Sort)是利用相鄰資料兩兩相比而

53、完成資料由小到大或由大到小排序，假設有五個整數資料要做排序，最少要做幾次相鄰資料相比較的工作？

- (1) 5

- (2) 10
- (3) 15
- (4) 20
- (5) 25

54、若有資料為 89, 98, 77, 59, 107, 21 若採 2-WAY merge sort, 則需要經過幾個過程 (PASS), 才能完成排序?

- (1) 1
- (2) 2
- (3) 3
- (4) 4

55、以下何者用 heap 處理, 較有效率?

- (1) priority queue
- (2) k-way merge
- (3) depth first search
- (4) breadth search

56、不論輸入資料為何, 插入排序的元素比較總數較泡沫排序的元素比較之總數少?

- (1) 正確
- (2) 不正確

57、下列數列, 經由插入排序後的 pass5 為何? 4, 7, 11, 8, 6, 17, 3

- (1) 4, 7, 11, 8, 6, 17, 3
- (2) 4, 7, 6, 11, 8, 17, 3
- (3) 4, 7, 11, 6, 8, 17, 3
- (4) 4, 6, 7, 8, 11, 17, 3
- (5) 4, 7, 8, 11, 6, 17, 3

58、Store the number 30, 12, 9, 8, 6, 2, 3, in a min-heap tree in that order. Which of the following array represents this min-heap tree.

- (1) $A[1]=8, A[2]=3, A[3]=9, A[4]=12, A[5]=30, A[6]=3, A[7]=2$.
- (2) $A[1]=2, A[2]=3, A[3]=30, A[4]=12, A[5]=9, A[6]=8, A[7]=6$.

- (3) $A[1]=30, A[2]=6, A[3]=12, A[4]=2, A[5]=9, A[6]=3, A[7]=8$ 。
- (4) $A[1]=2, A[2]=8, A[3]=3, A[4]=30, A[5]=9, A[6]=12, A[7]=6$ 。
- (5) $A[1]=6, A[2]=12, A[3]=9, A[4]=30, A[5]=3, A[6]=8, A[7]=2$ 。

59、QuickSort 是採用下列何種策略？

- (1) Divide and Conquer
- (2) Greedy Method
- (3) Dynamic Programming
- (4) Back Tracing

以 Radix_sort (以 MSD 為例) 來排序下列英文人名時，

60、JIM、CATE、JOHN、KIDD、OWEN、PET、ALEN、NICKSON，第一次排序後的結果為何？

- (1) JIM、CATE、JOHN、KIDD、OWEN、PET、ALEN、NICKSON
- (2) NICKSON、JIM、ALEN、PET、OWEN、CATE、JOHN、KIDD
- (3) ALEN、CATE、JIM、JOHN、KIDD、NICKSON、OWEN、PET
- (4) JOHN、JIM、CATE、KIDD、PET、NICKSON、OWEN、ALEN
- (5) OWEN、JIM、JOHN、CATE、PET、KIDD、ALEN、NICKSON

61、請將下列整數以 Radix_sort 排序法來排序，{452, 371, 833, 278, 056, 749, 107, 015, 575, 714, 456, 312}

- (1) {371, 452, 312, 833, 714, 015, 575, 056, 456, 107, 228, 749}
- (2) {107, 312, 714, 015, 228, 833, 749, 452, 056, 456, 371, 575}
- (3) {015, 056, 107, 228, 312, 371, 452, 456, 575, 714, 749, 833}
- (4) {056, 015, 107, 321, 456, 371, 452, 575, 833, 749, 714, 228}

62、若想要一次就找出排序中的最小或最大值時，下列哪一個排序法比較適合？

- (1) 泡沫排序法
- (2) 合併排序法
- (3) 快速排序法

(4) 選擇排序法

63、請問 Selection Sort 不需要一個額外的紀錄空間，以做為 SWAP 資料的緩衝區，對不對？

(1) 對

(2) 不對

64、插入排序是屬於下列哪一種排序？

(1) 內部排序

(2) 外部排序

65、下列哪一個排序演算法的速度最差？

(1) 氣泡排序法

(2) 合併排序法

(3) 快速排序法

(4) 選擇排序法

(5) 插入排序法

66、討論 BUBBLE SORT 的 average case 的時間複雜度？

(1) $O(n^2)$

(2) $O(n \log n)$

(3) $O(n)$

(4) $O(\log n)$

(5) $O(\log n^2)$

67、請問 Selection sort 是否為穩定排序？

(1) 是

(2) 否

68、請問 Merge sort 是否為穩定排序？

(1) 是

(2) 不是

69、在排序過程中將從剩下的元素中挑最小的排序方法是哪一種？

(1) 泡沫排序

(2) 選擇排序

(3) 快速排序

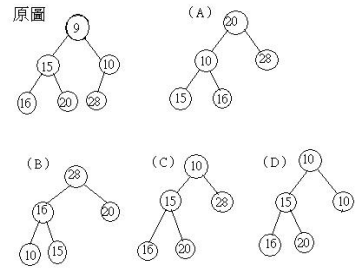
(4) 合併排序

70、右圖為 Radix_sort 的程式碼，哪一步驟選項錯誤？

- (1) (A)
- (2) (B)
- (3) (C)
- (4) (D)

最小堆疊樹(min_heap)是一個完整二元數(complete binary tree)，最小樹的定義是一個節點(node)的鍵值

71、(key value)不可大於它的兒子(如果有兒子的話)的鍵值。考慮將右圖的最小堆疊樹去掉(delete)鍵值為"9"之節點，而仍保持為一最小堆疊樹，請問右圖那個為正確解？



- (1) (A)圖
- (2) (B)圖
- (3) (C)圖
- (4) (D)圖

72、Radix_sort 又可分為最高有效數字排序(MSD)，最低有效數字排序(LSD)，下列敘述何者正確？

- (1) 以 LSD 法排序其數字會以小到大排列
- (2) 以 LSD 法排序其數字會以大到小排列
- (3) 以 MSD 法排序其數字會以小到大排列
- (4) 以上皆非

73、請問 MERGE SORT 跟 QUICK SORT 哪一個為穩定排序？

- (1) MERGE SORT
- (2) QUICK SORT
- (3) 兩個都是
- (4) 兩個都不是

74、一串數列其最小值為 13，最大值為 2006，若使用 Radix_sort 來排序，則需執行幾次才能排序完成？

- (1) 4
- (2) 6
- (3) 2
- (4) 3

75、以下說明是否正確?插入排序是屬於穩定排序，且所需的額外空間很少。

- (1) 對

(2) 不對

76、BubbleSort 是否為穩定排序？

(1) 是

(2) 不是

擬以快速排序法，將下列串列由小到大排序：Carol，

77、Bob，Alice，Peter，John。假若我們以串列的排序，經過一回合後，此串列的次序為何？

(1) Carol，Alice，Bob，Peter，John

(2) Carol，Alice，Bob，John，Peter

(3) Bob，Carol，Alice，Peter，John

(4) Alice，Bob，Carol，Peter，John

(5) Alice，Carol，Bob，Peter，John

78、請問 MERGESORT 跟 QUICK SORT 的 worse case 哪一個的時間複雜度較低？

(1) QUICK SORT

(2) MERGESORT

(3) 一樣

79、若資料量較大時可以先用何種排序法使資料呈現接近依序排列的情形？

(1) 泡沫排序法

(2) 合併排序法

(3) 快速排序法

(4) 選擇排序法

(5) 謝爾排序法

80、如右圖哪一個二元樹(Binary)是最小堆積樹(Minimum heap)？

(1) 右圖(A)

(2) 右圖(B)

(3) 右圖(C)

(4) 右圖(D)

