

致理技術學院

資訊網路技術系 實務專題報告

FLASH 動畫在資料結構分析中之應用

指導老師：林正平

學生：陳建邦 69334101

朱慧娟 69334108

謝盈盈 69334136

張嵐菁 69334140

中華民國 97 年 1 月

致理技術學院

資訊網路技術系 實務專題報告

FLASH 動畫在資料結構分析中之應用

學生：陳建邦 69334101

朱慧娟 69334108

謝盈盈 69334136

張嵐菁 69334140

本成果報告書經審查及口試合格特此證明。

指導教師：_____

中華民國 97 年 1 月

專題研究授權書

本授權書所授權之專題研究為_____陳建邦, 朱慧娟, 謝盈盈, 張嵐菁_____共____4____人, 在致理技術學院資訊網路技術系 96_____學年度第__1__學期完成資網實務專題。

專題名稱：

FLASH 動畫在資料結構分析中之應用

同意 不同意

本組同學共____4____人, 皆同意著作財產權之論文全文資料, 授予教育部指定送繳之圖書館及本人畢業學校圖書館, 為學術研究之目的以各種方法重製, 或為上述目的再授權他人以各種方法重製, 不限地域與時間, 惟每人以一份為限。

上述授權內容均無須訂立讓與及授權契約書。依本授權之發行權為非專屬性發行權利。依本授權所為之收錄、重製、發行及學術研發利用均為無償。上述同意與不同意之欄位若未勾選, 該組同學皆同意視同授權。

指導教授姓名：

專題生簽名：

學號：

中華民國 年 月 日

誌 謝

本論文的完成和先要感謝指導教授林正平老師這一年來耐心的指導、老師在專題上對於我們的關心 and 照顧，使我們一步步完成整個專題作業。林正平老師，除了參加我們的專題評審外，也對於我們的專題提供了很好的建議與指導。再此對林正平老師致上最誠摯的敬意和謝意。

在學習與研究的過程中，感謝所有幫助過我們的學長與同學們，當我們遇到專題相關作業程式的困難時，他們皆會適時的回答我們的問題並啟發我們的思維，解決我們在資料結構專題架構中產生的一些瓶頸，不致讓整個專題作業停滯不前。

僅以本論文獻給我們親愛的父母、師長和陪我們一起走過這段日子的同學、學長學弟們。感謝你們這一年來對我們的指導與鼓勵，謝謝！

。

摘 要

此專題旨在研習運用 FLASH 軟體，就其豐富的動態特質依繪圖技術、程式思考模型、問題處理的方法三要點，來建構出互動式的資料庫輔助教材。讓複雜不易理解的程式語言透過動畫來讓同學們更能夠理解書本上所表達的理論。

隨著網際網路的發展，以及網路傳輸技術的進步，網路化的教學將會更加普及，並且適用在各個階段的應用上。因為它可以超越時間與空間的侷限，不限定種族族群，而且可以迅速的連結到世界各地相關的研究資源，因此在學前教育，學校教育，甚至在社會中，個人生涯規劃，在職進修，藉由網路教學可以節省多少高昂的資源成本，並且更能增加管理的機制；其發展應是刻不容緩的，而更多時候使用動態的動畫比使用靜態式的圖片更能讓使用者理解其資料結構所代表的意義。此專題研究目標即為增進資料結構的教學效率。將課本上提供的資料結構作為主要研究對象，探討各項資料結構的共同性質以及其動畫上的表示方法。此次專題所開發出來的工具利於往後應用於資料結構的教學。

關鍵詞：FLASH 動畫 資料結構 JAVA 程式語言

目 錄

專題研究授權書.....	4
誌 謝.....	5
摘 要.....	6
目 錄.....	7
圖 目 錄.....	9
表 目 錄.....	15
第一章 緒論.....	16
第一節 Flash 輔助教學其重要性與發展演進	16
第二節 研究主題發展背景與動機	16
第三節 研究範圍與報告書架構	18
第二章 FLASH 及資料結構理論與技術探討	20
第一節 介紹 FLASH 其發展演進	20
第二節 FLASH 技術探討.....	23
第三節 資料結構的基本概念及原理	40
第三章 FLASH 動畫與資料結構輔助教學應用	63
第一節 系統架構與功能簡介	63
第二節 系統操作流程	66

第四章 系統呈現.....	67
第一節 系統畫面	67
第二節 專題實作遇到之問題與解決	91
第五章 結論	93
第一節 線上輔助教學未來的發展性	93
第二節 專題的心得與感想	93
參考文獻	94

圖目錄

圖 1-1	時間軸工具圖.....	25
圖 2-1	圖層與影格圖.....	26
圖 3-1	漸變技術製作.....	27
圖 3-2	漸變技術製作.....	27
圖 3-3	漸變技術製作.....	28
圖 4-1	導引線技術製作	29
圖 4-2	導引線技術製作	29
圖 4-3	導引線技術製作	30
圖 4-4	導引線技術製作	30
圖 4-5	導引線技術製作	31
圖 4-6	導引線技術製作	31
圖 4-7	導引線技術製作	32
圖 5-1	遮罩技術製作.....	33
圖 5-2	遮罩技術製作.....	33
圖 5-3	遮罩技術製作.....	34
圖 5-4	遮罩技術製作.....	34
圖 5-5	遮罩技術製作.....	35

圖 5-6 遮罩技術製作.....	35
圖 5-7 遮罩技術製作.....	36
圖 5-8 遮罩技術製作.....	36
圖 5-9 遮罩技術製作.....	37
圖 5-10 遮罩技術製作	37
圖 6-1 按鈕技術製作.....	38
圖 6-2 按鈕技術製作.....	38
圖 6-3 按鈕技術製作.....	39
圖 6-4 按鈕技術製作.....	39
圖 6-5 按鈕技術製作.....	39
圖 6-6 按鈕技術製作.....	40
圖 7-1 資料結構的基礎圖	41
圖 8-1 串列結構圖.....	41
圖 8-2 基本鏈結串列-在鏈結串列刪除節點圖	42
圖 9-1 環狀鏈結串列圖	43
圖 9-2 雙向環狀鏈結串列圖	44
圖 10-1 堆疊結構圖.....	44
圖 11-1 陣列堆疊的 push 圖	45
圖 12-1 陣列堆疊的 pop 圖	46

圖 1 2-2	鏈結陣列的 push 圖	46
圖 1 2-3	鏈結陣列的 pop 圖	47
圖 1 3-1	佇列結構圖.....	50
圖 1 3-2	鏈結串列實作佇列圖	50
圖 1 3-3	環狀佇列圖.....	51
圖 1 4-1	二元樹例圖.....	53
圖 1 4-2	中序走訪方式例圖	53
圖 1 4-3	前序走訪方式例圖	54
圖 1 4-4	後序走訪方式例圖	55
圖 1 4-5	二元搜尋樹例圖	56
圖 1 5-1	深度優先搜尋法例圖	57
圖 1 5-2	寬度優先搜尋法例圖	58
圖 1 5-3	基隆市與各都市間矩陣表示圖	60
圖 1 6-1	首頁畫面呈現	63
圖 1 7-1	中序運算式轉換成後序運算式首頁圖	67
圖 1 7-2	中序轉換成後序演算法簡介圖	68
圖 1 7-3	中序運算式的計算圖示	68
圖 1 7-4	「*」存入運算子堆疊圖	69
圖 1 7-5	中序轉後序運算式後所得結果圖	70

圖 1 8-1	建立二元樹教學動畫-初始畫面	70
圖 1 8-2	建立二元樹教學動畫-文字呈現。簡單介紹其規則圖 .	71
圖 1 8-3	建立二元樹教學動畫-圖形表示	71
圖 1 8-4	建立二元樹教學動畫 完滿二元樹架構圖	72
圖 1 8-5	建立二元樹教學動畫二元樹陣列表示法	72
圖 1 9-1	二元樹的中序走訪教學動畫「初始畫面」	73
圖 1 9-2	二元樹的中序走訪演算法「文字呈現」	73
圖 1 9-3	二元樹的中序走訪「圖形運算」	74
圖 2 0-1	二元樹的前序走訪演算法 文字呈現	74
圖 2 0-2	二元樹的前序走訪規則 文字呈現	75
圖 2 0-3	二元樹的前序走訪教學動畫 圖形運算	75
圖 2 0-4	二元樹的前序走訪教學動畫 圖形運算	76
圖 2 1-1	二元搜尋樹的節點刪除-初始畫面	77
圖 2 1-2	二元搜尋樹的節點刪除葉節點 圖形表示	77
圖 2 1-3	二元搜尋樹的節點刪除葉節點 圖形表示	78
圖 2 1-4	二元搜尋樹的節點刪除教學動畫 圖形表示	78
圖 2 1-5	二元搜尋樹的節點刪除教學動畫 圖形表示	79
圖 2 1-6	二元搜尋樹的節點刪除教學動畫-文字呈現	79
圖 2 2-1	二元樹的應用運算式處理教學動畫 初始畫面	80

圖 2 2-2	中序運算式的運算元 圖形表示	80
圖 2 2-3	二元樹的應用運算式處理教學動畫 圖形運算	81
圖 2 2-4	二元樹的應用運算式處理教學動畫 文字呈現	81
圖 2 2-5	二元樹的應用運算式處理教學動畫 文字呈現	82
圖 2 3-1	動態鏈結佇列首頁圖	83
圖 2 3-2	鏈結串列建立佇列架構圖	83
圖 2 3-3	鏈結串列建立佇列架構圖	84
圖 2 3-4	鏈結串列建立佇列架構圖	84
圖 2 4-1	環狀佇列基本架構圖	85
圖 2 4-2	環狀佇列-存入元素圖	85
圖 2 4-3	環狀佇列-存入元素圖	86
圖 2 4-4	環狀佇列-存入元素圖	86
圖 2 4-5	環狀佇列-取出元素圖	87
圖 2 4-6	環狀佇列-取出元素圖	87
圖 2 4-7	環狀佇列-取出元素圖	88
圖 2 4-8	環狀佇列-確認佇列是否已滿圖	88
圖 2 4-9	環狀佇列-確認佇列是否已滿圖	89
圖 2 4-1 0	環狀佇列-確認佇列是否已滿圖	89
圖 2 4-1 1	環狀佇列-確認是為空佇列圖	90

圖 2 4-1 2 環狀佇列-確認是為空佇列圖 90

圖 2 4-1 3 環狀佇列-確認是為空佇列圖 91

表 目 錄

表 1 基隆到各頂點城市間的距離表	60
-------------------------	----

第一章 緒論

第一節 Flash 輔助教學其重要性與發展演進

目前電腦輔助教學 (Computer Assisted Instruction) 是結合教學與電腦工具，以生動活潑的方式來表現教學內容，使用者可依自己的能力、興趣、時間及進度來調整學習速度。而且在學習的過程之中可以有互動性的產生，讓學習的效果能夠透過這種互動的產生而更加有趣，進而使學習欲望更加強烈，充分達到個人化教學與寓教於樂的目標。已成為教育資訊化最為重要的內容。因其不但可以超越時間和空間的限制，誘導學生思維，更能促進優質教學，在教學工作中充分應用資訊技術，正深刻地改變著我們的理念。而從最早期比較文字化到目前更是結合多媒體影音效果，更能夠增加使用者的學習欲望，而且得到較佳的成效。以動畫的形式來表達所要傳播的知識的電腦輔助教學可說是當今最主流的趨勢了，其中電腦輔助教學又可細分為單機的形式和網路的形式兩種。隨著科技的進步，CAI 所呈現的外貌會更加多變且多元化。

第二節 研究主題發展背景與動機

此次專題所發展的系統是利用動畫方式呈現抽象的資料結構以達到資料結構上教學的目的，除要考慮使用上的方便性也要讓使用者

能夠更輕易地了解各類型資料結構的運作方式。本系統的主要發展目的在提升使用者對資料結構的學習成效，讓使用者能以更有效率的方式理解抽象的資料結構。在目前教學上還未能找到完整呈現資料結構動畫的系統，本專題將大部份的資料結構動畫整合在一套系統，讓使用者更能了解自己所學的資料結構程式。本專題除研究各項資料結構與圖形化的相關性外。

目前大部分資料結構的教學中，總是以靜態的圖片來教導學生。可是大部分的資料結構著重在結構的變化上，老師必須花長時間在黑板上畫出這些資料結構的變化。但是一個網路化教學系統的建立，不僅只考量到網頁畫面，在選擇考慮教材的內容、教學的技術、材料、與媒介也是一項重要的工作，可用來組織架構和展現教學的內容。題材的呈現也需注意使用者是否能在最短時間內清楚理解整個課程概念、架構及內容。而此專題即希望能夠以動畫方式表現資料結構的動作，可讓老師教學方便、使用者更容易理解。由最簡單的基礎觀念延伸到複雜的邏輯分析，若使用者能夠按部就班，一步步跟進，將會很快的對課程得心應手。

由於人在學習東西時，若單只有平面閱讀文字上的解釋，往往不能有極度有效率地完全吸收；因此，我們在課程方面的製作，大多配合著圖形的呈現與透過動畫的解析。文字敘述加上生動的動畫圖形的

演繹，讓使用者能夠一目了然，迅速且清晰的了解一個概念或一個事件的過程，對於課程的學習能力自然大大提昇，以達到更佳的學習成效。

第三節 研究範圍與報告書架構

一、研究範圍

成品展現範圍-資料結構動畫呈現。

各項技術研究範圍

1. FLASH：運用漸變、位移、遮罩技術展現動畫、把平面所無法想像

實際的資料結構利用圖片、動畫解釋概念。

2. 資料結構：，做出簡易的基本觀念性資料結構，對有興趣的同學但

對資料結構的觀念並非常清楚者，給予一些基本的觀念，

我們的初衷只是希望對於不熟悉資料結構的同學，能夠給

予一點幫助，並增進對資料結構的興趣，在這方面我們利

用一些簡易的敘述及圖形的動畫來做輔助教學。

二、報告書架構

第一章旨在電腦輔助教學(CAI)與網路化教學的簡介與現況製作，並說明本系統的動機與目的，並且包含本系統的介紹與特色。第二章則旨在介紹 FLASH 軟體及資料結構之基本介紹及主要在資料庫程式

設計輔助教學中應用理論。讓真正有興趣參與研究資料結構的閱覽者更能進一步的進入狀況。第三章介紹了我們的系統架構、組織、功能。第四章則為系統之設計與實作及其優缺點並就製作本系統時所遇到的問題提出解決方法。第五章為結論，主要在介紹網路化教學未來的發展性，並說明完成本系統時所獲得的心得與感想。

第二章 FLASH 及資料結構理論與技術探討

第一節 介紹 FLASH 其發展演進

1995 年，是網際網路高速發展的一年，但同時，大部分人已經不滿足於網際網路的平面瀏覽模式，於是喬納森-蓋伊憑藉著敏銳的市場觀察力，設計出了 Future Splash Animator 向量動畫軟體，這就是 Flash 的前身。微軟公司在一開始，就看中了這個能在網路上傳播動畫的軟體，而迪斯尼公司，則使用 Future Splash Animator 設計了 Disney online 網站，這個網站解決了當網路帶寬和動畫之間的矛盾。同時給 Macromedia 公司重創。

當時，Macromedia 公司的 Shockwave 播放器，是網路上 Director 交互電影的唯一解決之道，而迪斯尼公司使用 Future Splash Animator 軟體製作動畫網站時，Macromedia 公司已經覺察到了危機，他們認為交互動畫在互聯網上的傳播，是很有前景的產業，於是，他們找到了喬納森-蓋伊，他也認為 Future Splash Animator 在 Macromedia 公司的經營下，才可以真正得到發展，雙方促成了這場並購案。1996 年 11 月，Future Splash Animator 正式更名為 Flash 1.0（也有說是 Flash2.0）。實際上 Macromedia 也沒有想到，今天的

Flash 會如此火爆，Flash 當時僅為 Shockwave 在互聯網領域的一個解決之道，這一點，從 Flash 文件的擴展名就可以看出（swf 文件名是 Shockwave Flash 的縮寫）。

但是當 1998 年 5 月 Flash 3.0 推出時，這一切似乎已經發生了變化，Flash 作為真正的互聯網多媒體軟體，被應用於更多的領域。因為 Flash 的出眾表現，在 1998 年的下半年，多媒體軟體被 Macromedia 公司重新洗牌了。Director 主要應用於交互電影的製作，而 Flash 則應用於互聯網。1999 年 6 月，Macromedia 公司推出了 Flash4.0，並且推出了 Flash4.0 播放器，這一舉動現在看來，帶給了 Flash 無限廣闊的前景，正是因為播放器的變革，使得 Flash 成為真正意義上的交互多媒體軟體。

2000 年 8 月，Macromedia 公司推出了 Flash5.0，在原有功能表命令的基礎上，模擬 Director 軟體 Lingo 語言的模式，發展出第一代 Flash 專用交互語言，命名為 Action Script1.0。這是 Flash 的一項重大革命，因為在此之前，Flash 只可以稱之為流媒體軟體，而當大量的交互語言出現後，Flash 才成為了交互多媒體軟體，這項重大的變革對今後 Flash 的發展，意義都是相當深遠的。同時，Flash 中的繪畫部份停滯不前，這表明 Macromedia 公司其實並不想讓 Flash 成為專業的動畫軟體，他們的想法更為廣闊。

在 Flash5.0 發佈時，Macromedia 將 Flash 的發展，與 Dreamweaver 和 Fireworks 整合在一起，被稱之為網頁三劍客。在 2002 年 3 月 Flash MX (Flash6.0) 發佈時，這一策略被 Macromedia 重新調整，在 MX Studio 的陣營中，新增加了 Freehand 10 和 Coldfusion MX。新增加的這兩個軟體，FreeHand 是向量繪圖軟體，可以看作是補充 Flash 在繪畫方面的不足，而 Coldfusion MX 是多媒體後台，Macromedia 用它來補充 Flash 在後台面缺陷，從這一次產品整合可以看出，Flash MX 實際上已經成為 MX Studio 系列產品中的主打產品。

2003 年 8 月，Macromedia 推出了 Flash MX 2004，從 Flash MX 開始，Flash 就陸續增加了動態圖像、動態音樂、動態流媒體等技術，並且為 Flash 添置了組件，項目管理、預建數據庫等等功能，使 Flash 已經具備了挑戰 HTML，成為網站主流技術的可能性。同時，Macromedia 已經不局限于讓 Flash 在網路上發展，Flash MX 2004 實現了對手機和移動設備的支援，為 Flash 成為跨媒體播放軟體，創造了條件。如今，Flash 已經初步具備跨平臺交互多媒體的特性，被稱之為“最小巧的多媒體平臺”。可以說，這一切的發展，是 Macromedia 公司和喬納森·蓋伊始料不及的，但 Flash 的取勝之道，卻是喬納森·蓋伊在最初設計 Flash 時，就已經奠定了，那就是向量動畫、關鍵性技術和流

式播放。

第二節 FLASH 技術探討

Flash 軟體的問世，為網頁設計者提供了另一種製作動態網頁的新選擇，它結合了音效、影像、動畫，具備了高品質、體積小、互動性高的優點，FLASH 動畫有別於以前我們常用於網路的 GIF 動畫，他採用的是向量繪圖技術，向量圖就是可以無限放大，而圖像品質不損失的一種格式的圖，由於動畫是由向量圖構成的，就大大的節省了動畫文件的大小，在網路頻寬局限的情況下，提升了網路傳輸的效率；可以方便的下載觀看。它的特色大約可分下列幾項

Flash 動畫設計的三大基本功能

Flash 動畫設計的三大基本功能是整個 Flash 動畫設計知識體系中最重要、也是最為基礎的，包括：繪圖和編輯圖形、補間動畫和遮罩。這是三個緊密相連的邏輯功能，並且這三項功能自 Flash 誕生以來就存在。

[1]繪圖和編輯圖形:繪圖和編輯圖形不但是創作 Flash 動畫的基本功能，也是進行多媒體創作的基本工具。只有基本技巧紮實，才能在以後的創作 FLASH 上得心應手。使用 Flash professional 8 繪圖和

編輯圖形。這是 Flash 動畫創作的三大基本技巧的最需要學習的。在繪圖的過程中要學習怎樣使用元件來組織圖形元素，這是 Flash 動畫的入門首要技巧。

[2]補間動畫:補間動畫是整個 Flash 動畫設計的核心，也是 Flash 動畫的最大優點，它有動畫補間和形狀補間兩種形式。在應用影片剪輯元件和圖形元件創作動畫時，有一些細微的差別，這些細微的差別會影響力整個 FLASH 動畫的效果。

[3]遮罩:遮罩是 Flash 動畫創作中所不可缺少的這是 Flash 動畫設計三大基本功能中最特殊的一點。使用遮罩配合補間動畫，更可以創建更多豐富多采的動畫效果：圖像切換、火焰背景文字、管中窺豹等都是實用性很強的動畫。並且，從這些動畫實例中，可以舉一反三設計更多實用性更強的動畫效果。遮罩的原理非常簡單，但其實現的方式多元化，特別是補間動畫以及影片剪輯元件結合起來，可以創建千變萬化的形式，可以應用這些形式作個總結概括，從而使自己可以創建各種形式的動畫效果。flash 動畫說到底就是“遮罩+補間動畫+逐幀動畫”的混合物，通過這些元素的不同組合，從而可以創建千變萬化的效果。

FLASH 所使用之技術

一· 動畫製作:

時間軸技術：時間軸是用來控制並記錄我們的 Flash 互動電影，它顯示每一個 Scene 場景，Frame 影格，以及 Layer 圖層。我們利用時間軸來將不同的場景或是影格串連成一個互動的 Flash 電影。

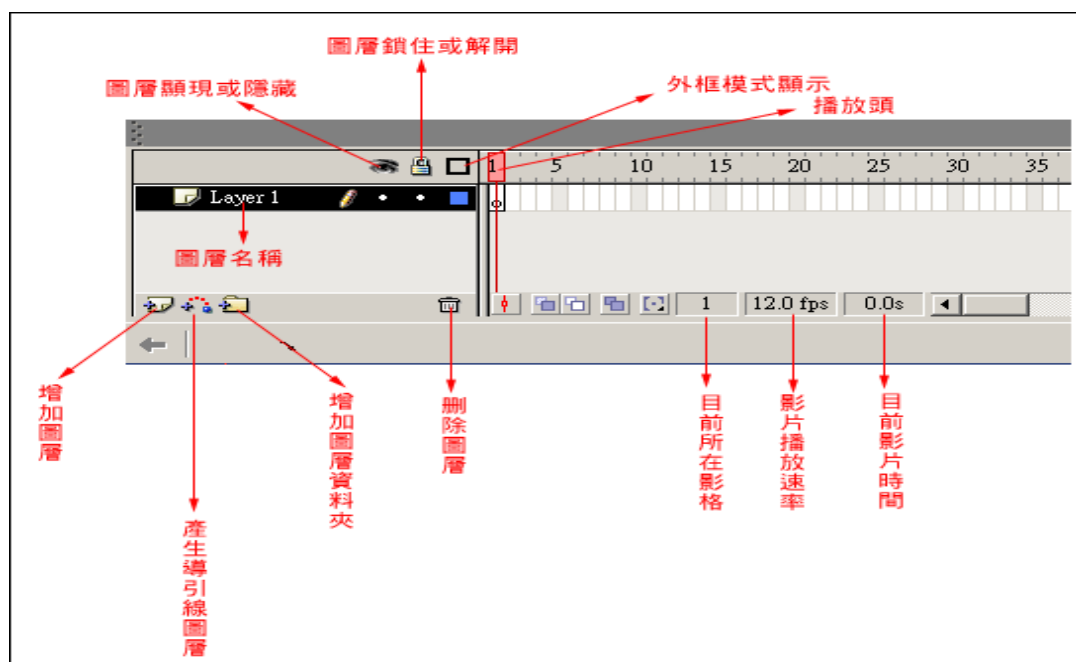


圖 1-1 時間軸工具圖

關於時間軸:其實 Flash 就如同一本互動的數位化書籍，而時間軸就是呈現所有互動內容的面版。每一個場景就如同書籍的各個章節，每一影格就如同章節的各個頁面。當然 Flash 要比一般實體書更複雜也更有意思了。而每一影格至少都會有一"片"畫面來讓我們繪製或是

讀入物件，但我們可以依需求再增加數個透明片，然後在特定的透明片來繪製物件，以方便我們管理及做更複雜的設計。

這個透明片就是"圖層"。如圖 2-1，我們在影格 1 加入圖 2(Layer2)，讓 Frame 1 這個影格畫面擁有兩個圖層。

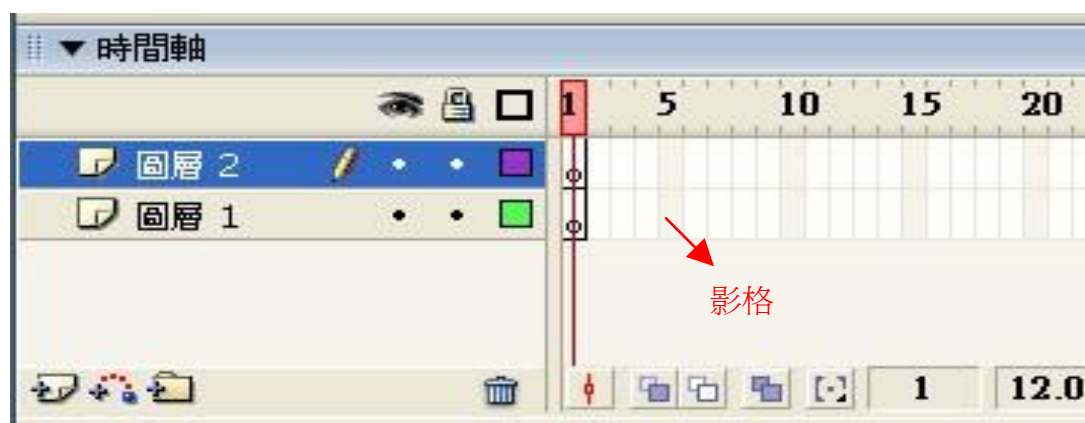


圖 2-1 圖層與影格圖

影格(Frame)是一般動畫軟體所通用的計數單位，而時間軸就以影格為時間基準來安排每個元件的演出。影格 Frame 可分為一般影格(Frame)以及關鍵影格(Keyframe)。

一般影格：除了關鍵影格之外，出現在時間軸的影格都是一般影格。

關鍵影格：決定動畫變化的開始或結束的影格，也就是至少有 2 個關鍵影格才能構成一段動畫。

漸變技術:在圖層 1 的影格內繪製一個藍色的球，在影格 20 插入空白
關鍵影格，圖 3-1。



圖 3-1 漸變技術製作

1. 接下來在這個空白的關鍵影格畫一個黃色的矩形，如圖 3-2。

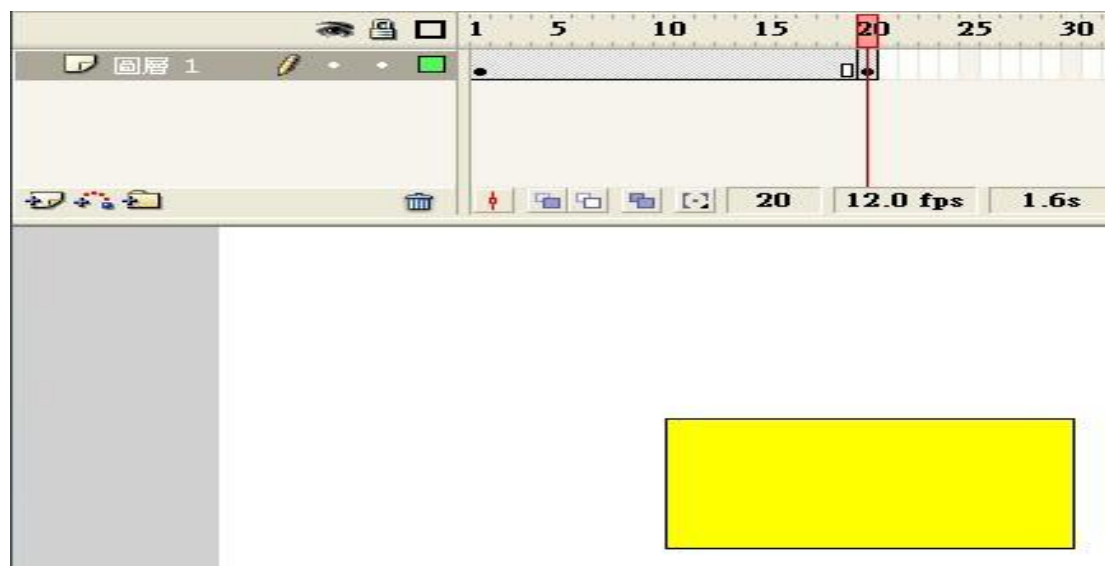


圖 3-2 漸變技術製作

滑鼠點在 Frame25 之前的任一影格，在屬性面板的補間動畫欄位選擇形狀如圖 3-3。

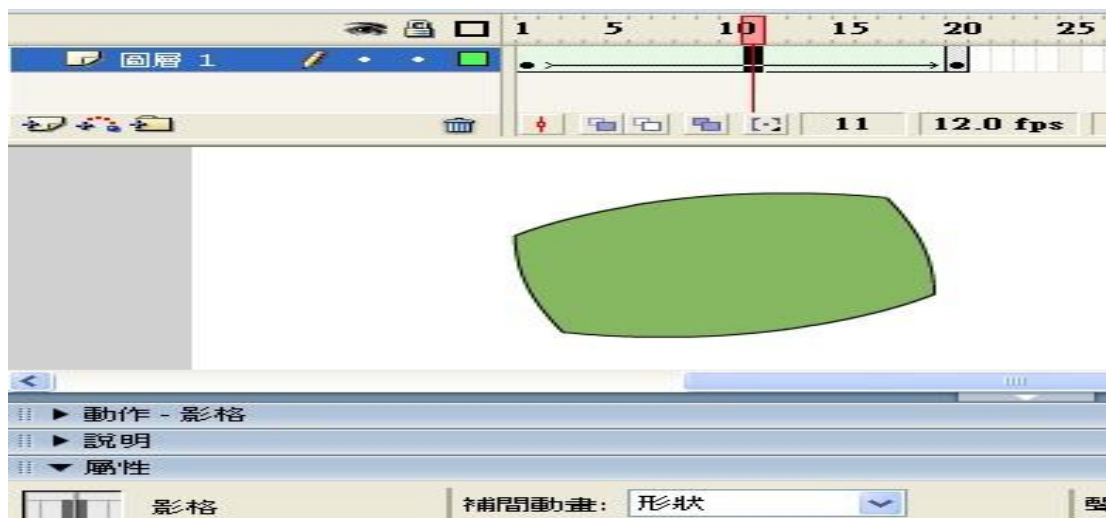


圖 3-3 漸變技術製作

就完成變形動畫了，文字及點陣圖也可以做變形動畫，但必須先打散變成形狀，才能做出完整的漸變動畫。

導引線的使用：我們可以畫一條不規則的線條，讓元件的動畫，跟著這條路線來做不規則曲線運動。畫一個簡單的人型，再把它轉成元件，再將圖層 1 命名為"人型"，做一段長度為 25 格的動畫，如圖 4-1

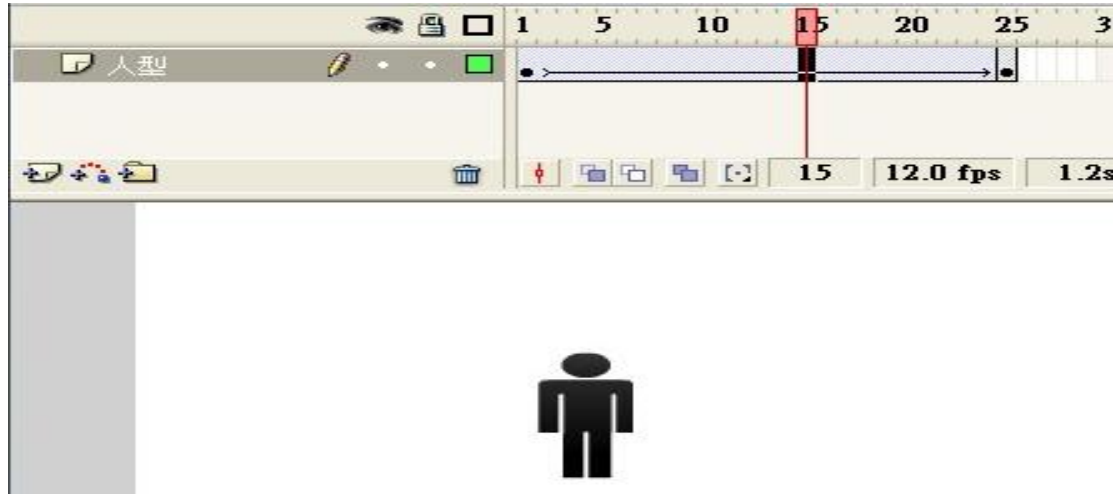


圖 4-1 導引線技術製作

，執行加入導引線的指令，請直接點選增加導引線的圖示按鈕，就會產生一個導引線:人型的圖層，如圖 4-2。

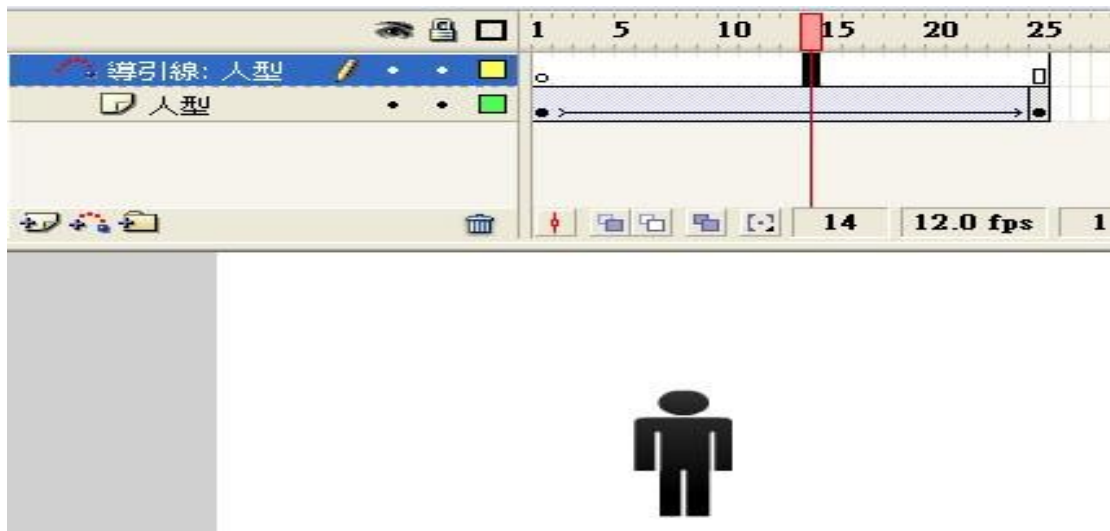


圖 4-2 導引線技術製作

用鉛筆在導引線:人型圖層，隨意畫線，如圖 4-3。請注意線條只能有一條，可以隨意交叉，但不能重疊，也不能有其他分支。

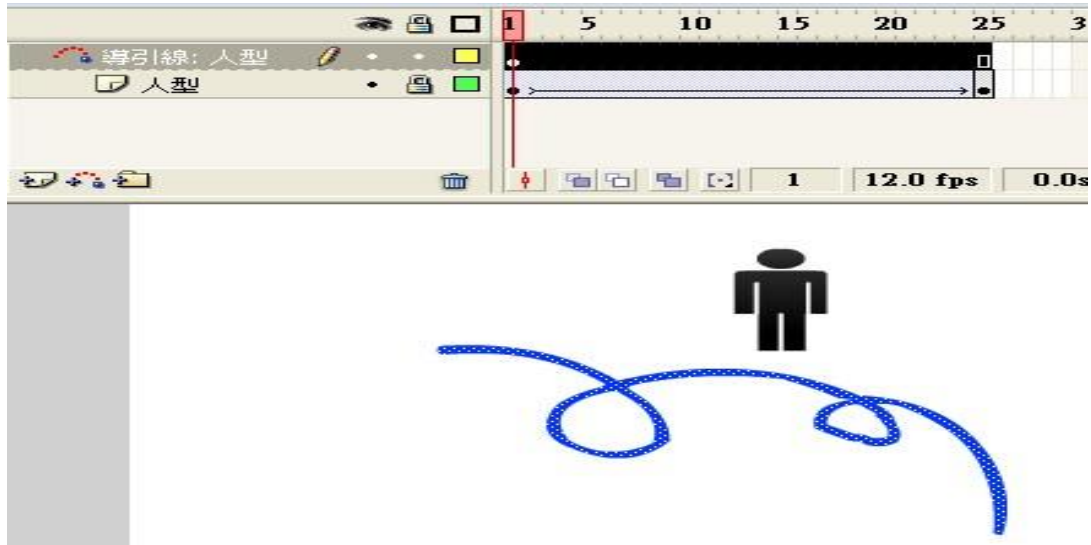


圖 4-3 導引線技術製作

再將導引線圖層鎖住，將"人型"圖層設為編輯狀態，確定貼齊物件工具是按下去的狀態，點選影格 1，再選擇舞台上的飛碟軸心，拖曳貼齊到線條的其一端點，如圖 4-4。

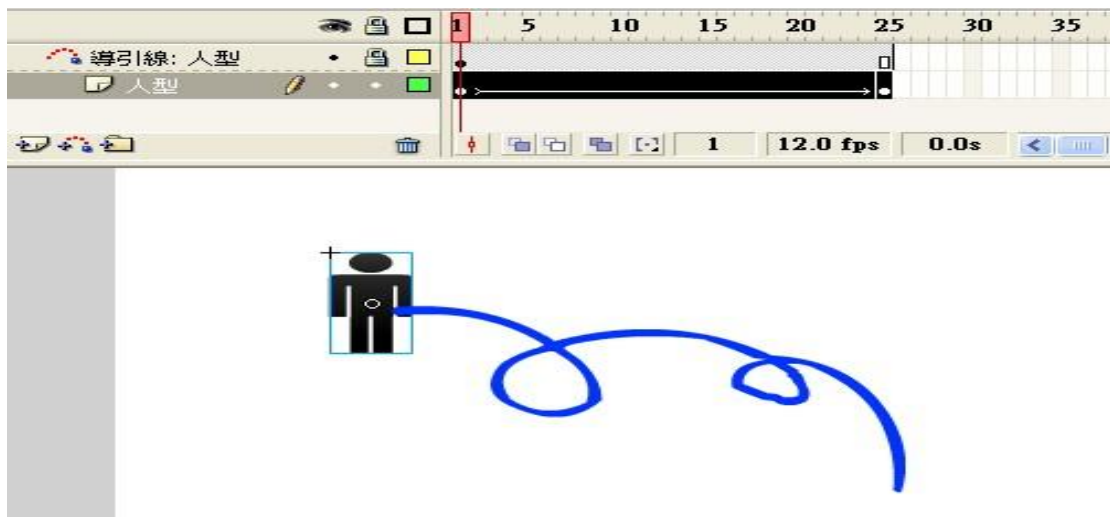


圖 4-4 導引線技術製作

點選影格 40，再選擇舞台上的人型軸心，移到線條的另一端點，如圖 10 因為有開啟貼齊工具，所以到一定距離時，人型的軸心會自動靠到端點。

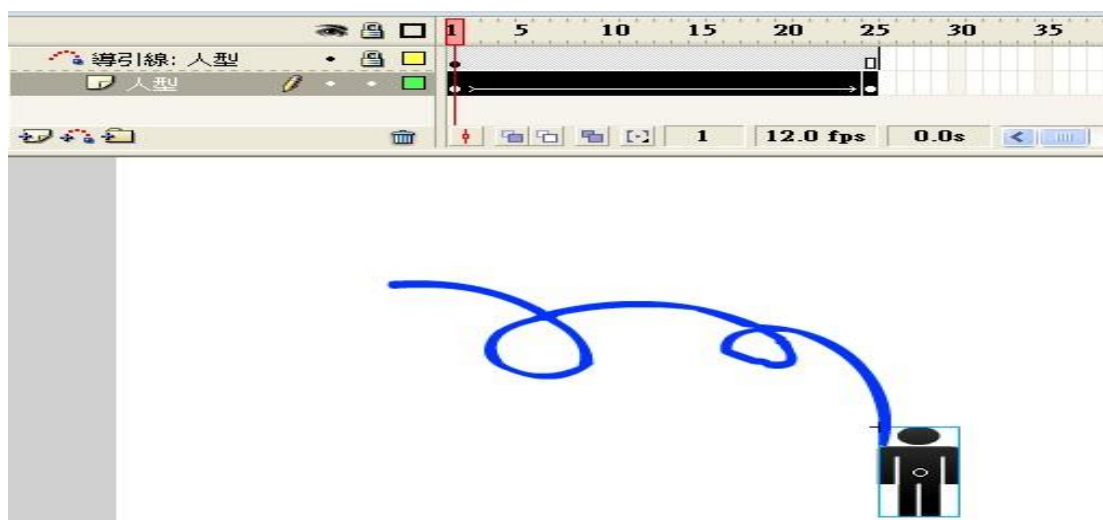


圖 4-5 導引線技術製作

加入導引線圖層의其它方法

方法 1. 如圖 4-6。



圖 4-6 導引線技術製作

方法 2. 在需要導引線圖層的某一圖層按右鍵，執行增加導引線，如圖 4-7



圖 4-7 導引線技術製作

2. 遮罩技術：遮色片是 Flash 裡面一個特別的圖層，作用與一般影像繪圖軟體的遮片功能類似，你只要在原本圖片的圖層上新增一個圖層，再按下滑鼠右鍵設成「遮色片」，原本的圖片就可以照遮色片去剪裁。

先建立一個圖層，並且繪製了人形圖案，現在再新增一個空白圖層，
圖 5-1。

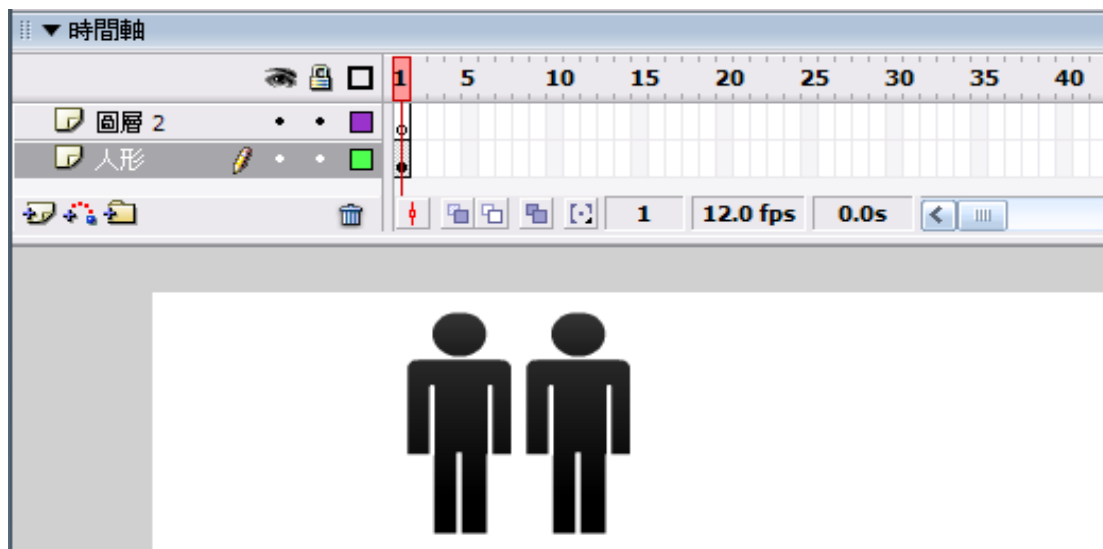


圖 5-1 遮罩技術製作

在新的圖層上繪製圓形，並建立成移動補間動畫，如圖 5-2。

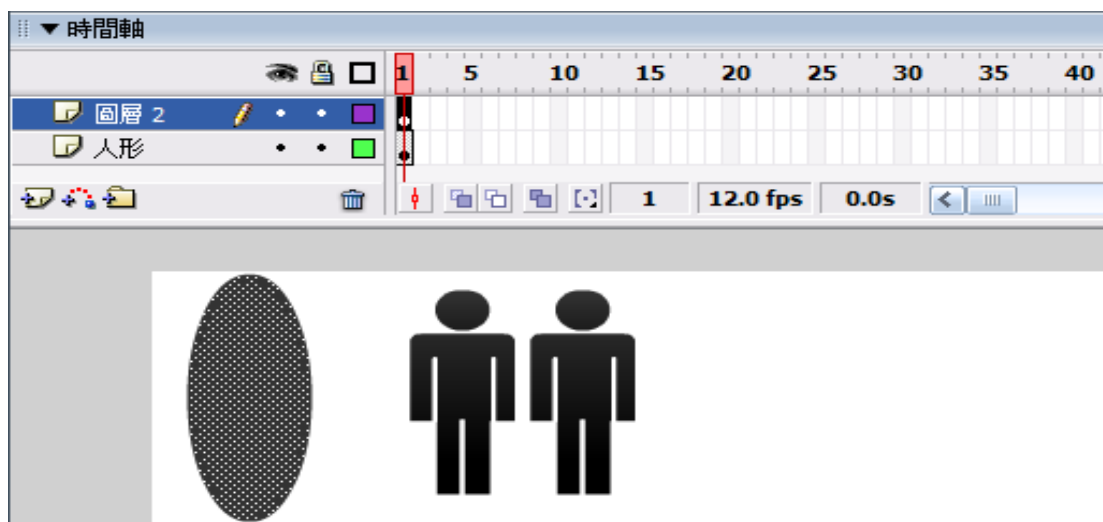


圖 5-2 遮罩技術製作

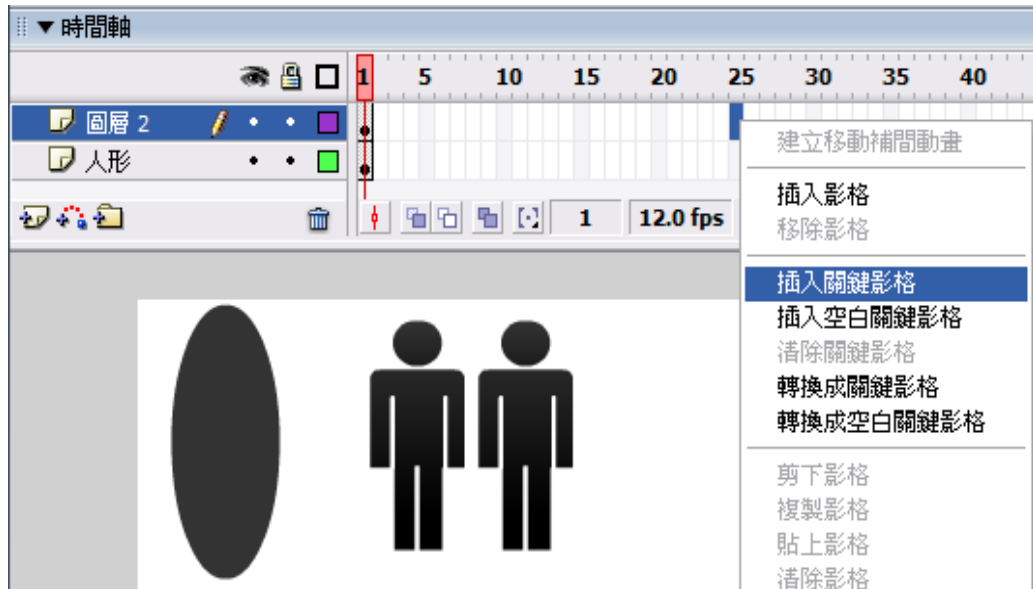


圖 5-3 遮罩技術製作

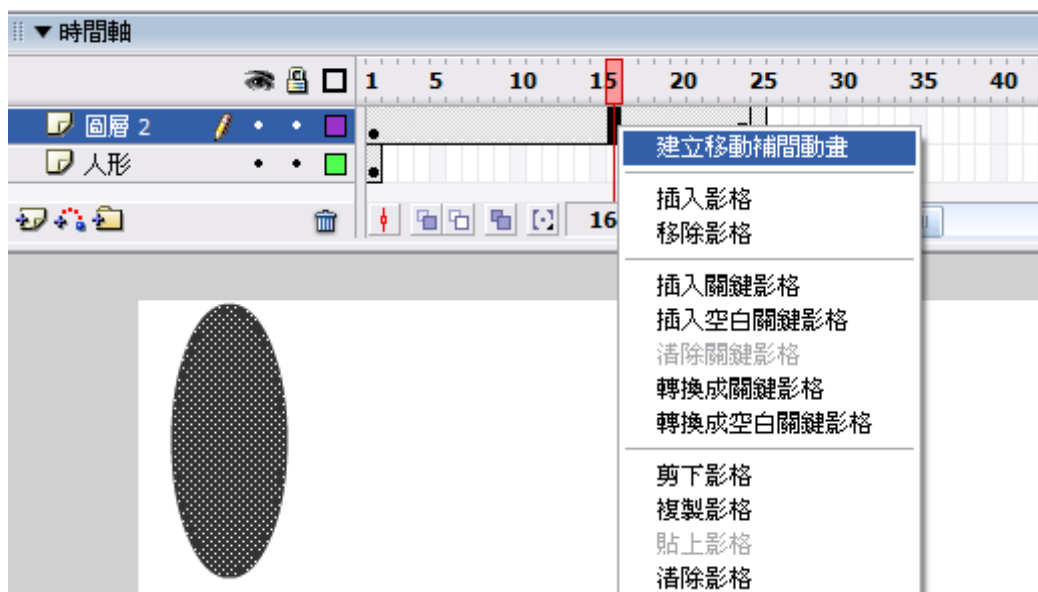


圖 5-4 遮罩技術製作

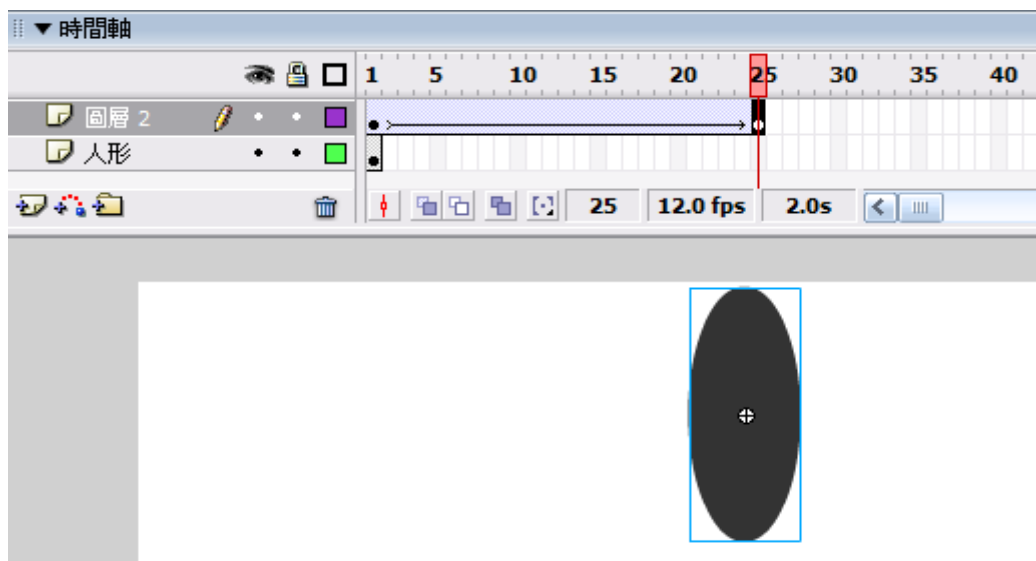


圖 5-5 遮罩技術製作

再回到原本這個人形的影格，在圖層 1 結束的相同位置，同樣插入關鍵影格，並建立移動間補動畫：

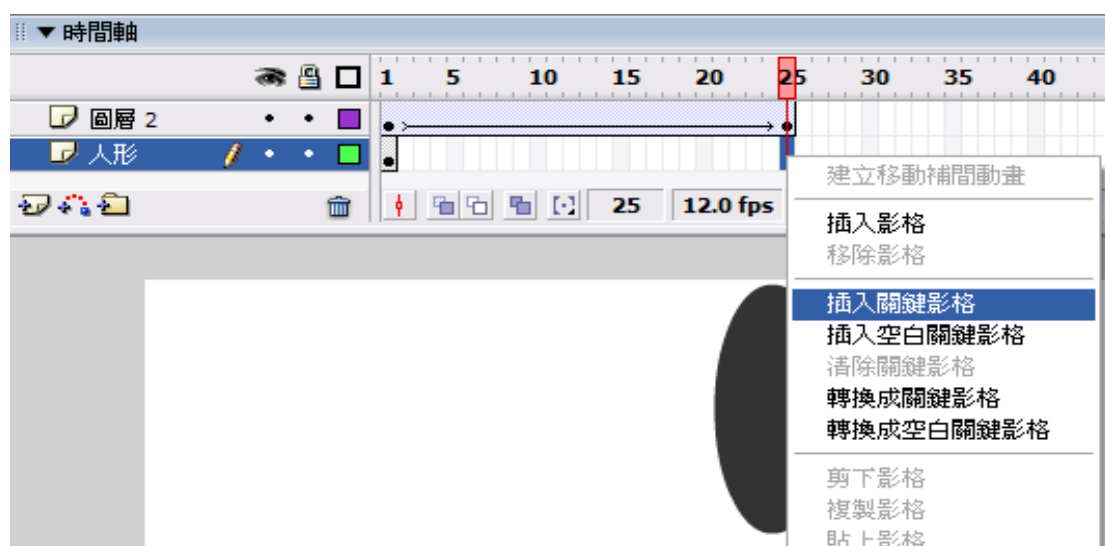


圖 5-6 遮罩技術製作

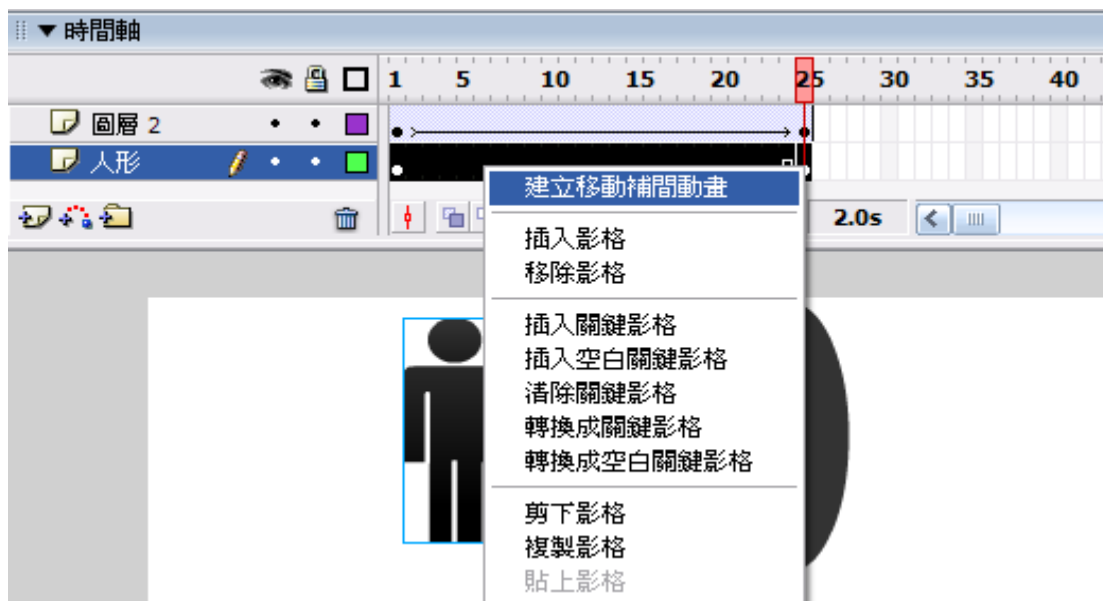


圖 5-7 遮罩技術製作

注意這個圓形的圖層要在人形圖層的上方，並按右鍵，選擇遮色片：



圖 5-8 遮罩技術製作

可以視情況修改遮色片圖層的名字：

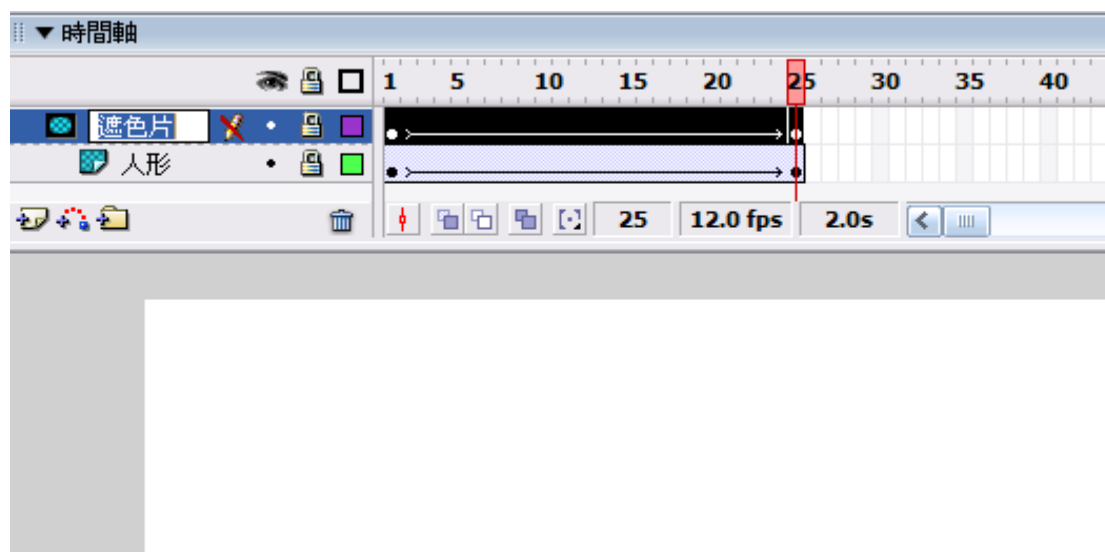


圖 5-9 遮罩技術製作

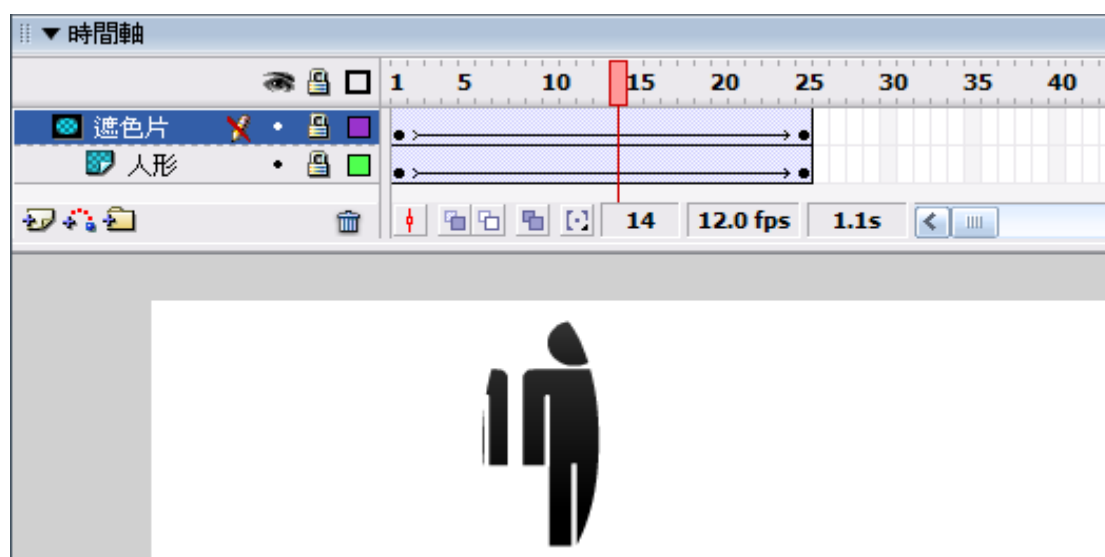


圖 5-10 遮罩技術製作

3. 按鈕技術：按鈕是網頁中常常使用到的東西，在 Flash 中也是非常重要的一種特殊元件。而且使用 Flash 可以做出有趣的按鈕效果！

先畫好按鈕在一般狀態下的圖形，如圖 6-1。



圖 6-1 按鈕技術製作

按 F8 將整個圖形轉換為元件，性質選擇按鈕。



圖 6-2 按鈕技術製作

在按鈕元件上按滑鼠右鍵，選擇在原地編輯。此時會進入按鈕元件的編輯狀態。

按鈕元件一種特殊的元件，它的前四個影格有特別的性質，

分別是：一般、滑鼠經過、按下和反應區（執行）。



圖 6-3 按鈕技術製作

在平常的時候，按鈕元件會顯示出一般影格，當滑鼠滑過按鈕時，會顯示滑鼠經過影格，按下按鈕時則會顯示按下影格，反應區（執行）影格則定義了按鈕可以接受滑鼠感應的範圍。

切換到滑鼠經過影格，按 F6 插入關鍵影格。在這個影格中畫上滑鼠滑過時按鈕的圖形。



圖 6-4 按鈕技術製作

切換到按下影格，按 F6 插入關鍵影格。在這個影格中畫上按鈕被按下時的圖形。



圖 6-5 按鈕技術製作

切換到反應區（執行）影格，按 F6 插入關鍵影格。在這個影格中畫上按鈕接受滑鼠感應的範圍。

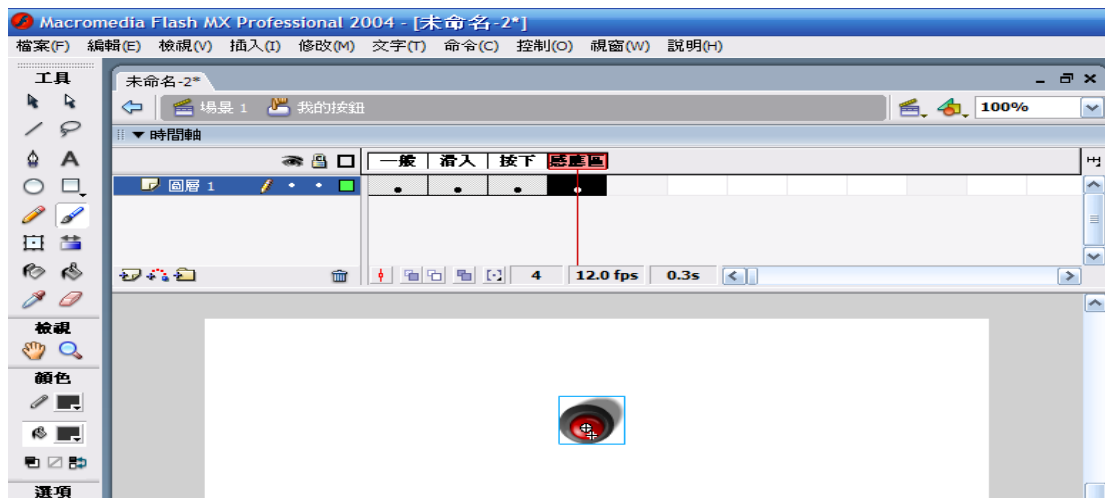


圖 6-6 按鈕技術製作

若按鈕按下時要有聲音，則先切換到按下影格，再從圖庫中將之前檔案->讀入的聲音給拖曳到畫面內。完成後跳出按鈕編輯模式，選擇控制->啟動簡易按鈕，就可以試試按鈕的效果了！

第三節 資料結構的基本概念及原理

資料結構(Data Structures)是計算機科學領域的一門重要課程,其目的研究程式使用的資料在電腦記憶體中的儲存方式,以便撰寫程式處理問題時,能夠使用最佳的資料結構,在學習基本程式設計,如 Java 語言後,資料結構提供的相關理論,可以快速應用程式設計的視野,在 Java 程式運用陣列、串列、堆疊、佇列、二元樹和圖形等各結構來處理資

料讓程式執行更有效率。資料結構的基礎:策略或方法是指如何選擇最恰當的資料結構，並且將這些資料轉換成有用的資訊，如圖所示:

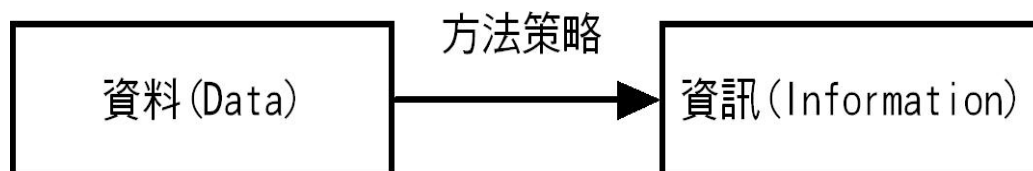


圖 7-1 資料結構的基礎圖

一、鏈結串列(Linked Lists): 「鏈結串列」是一種動態的資料結構，我們使用陣列將元素以一定的距離來儲存，但是，在有序串列裡，有些運算如插入、刪除其中的元素就變的非常複雜。鏈結串列屬於一種動態的資料結構，換句話說，我們可以插入或刪除串列的節點，藉著鏈結表示法(Linked representation)，此問題可獲得巧妙的解法。我們儲存串列中下一個元素的位址，因此，和串列的各個元素相關連的是一個節點(node)，它包含了資料部分和一個指向串列中下一個元素的指標(pointer)。

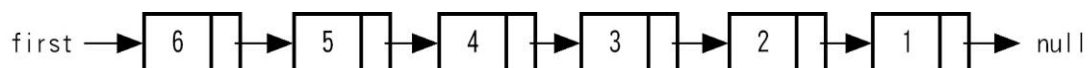


圖 8-1 串列結構圖

但因為節點位置的不同，可以分為三種情況，如下所示：

- 將節點插入串列第一節點前: 只要將新節點的指標指向串列的第一個節點, 指向新節點的指標就成為串列的 first 指標。
- 將節點插入串列最後一個節點後: 將串列最後一個節點的指標指向新節點, 然後將新節點的指標指向 null, last 指標指向新節點。
- 將節點插在串列的中間: 如果節點是插在 p 和 q 節點指標之間, p 是指向 q 的前一個節點, 只需將 p 指標指向新節點, 然後將新節點指標指向 q, 即可插入節點。

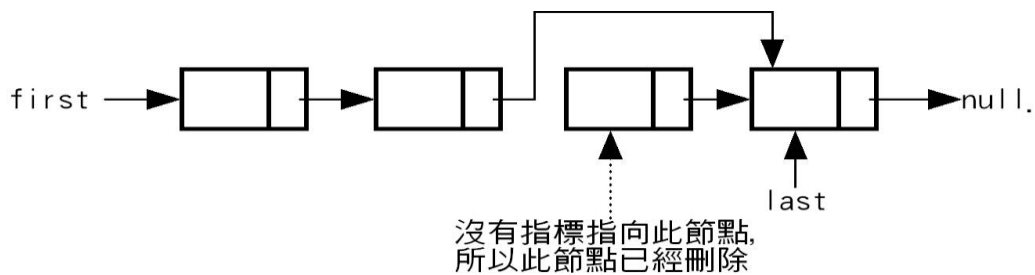


圖 8-2 基本鏈結串列-在鏈結串列刪除節點圖

二、環狀鏈結串列: 在單向鏈結串列中, 節點的指標都指向同一方向的下一節點, 直到最後指向 null。如將最後一個節點指標改為指向向鏈結串列開始的第一個節點, 串列指標的方向將成為環狀, 如下圖所示:

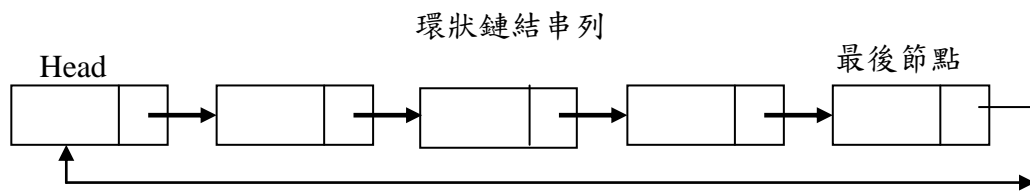


圖 9-1 環狀鏈結串列圖

其具有以下二大特點：

1. 最後節點的鏈結不接地，而是指向首節點
2. 從任何一個節點開始，都可以走訪所有節點。只要繞回原起點就可以停止

雙向鏈結串列：是一種常見的鏈結串列結構，這是一種允許無方向性走訪的鏈結串列，因為單向鏈結串列擁有方向性，在走訪串列時，只能單方執行走訪，換句話說，從第一個節點搜尋到最後一個節點很快，但，反過來走訪就會大費周章。

為改進單向鏈結串列的走訪，可將二方向相反的單向鏈結串列結合起來，建立一種無方向性的鏈結串列，這就是雙向鏈結串列，如下圖所示：

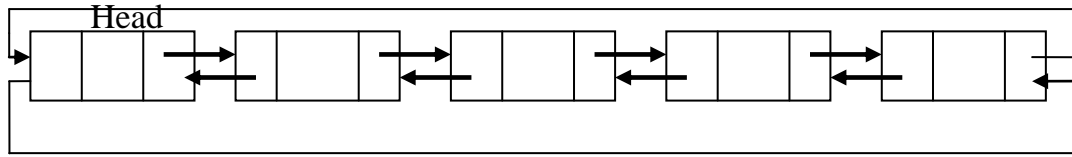


圖 9-2 雙向環狀鏈結串列圖

二、堆疊(Stacks): 定義，堆疊是一種抽象的有序串列，其插入與刪除皆須在同一端進行，而此一端稱為頂端(top)，另一端稱為底部(Bottom)。另堆疊有以下兩大特性:只允許從頂端存取資料，存取的順序是後出先進 (Last Out, First In)，也就是後存入堆疊的資料，反而先行取出所以又稱為 LIFO (Last-in-First-out) 結構。

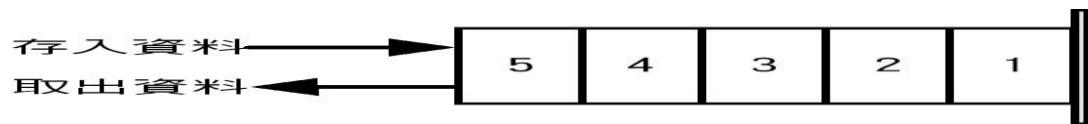


圖 10-1 堆疊結構圖

堆疊可以用陣列或串列結構來實作，以下是簡單的方法介紹:

有幾個運算可以作用在堆疊這種結構上面：

本系統將特別介紹 二種 push() 方法，pop() 方法

一、 push() 方法:

堆疊的特性是只能從堆疊頂端存取資料，所以需要額外的 top 成員變動來指向堆疊頂端的陣列索引值，使用此索引值將資料存入堆疊。

Push()方法將資料存入堆疊的步驟，如下所示：

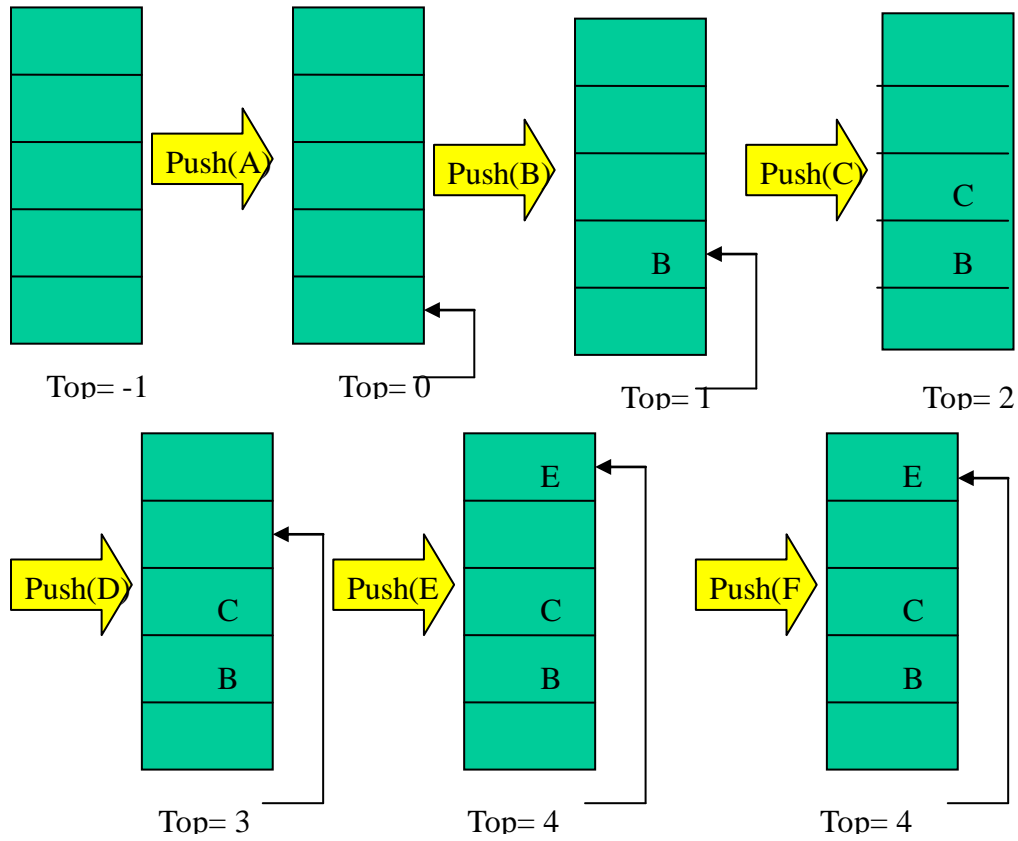


圖 11-1 陣列堆疊的 push 圖

二、 pop()方法:從堆疊取出資料的是 pop()方法,取出資料的步驟

以下圖所示:

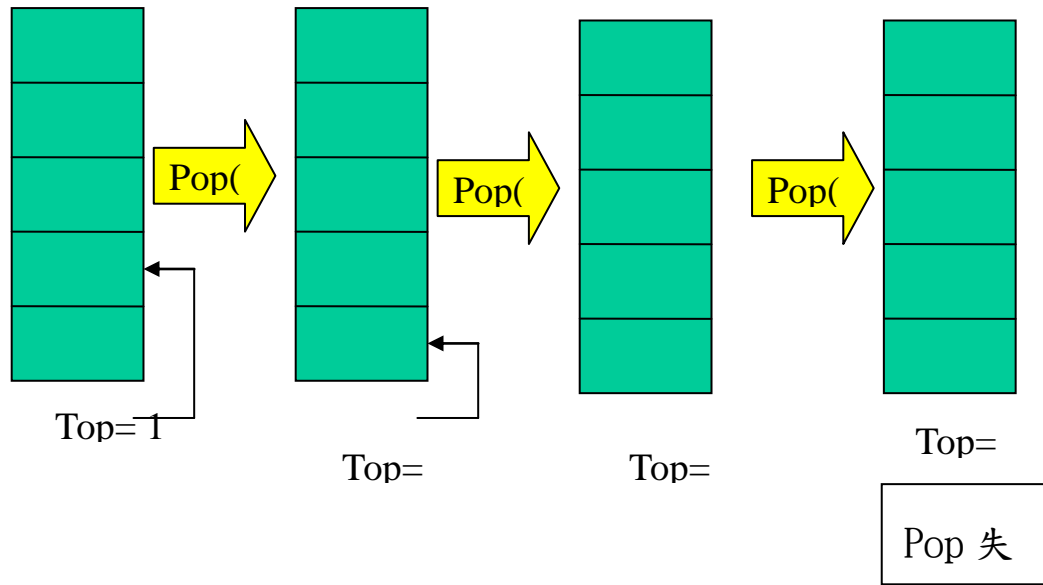


圖 12-1 陣列堆疊的 pop 圖

堆疊的介面方法為：`isStackEmpty()`、`push()`和 `pop()`方法，除了可以使用陣列來實作外，還可以用單向鏈結串列來建立堆疊結構。

`Push()`方法:可將資料存入堆疊，也即是新增節點成為串列的第一個節點,其步驟如下圖所示:

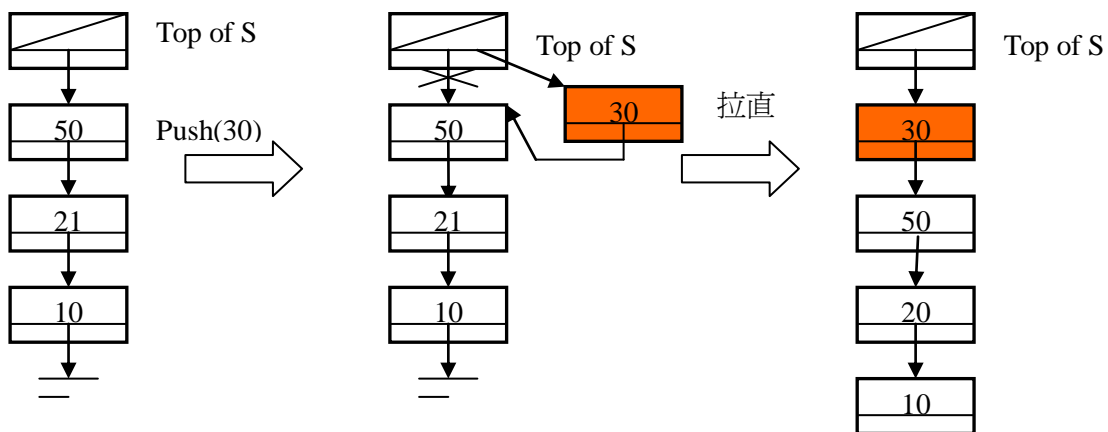


圖 12-2 鏈結陣列的 push 圖

`Pop()`方法:從堆疊取出資料的是 `pop()`方法，也就是刪除串列中的

第一個節點，其取出的步驟如下圖所示：

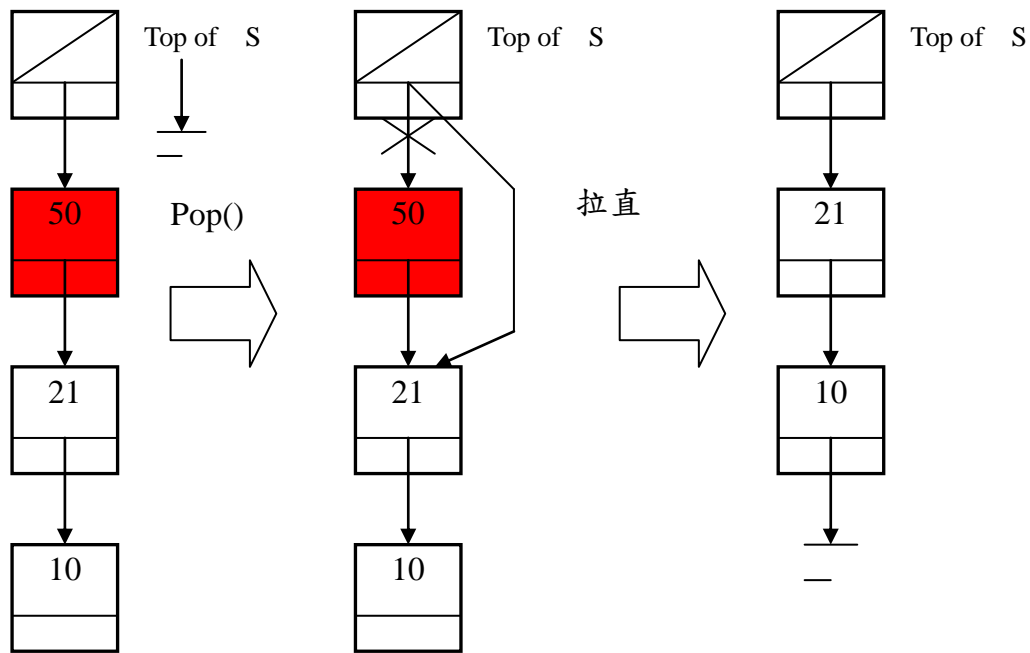


圖 12-3 鏈結陣列的 pop 圖

堆疊最常應用的問題是運算式的計算與轉換，可利用堆疊來儲存運算元式運算子，來計算前序、中序和後序運算式的結果，或是進行運算式的轉換，如將中序運算式轉換為後序運算式。運算式的種類：依運算子位在運算式中的位置，可分為三種，如下所示：

1. 中序式 (infix)：運算子在運算元的中間，例如： $A + B$
2. 後序式 (postfix)：運算子在運算元的後面，例如： $A B +$

後序運算式計算的演算法：後序運算式計算的演算法可以依照前述計算過程來推導，其完整步驟如下所示：

Step1: 使用迴圈從左至右依序讀入後序讀入後序運算式。

(1) 若讀入的是運算子，、則：

1)從運算元堆疊取出所需的運算元。

2)計算此運算元和運算子的值後，存回運算元堆疊。

Step2:最後取出運算元堆疊的內容，就是後序運算式的計算結果。

3. 前序式 (prefix)：運算子在運算元的前面，例如： $+ A B$

運算式轉換可分為中序轉前序和中序轉後序表示法，這兩種運算式轉換的步驟十分相似，其差異在運算子是放在運算元之前或之後。

中序轉後序演算法： $(A + B) * C - D / E$ 轉後序式

第一步：依照運算子的優先順序，將每個運算子和相關的運算元用括號括起來

$((A + B) * C) - D / E$ (對 $*$ 號加括號)

$((A + B) * C) - (D / E)$ (對 $/$ 號加括號)

$(((A + B) * C) - (D / E))$ (對 $-$ 號加括號)

第二步：對每個運算子找到在它右邊最接近而且未配對的右括號，加上箭頭

$((((A + B) * C) - (D / E)))$

第三步：將運算子移到箭頭所指到的地方，並且去掉所有的括號

$A B + C * D E / -$

堆疊與傳遞有異曲同工之妙，因為遞迴方法是直接使用作業系統的堆

疊，當然也可以自行在程式建立堆疊執行回溯控制，換句話說，走迷宮問題 (A Mazing Problem) 可使用堆疊或遞迴方式來找出迷宮的路。河內塔 (Tower of Hanoi) 問題是說明遞迴觀念時，不可錯過的一個重要範例，在系統中有很詳盡的介紹。

河內塔問題的演算法：對於未知 n 值的河內塔問題需要使用遞迴。首先河內塔問題分解成小問題，並且實際搬搬看，最後可以歸納出三個步驟，如下所示：

Step 1: 將最上面 $n-1$ 個盤子從木樁 1 搬移到木樁 2。

Step 1: 將最後一個盤子從木樁 1 搬移到木樁 3。

Step 1: 將木樁 2 的 $n-1$ 個盤子從搬移到木樁 3。

事實上，河內塔問題就是在解決上述三個步驟，任何子問題只是 n 值的不同，終止條件是 n 等於 1。

三、佇列 (QUEUE S)：佇列是一種和堆疊極度相似的資料結構，用途如同資料儲存之緩衝區，例：使用其作為生產者和消費者模式的資料緩衝區，乃為有序串列，另佇列有以下兩大特性：其中所有的插入發生在串列的固定一端，所有的刪除發生在串列的另一端，插入的一端稱為後端 (Rear)；刪除的一端稱為前端 (Front)，從佇列的一端存入資料，從另一端讀取資料。資料存取的順序是先進先出 (First In, First Out)，也就是先存入佇列的資料，先行取出。

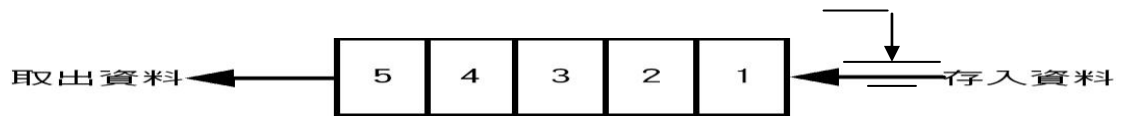


圖 13-1 佇列結構圖

佇列和堆疊同樣可使用鏈結串列實作佇列，使用串列建立的佇列不需檢查佇列是否已滿，因除非電腦記憶體不足，否則佇列並不會全滿。

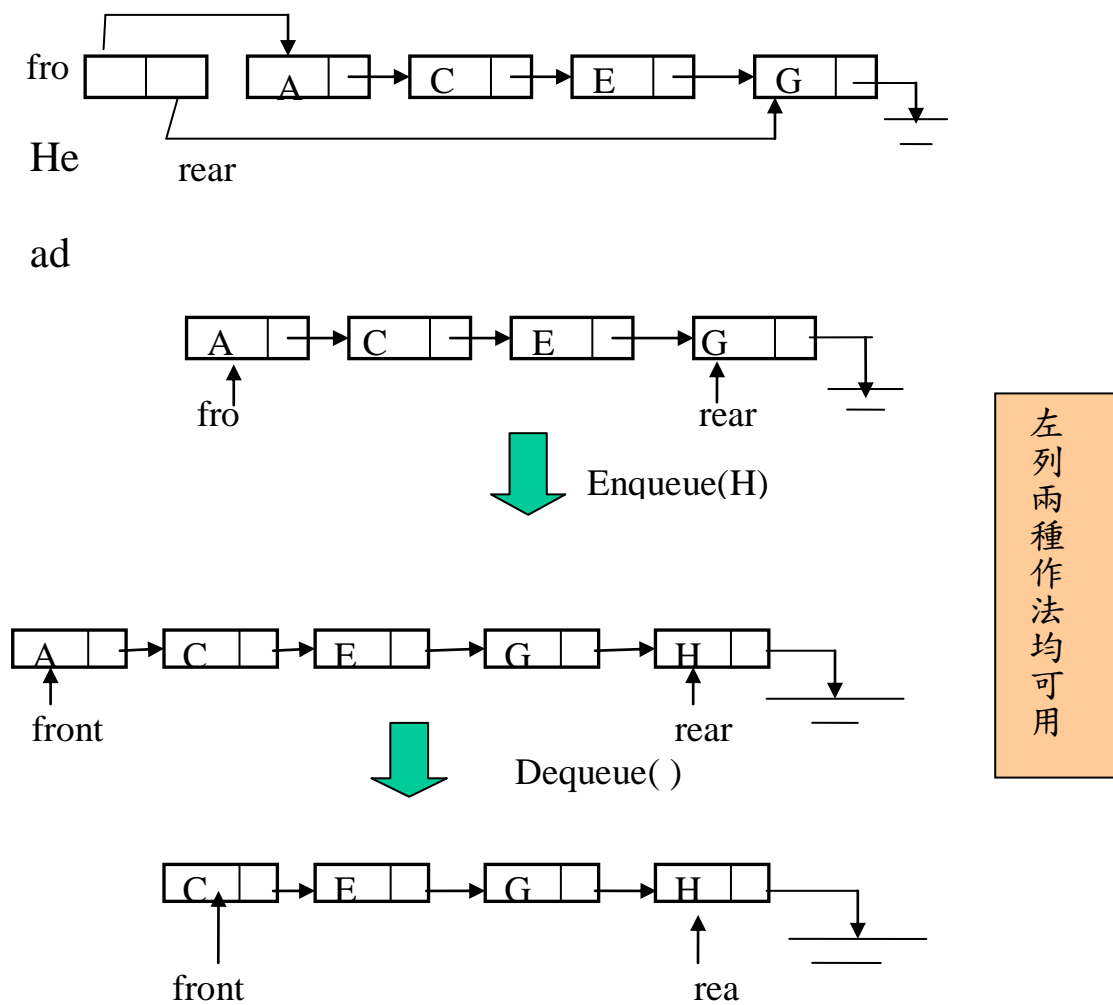


圖 13-2 鏈結串列實作佇列圖

環狀佇列 (Circular Queue)：也是使用一維陣列[]實作的有限元

素數佇列，其差異只在使用特殊技巧來處理陣列索引值，將陣列視為一個環狀結構，佇列的索引指標周而復始的在陣列中環狀的移動，如下圖所示：

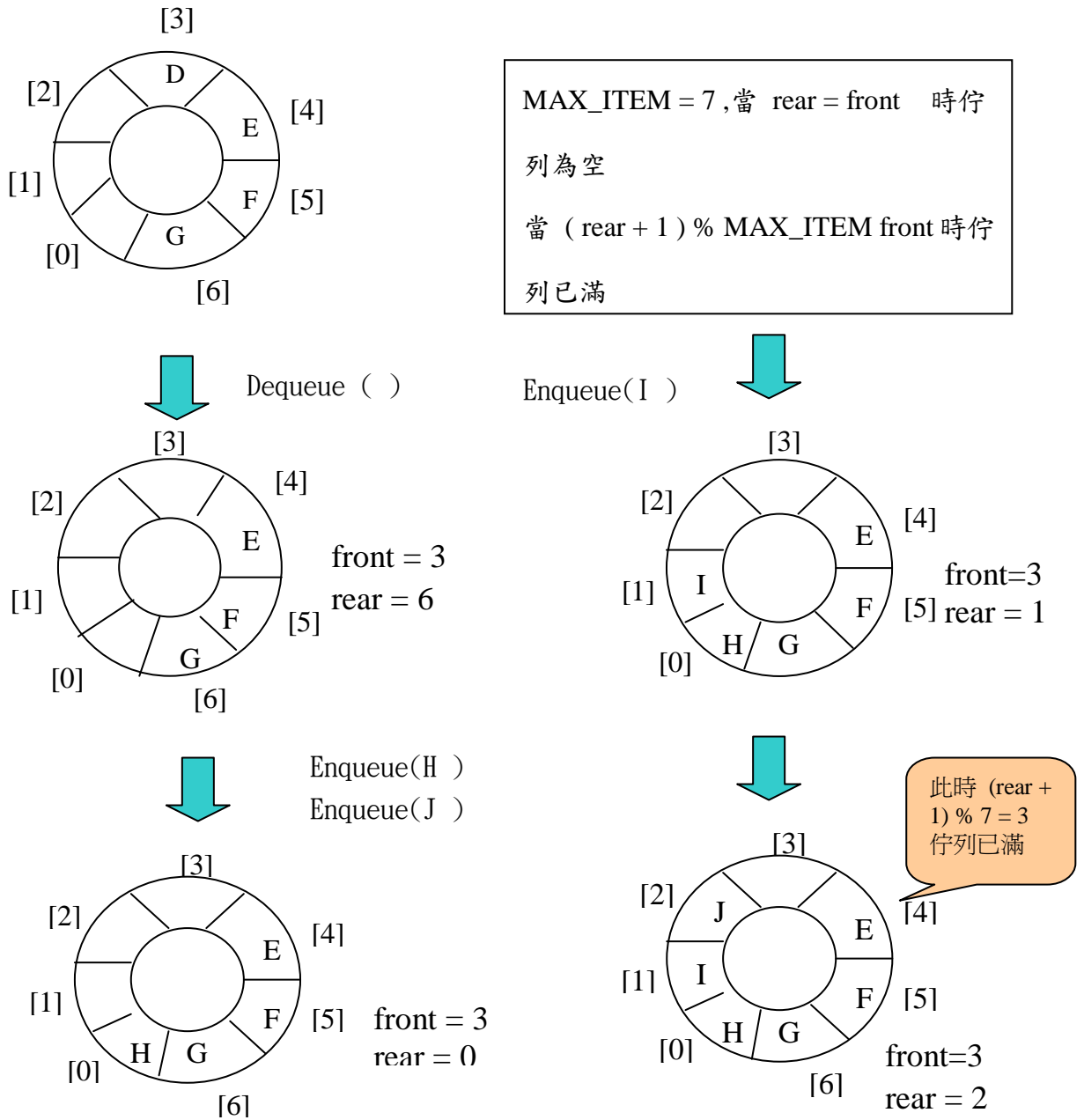


圖 13-3 環狀佇列圖

環狀佇列的性質整理如下：

1. rear 指向排尾，front 指向排頭的前一個空位。
2. 當 $\text{rear} = \text{front}$ 時為空佇列。
3. 當 $(\text{rear} + 1) \% \text{MAX_ITEM} = \text{front}$ 時（當 rear 順時針推進一格會碰到 front 時）為滿佇列。

四、樹(TREE): 樹狀結構乃是模仿真實樹木的架構，只是將它倒立顯示，樹根稱為「根節點」(Root)，是一個(或以上)的節點所形成的集合，其餘節點在根節點下如同樹的樹枝一般，可以擁有 0 到 n 個「子節點」(Children)。

Binary Trees 定義：二元樹屬於樹狀結構的一種特例最多只有 2 個子節點，可以是一棵空的樹(Empty Tree)或者有一個樹根及兩棵子樹，分別稱為左子樹與右子樹。左右子樹也都是二元樹。我們將以

FLASH 動畫方式，簡單介紹二元樹的基本架構)。

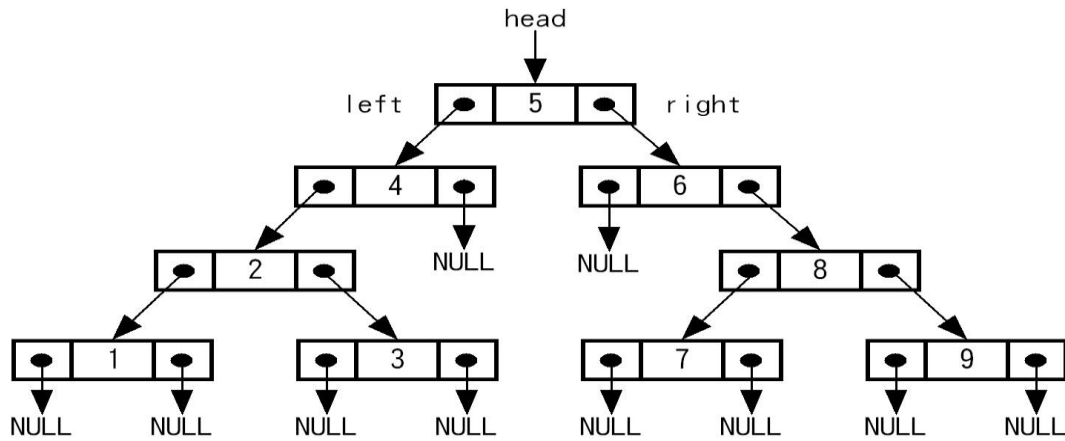


圖 14-1 二元樹例圖

二元樹的走訪可以顯示所有節點資料，屬於一種遞迴走訪方式，因為遞迴方法中遞迴呼叫的排列順序不同，可以分為三種走訪方式，如下所示：

中序走訪方式 (Inorder Traversal)。先以中序順序拜訪左子樹，再拜訪目前的節點。再以中序順序拜訪右子樹。

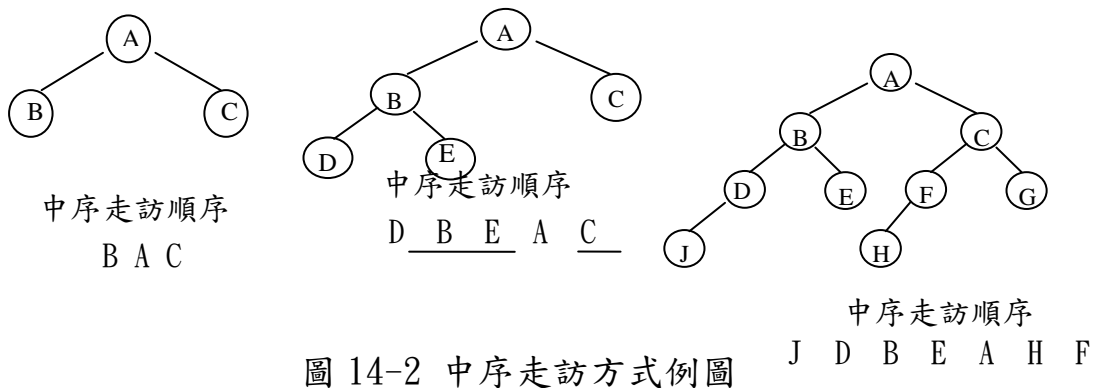


圖 14-2 中序走訪方式例圖

Step1: 檢查是否可以繼續前進，即指標 ptr 不等於 null。

Step2: 然後找到最後一個節點 previous 且將其指標指向新節點。

如果可以前進，其處理方式如下所示：

- 1) 遞迴呼叫 inOrder(ptr. left) 向左走。
- 2) 處理目前的節點，顯示節點資料。
- 3) 遞迴呼叫 inOrder(ptr. right) 向右走。

前序走訪方式 (Preorder Traversal)

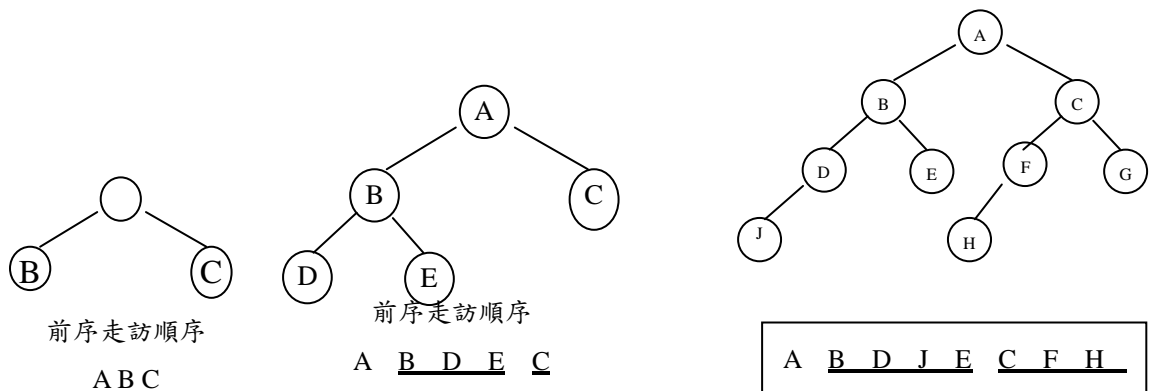


圖 14-3 前序走訪方式例圖

後序走訪方式 (Postorder Traversal)

先以後序順序拜訪左子樹，再以後序順序拜訪右子樹，再拜訪目前的節點。

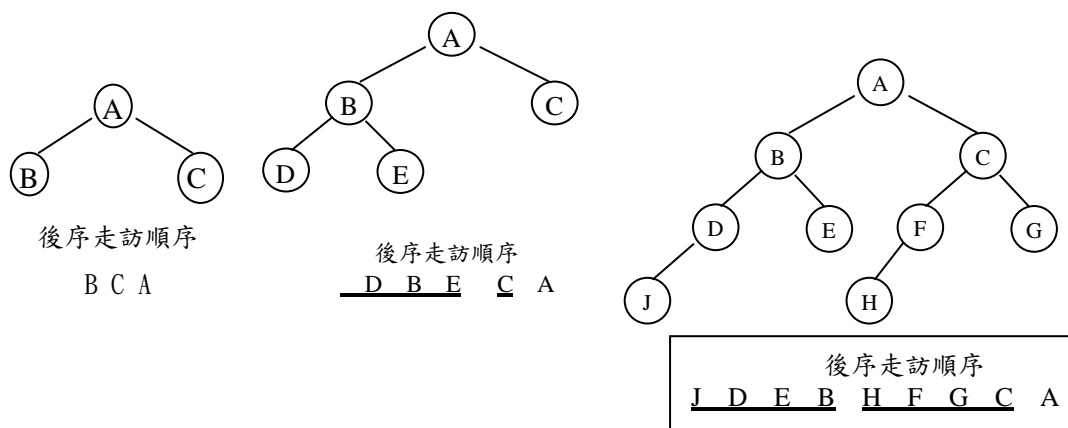


圖 14-4 後序走訪方式例圖

Step1:檢查是否已經到達葉節點，即指標 ptr 等於 null。

Step2:如果不是葉節點表示可以繼續走，其處理方式如下所示。

- 1) 遞迴呼叫 postOrder(ptr. left)向左走。
- 2) 遞迴呼叫 postOrder(ptr. right)向右走。
- 3) 處理目前的節點，顯示節點資料。

二元搜尋樹(Binary Search Trees)

1. 二元樹的每一節點值都不同，整棵二元樹的每一節點皆擁有不同值。
2. 每一節點的資料大於左子節點的資料，但小於右子節點的資料。
3. 節點的左、右子樹也是一棵二元搜尋樹。

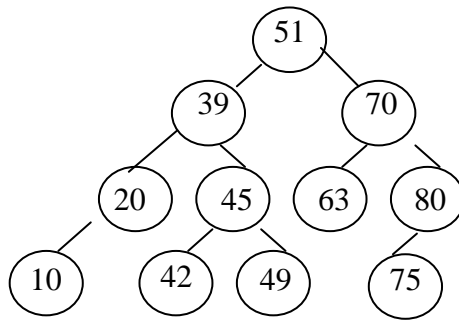


圖 14-5 二元搜尋樹例圖

五、圖形(GRAPH): 圖形是資料結構中一種十分重要的結構，是由有限的 V : 頂點(vertices)和 E : 邊線(edges)二個集合所組成以 $G=(V, E)$ 表示。圖形共分為兩類：無方向性圖形(undirected graph)，在無方向性圖形中，邊線無標示方向的箭頭，邊線只代表頂點是相連的，因此 $(v1, v2)$ 和 $(v2, v1)$ 這兩個頂點對代表同一個邊。

有向圖形(directed graph): 在方向性圖形中的邊線加上箭號標示頂點間的順序性，每一個邊用一個有序對 $\langle v1, v2 \rangle$ 表示， $v1$ 是該邊的尾部(tail)，而 $v2$ 是該邊的頭部(head)。因此 $\langle v1, v2 \rangle$ 與 $\langle v2, v1 \rangle$ 代表兩個不同的邊。圖形是由有限個節點組合而成，節點與節點之間是由邊相互連接，對於每個節點之拜訪順序，擁有特定的走訪方式也就是圖形之搜尋法。

二種搜尋法：

深度優先搜尋法(Depth First Search, DFS)：當在鄰接串列表表示法走訪某一頂點後，優先找尋頂點在物件陣列中的鄰接頂點。

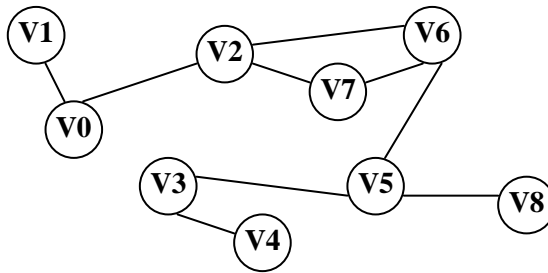


圖 15-1 深度優先搜尋法例圖

如果以 V0 為起點，其中一組 DFS 拜訪順序為：

V0, V1, V2, V6, V7, V5, V8, V3, V4

從起點 V_s 開始，接著選擇和 V_s 相鄰的任一頂點 V_x ，並由 V_x 繼續做深度優先拜訪。當到達某個頂點 V_u 時，若 V_u 所有相鄰的頂點都已經被拜訪過，而無法繼續前進深入時，則退回到 V_u 的上個拜訪頂點，繼續作深度優先拜訪。

寬度優先搜尋法(Breadth First Search, BFS)：當在鄰接串列表表示法走訪某一頂點後，優先找尋頂點串列的所有鄰接頂點。

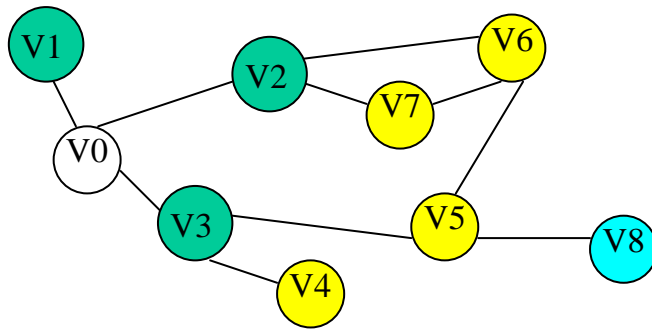


圖 15-2 寬度優先搜尋法例圖

如果以 V0 為起點，其中一組 BFS 拜訪順序為：

V0, V1, V2, V3, V6, V7, V5, V4, V8

第一層

第二層

第三層

從起點 V0 開始，拜訪和起點 V0 相鄰的所有頂點。再拜訪更外一層的頂點，也就是與起點 V0 相鄰頂點的相鄰頂點。直到所有連通頂點都拜訪過為止。

最小成本擴張樹(Minimum Cost Spanning Tree)

擴張樹結構：將圖形走訪過的頂點順序，使用邊線一一連接起來，就建立成擴張樹。擴張樹在實際的應用上不止是找出頂點和邊而已，如果一個相連圖形的邊加上權重值 (weight)，來代表邊的成本、距離等。則我們希望所產生的擴張樹之所有邊的權重值加總為最小，具有

這樣性質的擴張樹稱為最小成本擴張樹 (minimum-cost spanning tree)。

以最少成本為原則，須滿足下列限制：

只能使用這個圖裡的邊。

- 只能使用 $n-1$ 個邊(若有 n 個節點)。
- 所使用的邊不能產生一個環路。

圖形的應用 - 最短路徑：最短路徑問題 (The Shortest Path Problem) 花費最少擴張樹之總花費是連繫網路中每一個頂點所須的最少花費，但連繫樹中任兩頂點的路徑不一定是一條花費最少的路徑。其中一個頂點對多頂點的最短路徑。Dijkstra 演算法是我們會介紹到的。例如：從基隆到高雄，中間經過台北、新竹、台中或桃園，各城市間的距離，如下圖所示：

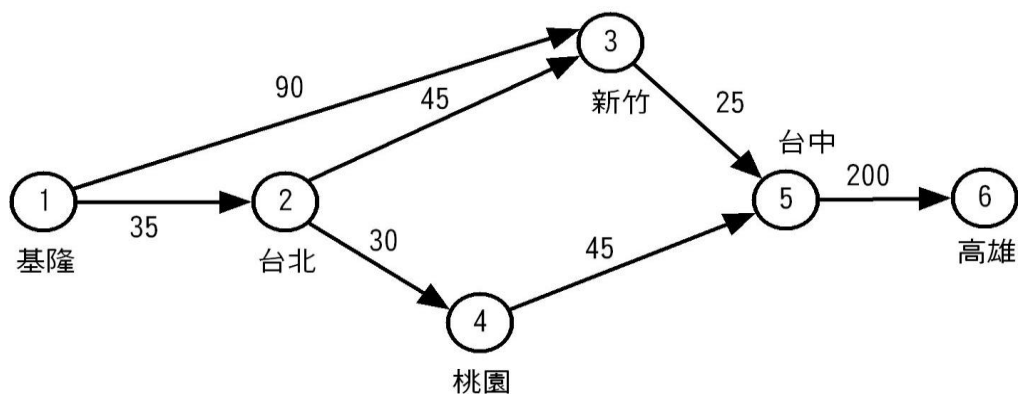


圖 15-3 基隆市與各都市間矩陣表示圖

而 頂點 1 基隆到各頂點城市間的距離，很明顯的！可以發現基隆到新竹的最短路徑，也是基隆到台中或高雄最短路徑的一部分，也就是說，已知的最短路徑頂點，可能就是其它頂點最短路徑上經過的頂點，如下表所示：

開始頂點	結束頂點	路徑	距離
基隆-V1	台北-V2	1, 2	35
基隆-V1	新竹-V3	1, 3	90
基隆-V1	新竹-V3	1, 2, 3	$35+45 = 80$
基隆-V1	桃園-V4	1, 2, 4	$35+30 = 65$
基隆-V1	台中-V5	1, 3, 5	$90+25 = 115$
基隆-V1	台中-V5	1, 2, 3, 5	$35+45+25 = 105$
基隆-V1	台中-V5	1, 2, 4, 5	$35+30+45 = 110$
基隆-V1	高雄-V6	1, 2, 3, 5, 6	$35+45+25+200 = 305$
基隆-V1	高雄-V6	1, 2, 4, 5, 6	$35+30+45+200 = 310$

表 1 基隆到各頂點城市間的距離表

Dijkstra 演算法是依據上述最短路徑的特性，提出解決「單來源最短路徑的問題」(Single-source Shortest Paths)的演算法，Dijkstra 演算法使用鄰接矩陣表示法建立圖形。

Dijkstra 演算法的完整操作步驟，如下所示：

- Step 1：初始相關陣列的內容：
 - (1) 將 $graph[source][i]$ 來源頂點複製到一維陣列 $dist[i]$ 。
 - (2) 設為 $selected[i]$ 和 $pi[i]$ 的陣列元素初值。
- Step 2：執行頂點總數減 1 次的迴圈來計算最短路徑，如下所示：
 - (1) 走訪陣列 $dist[]$ 找出最短距離的頂點 W ，且此頂點沒有選過。
 - (2) 將頂點 W 設為選過，即 $selected[W] = 1$ 。
 - (3) 走訪陣列 $dist[]$ ，如果有未選過的頂點 X ，則：
 - 1) 比較 $dist[X]$ 和 $graph[W][X]+dist[W]$ 的距離大小：
 - a. 如果 $graph[W][X]+dist[W]$ 小，存入 $dist[X]$ 。

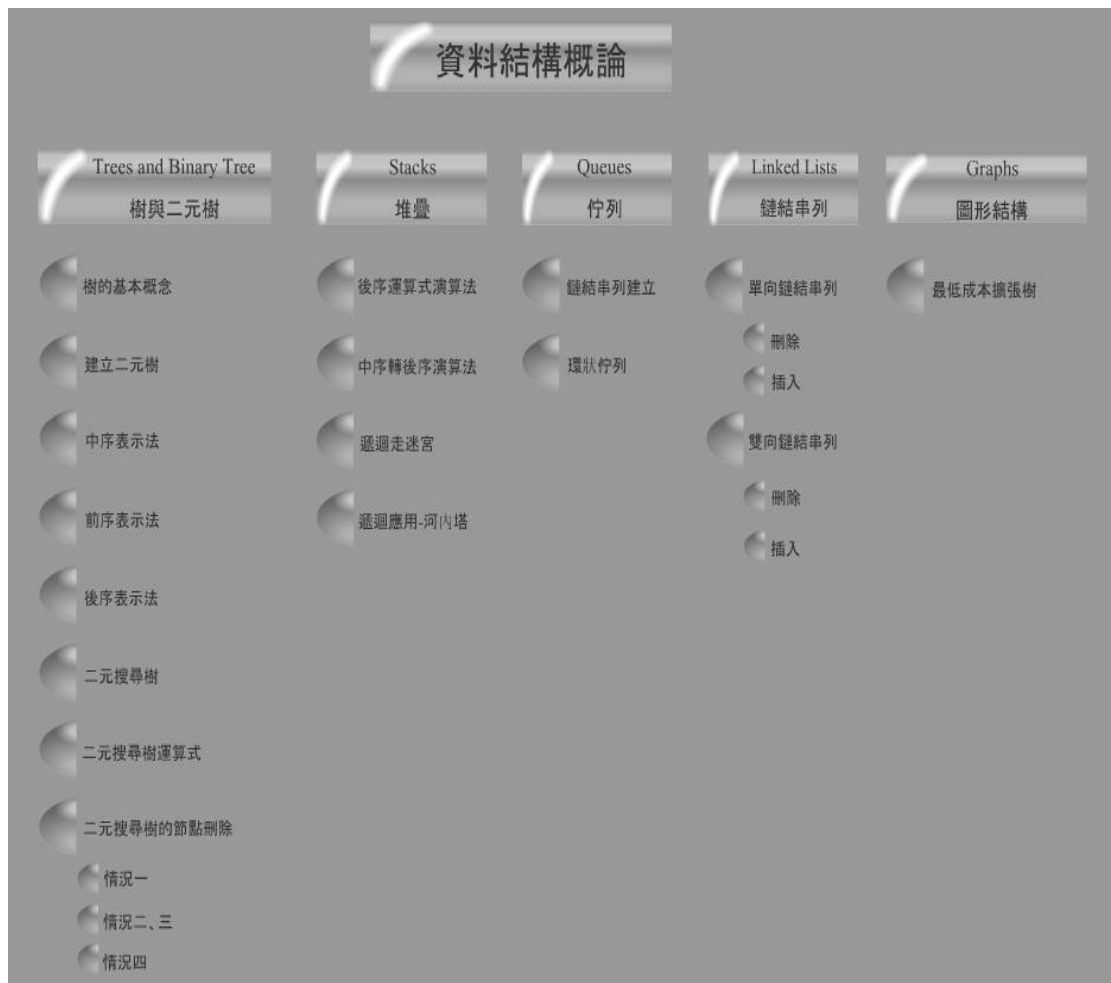
- b. 如果更改距離，指定頂點 X 的前頂點為 W ， $pi[X] = W$ 。

利用 Dijkstra 演算法能找到單一來源頂點到其它頂點的最短路徑。

以上皆是我們在系統中會介紹給大家的資料結構的理論與觀念。

第三章 FLASH 動畫與資料結構輔助教學應用

第一節 系統架構與功能簡介



系統架構圖與功能簡介逐項說明：

圖 16-1 首頁畫面呈現

這主要是提供給一般學生所使用的系統，最主要的功能是「資料結構概論」教學內容及動畫呈現，為最主要的教學內容課程；「參考

資料」即我們在架構成立此系統時所參考之資料來源連結或介紹；「即時留言」可以連結到我們所共用信箱，在此提出問題及回答問題。在教材部份，基本概念方面介紹了資料結構基本概念、包含鏈結串列、堆疊、佇列、樹、圖形。五大類型。選擇進入各項分類教學將有一連串以 FLASH 製作成之資料以供學習，讓使用者知道資料結構、程式寫法等訊息，其內容如堆疊基本原理架構、資料加入動作原理、資料取出動作原理、加入動作程式說明、取出動作程式說明、…等等，並且有靜態、動態之分。希望藉由這些內容來讓學生知道他所學的是什麼，以及一些應該先了解清楚的基本觀念。

我們利用圖形再搭配文字上的說明，以動畫的表現方式來讓學生能夠清楚地明白，在各種資料結構中如何去做到建立、刪除、插入…等等的動作。學習資料結構應該從最基本的陣列結構的使用及應用在各種問題上開始；再來是演算法的分析方法及簡單的練習；再進入最常用的搜尋演算法和排序演算法。

有關於鏈結串列，說明鏈結串列的結構內容，以及在程式中我們該如何去宣告、如何去取得節點的記憶體空間，也就是環狀鏈結串列中，我們利用圖形再搭配文字上的說明，以動畫的表現方式來讓使用者能夠清楚地明白，在鏈結串列這種資料結構中如何去做到建立、刪除、插入等等的動作。在刪除動作時，由於刪除不同的節點，會有不

同的步驟及結果，一步步的程式流程都是為了滿足這些鏈結串列的各種動作，以加強學生對此課程的理解。

堆疊(Stack)則是一種特殊的有序串列，它限制所有資料的加入與取出（或刪除）僅在串列的一端進行，而此端稱為頂端（Top）；好比一個圓筒容器，只能在此容器固定的一端放入或取出物件，如同堆疊運算上暗示了如果我們依序將 A, B, C, D, E 加入堆疊中，則 E 一定是第一個自堆疊中取出的元素。我們可以分別利用加入（push）和取出（pop）兩動作來存取資料，教材課程這部份，全都是利用 Flash 來製作，這也是因為 Flash 具有強大的動畫功能，非常合乎我們所要達成的圖形顯示結果。

佇列中則簡單介紹了環狀佇列：我們以鏈結串的方法來表示環狀佇列，而我們要到底要怎樣才能以鏈結串列來建立環狀佇列呢？其實方法很簡單，首先我們先建立好一個基本的鏈結串列，接著再把最後一個節點指向第一個節點就可以了。而現在所建立起來的環狀佇列我們稱之為環狀鏈結串列，這是一般佇列的演進。

樹則是介紹了二元樹以及二元樹鍵結表示法，更引申中序、後序、前序三種表示法，對於二元搜尋樹也有完整的教學。

圖形結構中介紹了圖形的走訪：深度優先搜尋法 D F S、寬度優先搜尋法 B F S。在圖形的應用—最短路徑中也介紹了 Dijkstra 演

算法。

第二節 系統操作流程

使用者可以直接進入我們的課程內容中，在「資料結構概論」中有「基本概念」、「串列」、「堆疊」、「佇列」、「樹」等單元。

學生可以自由選擇其中的單元學習，學生可以就學習過程中所遇到的問題，在「即時留言」區中提出問題，讓老師或其他的學生回應。相互討論，有雙向互動能更深刻去了解資料結構。

第四章 系統呈現

第一節 系統畫面

本教學影片以 FLASH 技術運用於資料結構的程式儲存方法與步驟，以畫面方式詳盡的將陣列、串列、序列、堆疊、二元樹和圖形等各種結構，與程式碼演算，讓已有相關程式設計基礎的使用者學習。堆疊的應用-運算式的計算與轉換:堆疊最常應用的問題是運算式的計算與轉換，可以使用堆疊來儲存運算元或運算子，來計算前序、中序和後序運算式的結果，或是進行運算式的轉換。

簡單介紹以下幾種演算法:

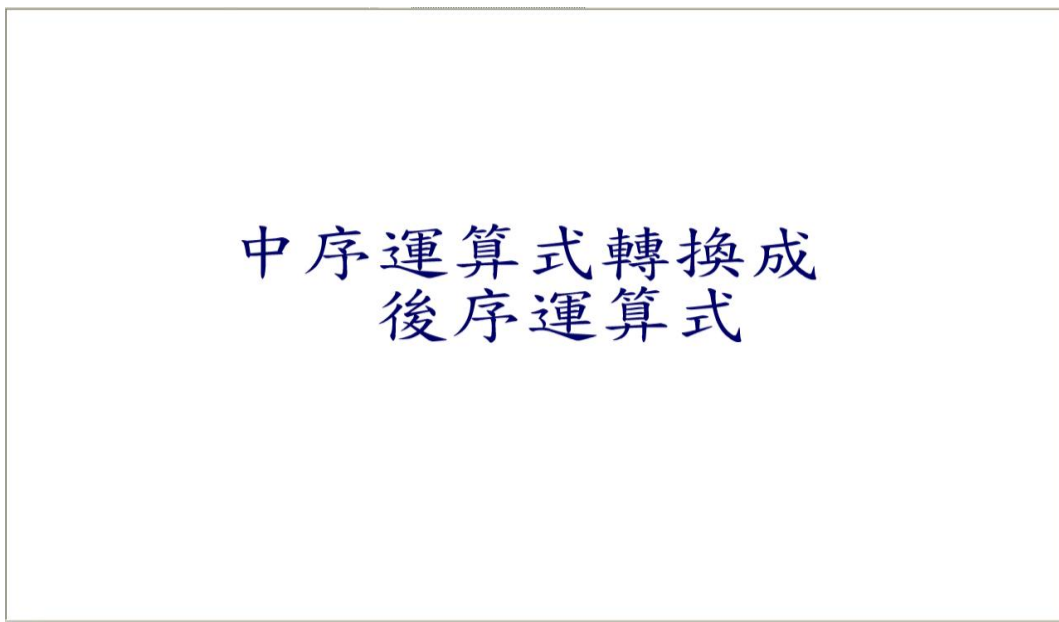


圖 17-1 中序運算式轉換成後序運算式首頁圖

Step1：使用迴圈讀取中序運算式的運算元和運算子，若：

- (1)讀取的是運算子：
 - 1)如果運算子堆疊是空的或是左括號，存入運算子堆疊。
 - 2)如果是右括號，從堆疊取出運算子輸出，直到左括號為止。
 - 3)如果堆疊不是空的，持續和堆疊的運算子比較優先順序，若：
 - a. 優先順序比較低，輸出運算子。
 - b. 優先順序比較高或堆疊空了，將運算子存入運算子堆疊。
- (2)讀取的是運算元，直接輸出運算元。

圖 17-2 中序轉換成後序演算法簡介圖

中序運算式： $(9+6)*6$ 的計算過程是主迴圈從左到右依序讀取運算式的運算元(子)，第一個讀入的字元「(」首先會存入運算子堆疊。

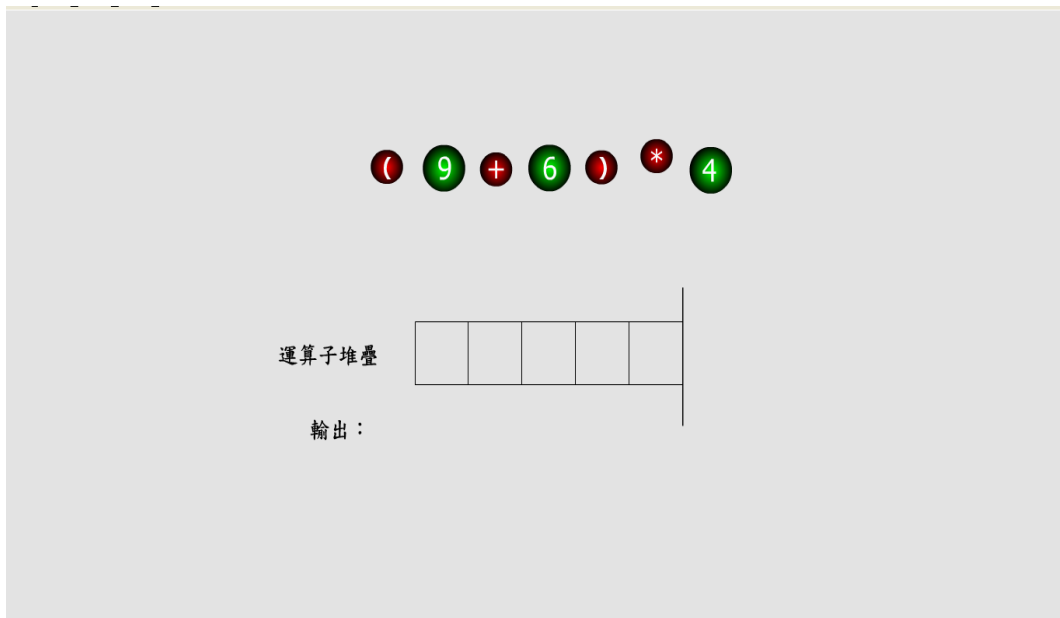


圖 17-3 中序運算式的計算圖示

接著讀入的字元是運算元「9」，直接輸出運算元後再讀入字元「+」是運算子，因非右括號且「+」號的優先順序大於堆疊中的「(」，所以將運算子存入運算子堆疊，再讀出運算元「6」可直接輸出而右括號「)」，我們需從運算子堆疊取出運算子「+」輸出，直到左括號為止，左括號在此的功能則為一個標籤，所以並不用輸出，但此時堆疊已全部空出，如再讀入字元「*」是為運算子則存入運算子堆疊如下：

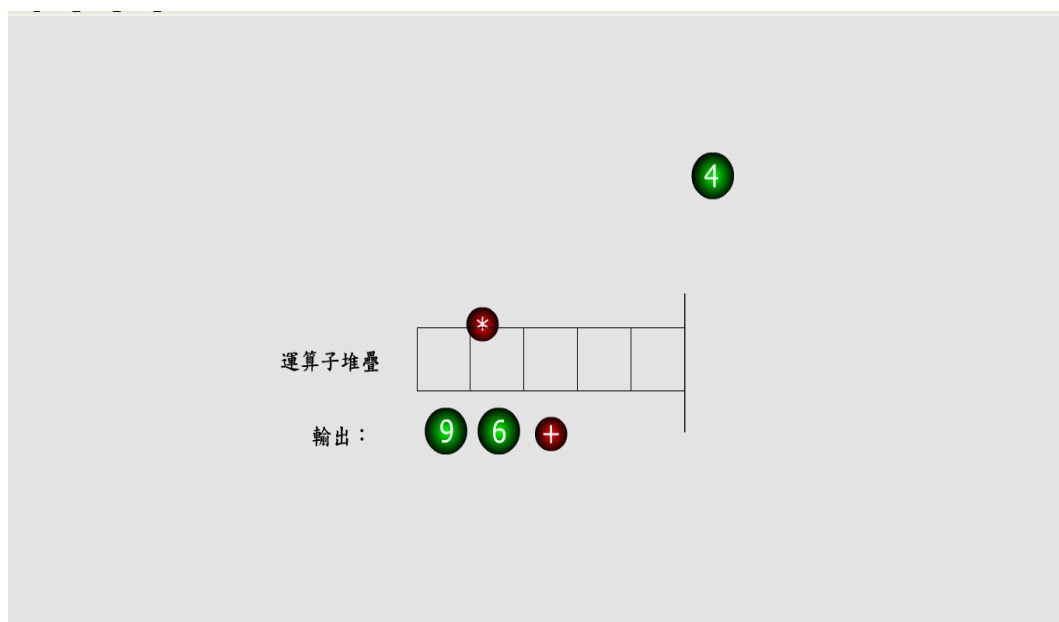


圖 17-4 「*」存入運算子堆疊圖

再接著讀入運算元「4」，直接輸出運算元。現在已經讀出整個中序運算式，接著再從運算子堆疊取出剩下的運算子「*」且輸出，即可得到轉換的後序運算式。

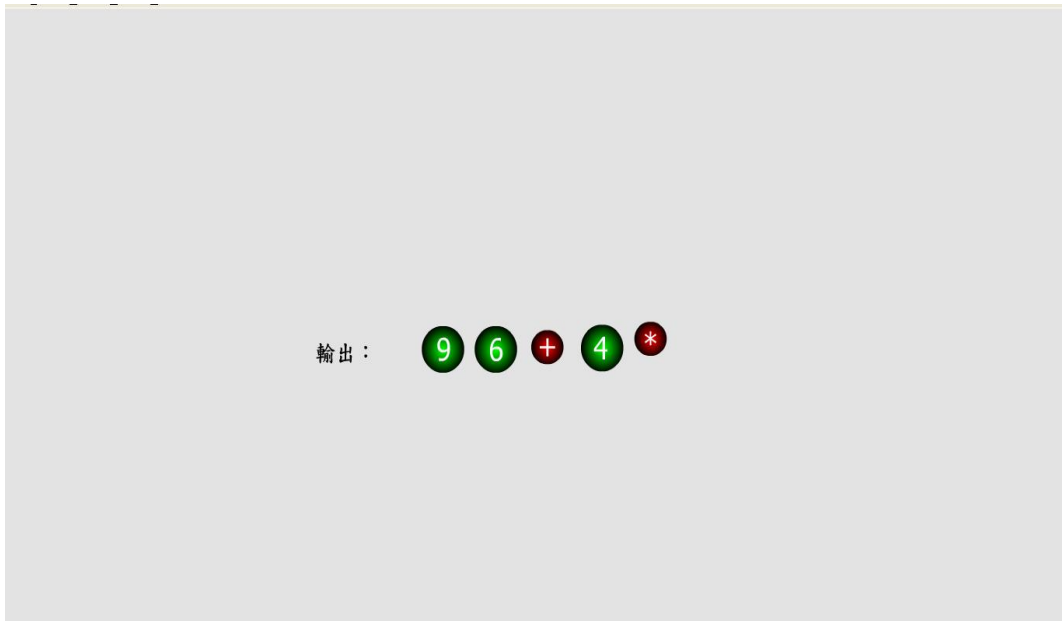


圖 17-5 中序轉後序運算式後所得結果圖

樹的部分以下以有特色的建立二元樹的中序、前序走訪與二元搜尋樹的節點刪除及運算式處理應用為例。



圖 18-1 建立二元樹教學動畫-初始畫面

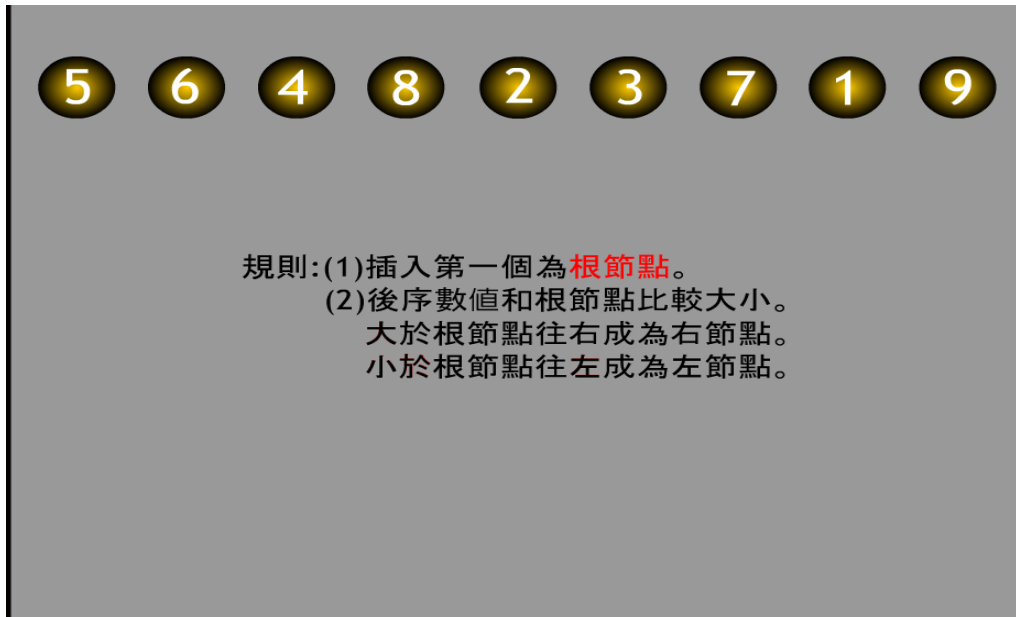


圖 18-2 建立二元樹教學動畫-文字呈現。簡單介紹其規則圖

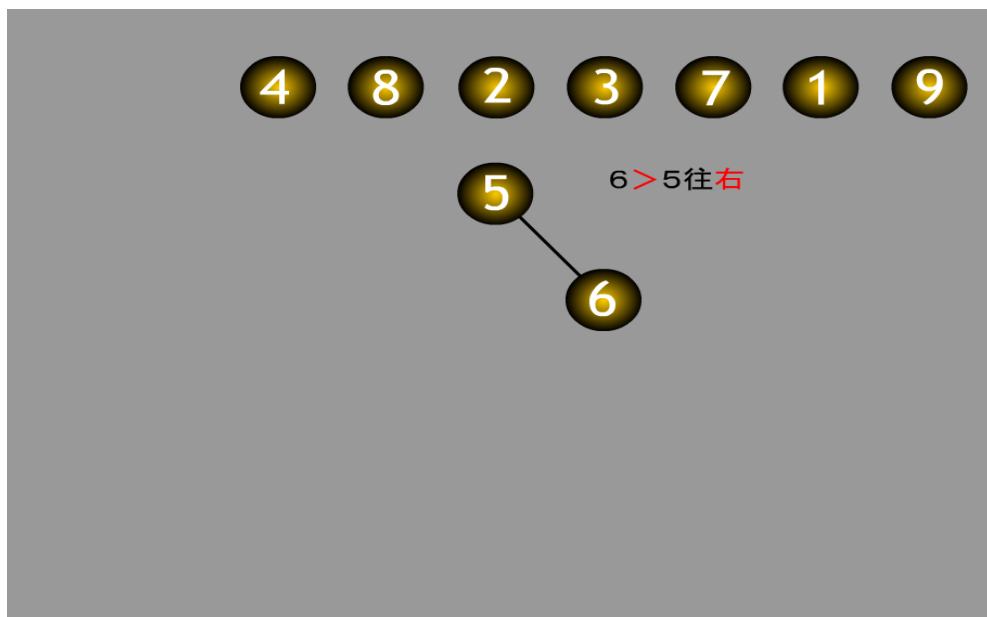


圖 18-3 建立二元樹教學動畫-圖形表示。

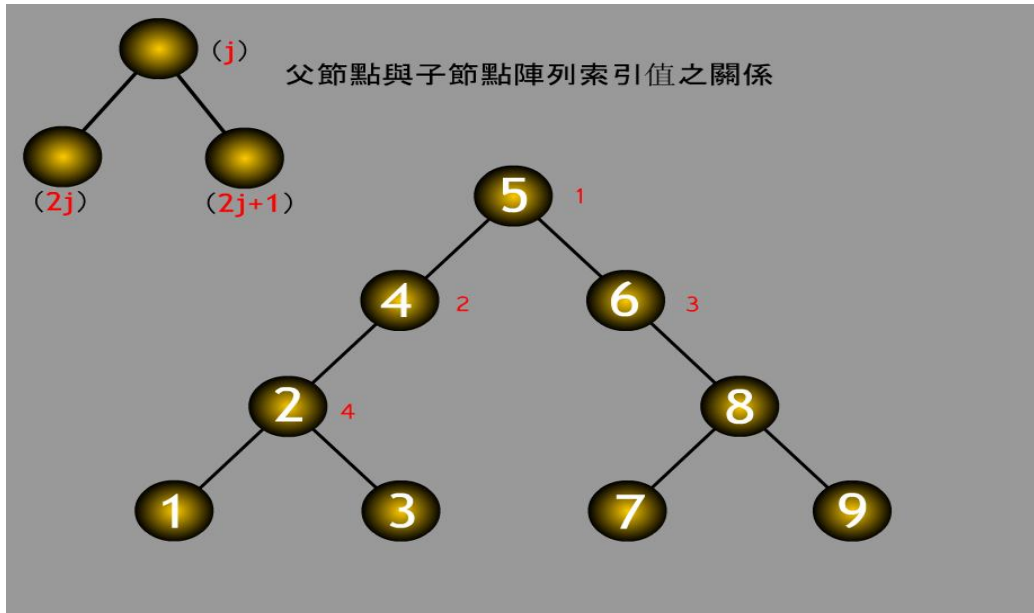


圖 18-4 建立二元樹教學動畫 完滿二元樹架構圖。

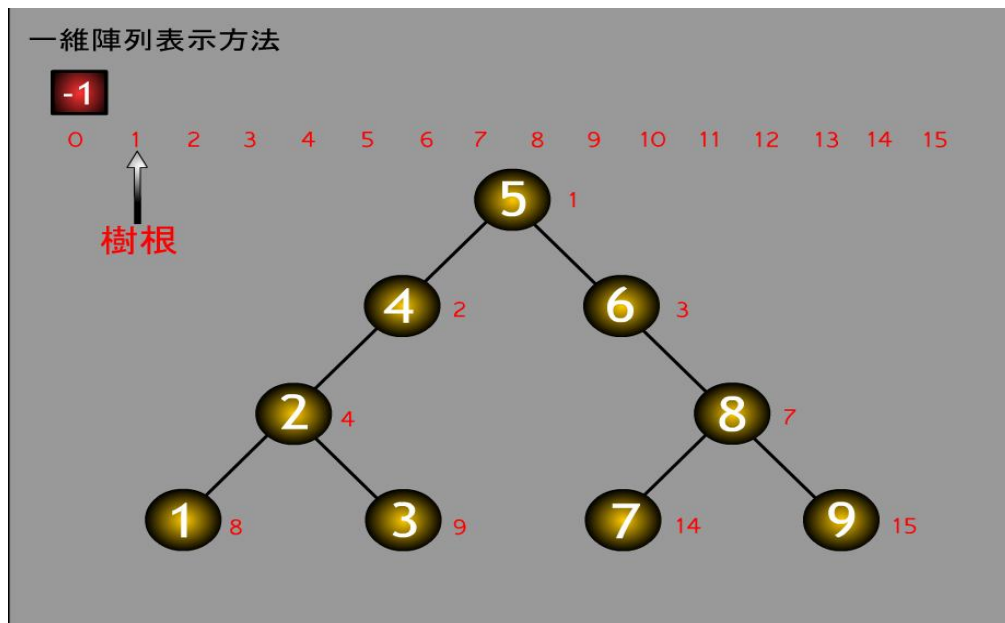


圖 18-5 建立二元樹教學動畫二元樹陣列表示法

一、二元樹的中序走訪方式

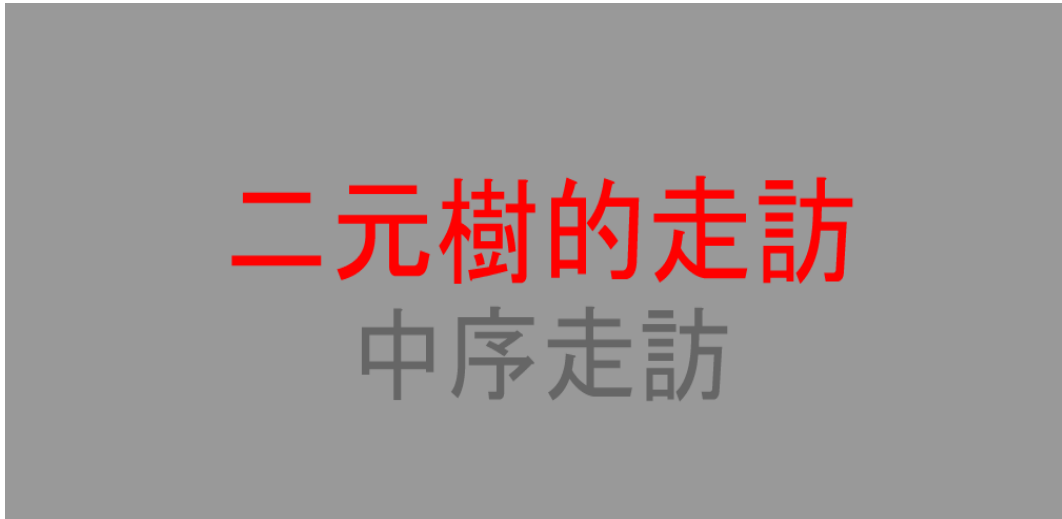


圖 19-1 二元樹的中序走訪教學動畫「初始畫面」。

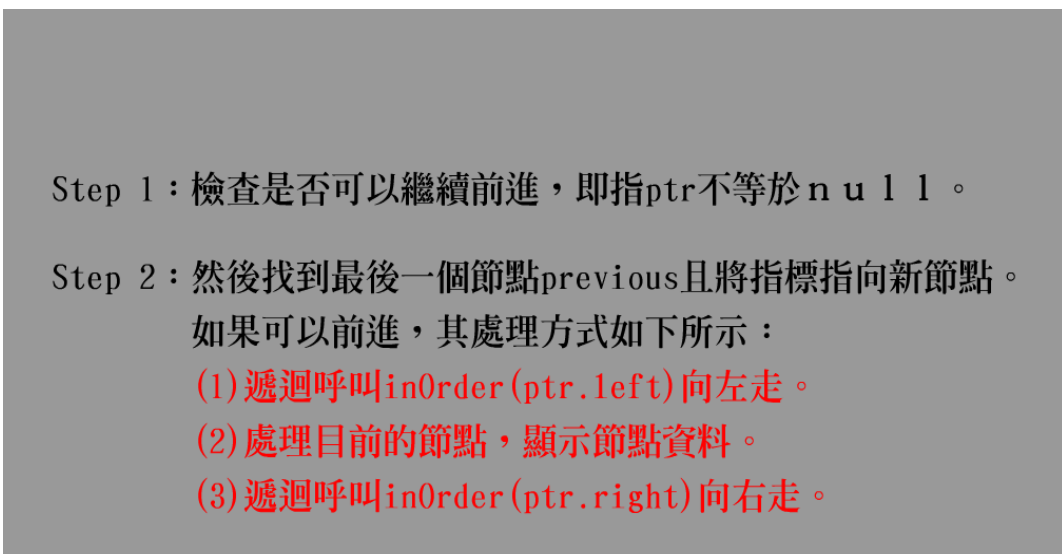


圖 19-2 二元樹的中序走訪演算法「文字呈現」。

沿著二元樹左方開始往下走，無法前進後顯示節點，退回父節點

再往右走，如右邊也無法前進顯示節點，再退回上一層。

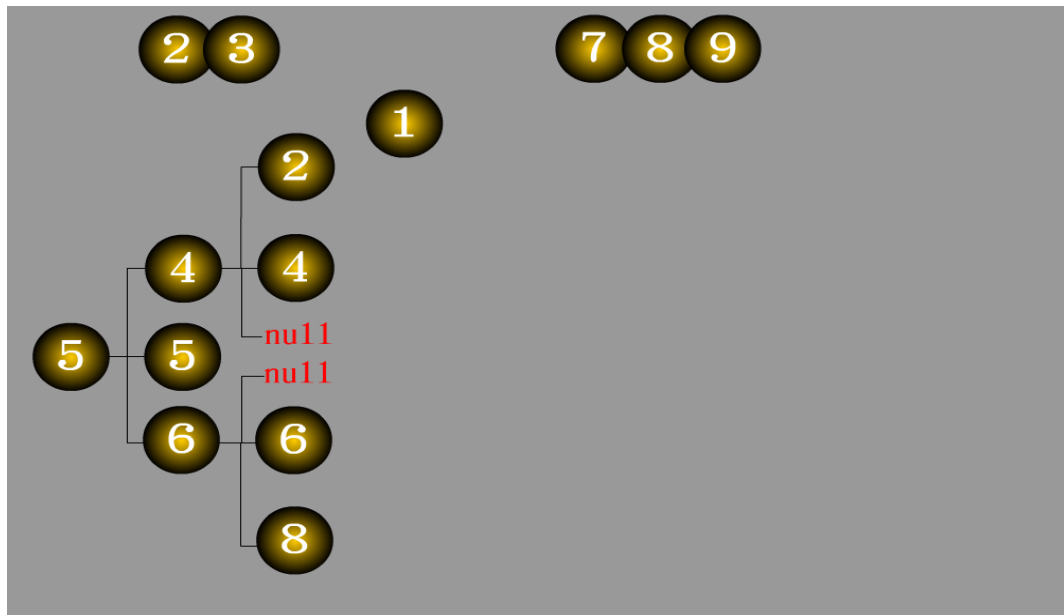


圖 19-3 二元樹的中序走訪「圖形運算」

二、二元樹的前序走訪方式

Step 1：先檢查是否已經到達葉節點，即指ptr等於null。

Step 2：如果不是葉節點表示可以繼續走其處理方式如下所示：

- (1) 處理目前的節點，顯示節點資料。
- (2) 遞迴呼叫preOrder(ptr.left)向左走。
- (3) 遞迴呼叫preOrder(ptr.right)向右走。

圖 20-1 二元樹的前序走訪演算法 文字呈現

前序走訪順序爲：**根節點** > 左子樹 > 右子樹

圖 20-2 二元樹的前序走訪規則 文字呈現

前序走訪方式是走訪到二元樹節點，就立刻顯示節點資料，順序是先向樹的左方走到無法前進後，才轉右方走。

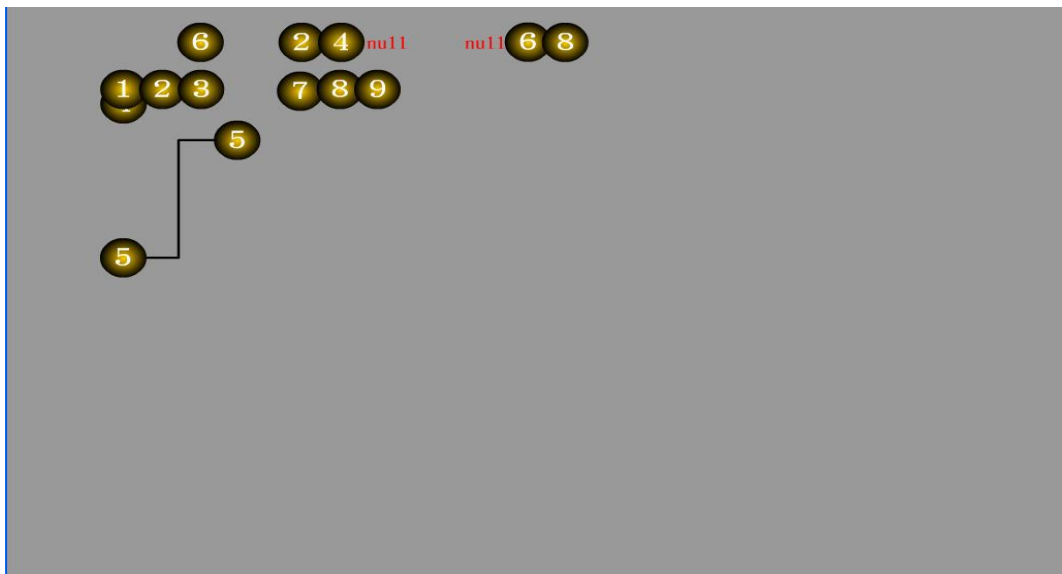


圖 20-3 二元樹的前序走訪教學動畫 圖形運算

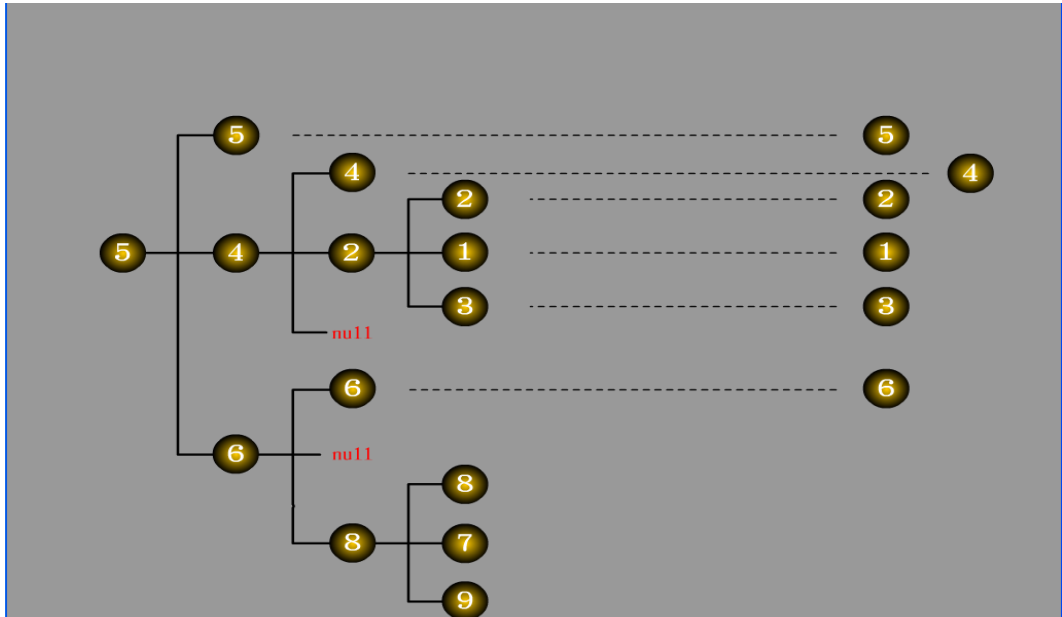


圖 20-4 二元樹的前序走訪教學動畫 圖形運算

三、二元搜尋樹的節點刪除

二元搜尋樹的節點刪除可以分成多種情形，因二元搜尋樹在刪除節點後，仍需要滿足二元搜尋樹的特性，所以節點刪除分成四種情況，如以下各圖所示：

二元搜尋樹的節點刪除

圖 21-1 二元搜尋樹的節點刪除-初始畫面

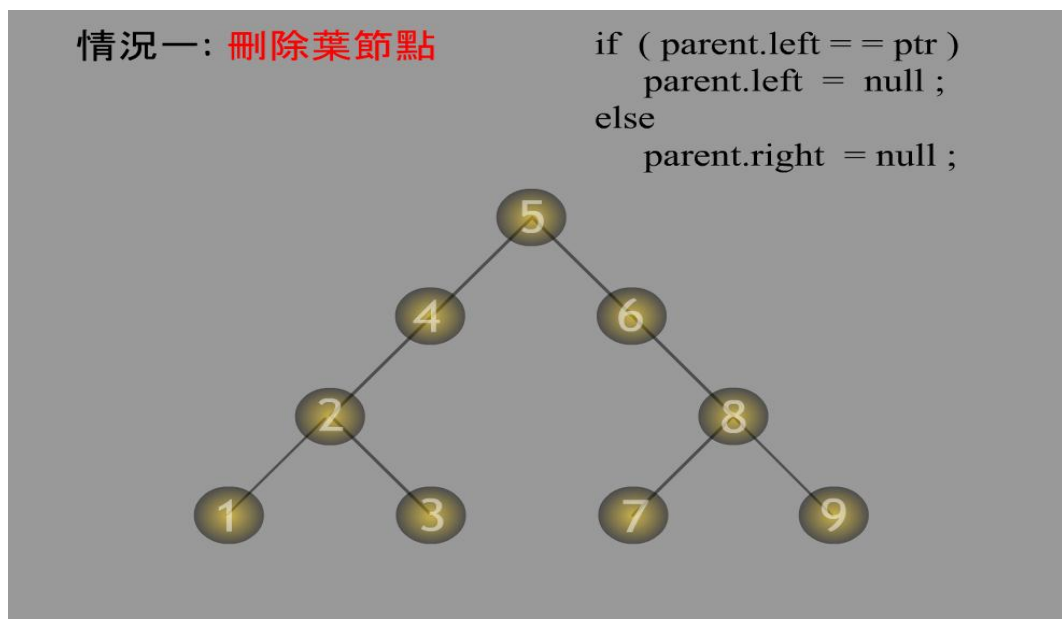


圖 21-2 二元搜尋樹的節點刪除葉節點 圖形表示

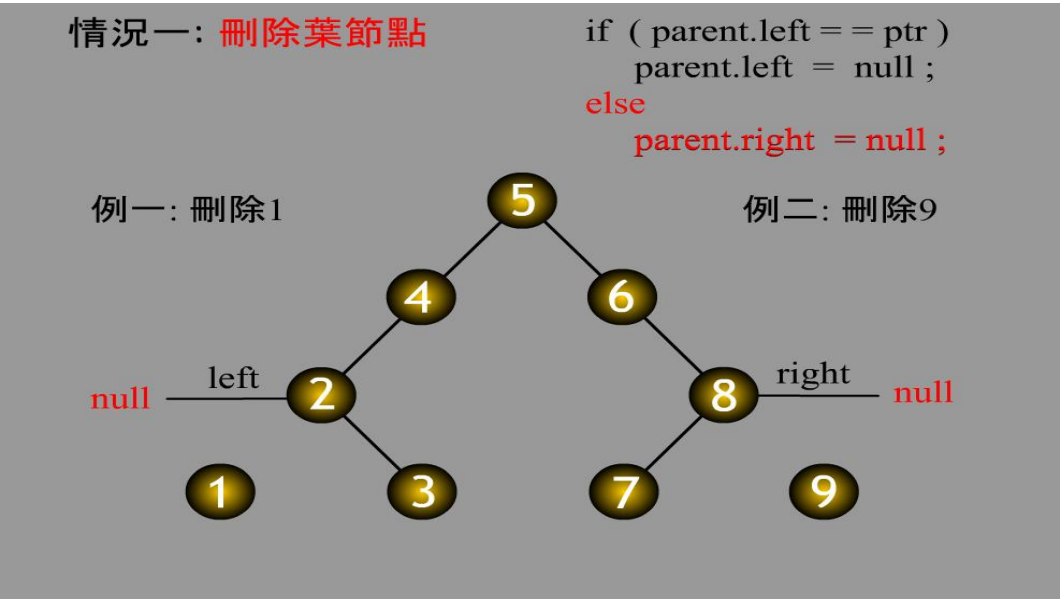


圖 21-3 二元搜尋樹的節點刪除葉節點 圖形表示

若刪除節點沒有左子樹，只需將根節點指標指向其右子樹節點，如下圖所示：

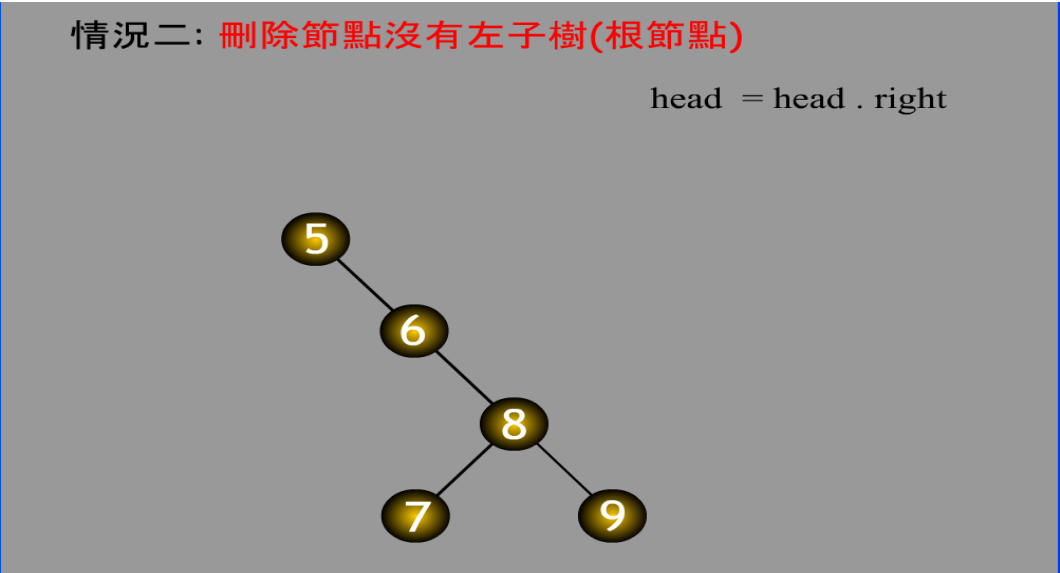


圖 21-4 二元搜尋樹的節點刪除教學動畫 圖形表示

刪除的若是中間節點 2 或是 6 時，因為其無左子樹，此時是將刪除節點的父節點，指向其右子節點即可，如下圖所示：

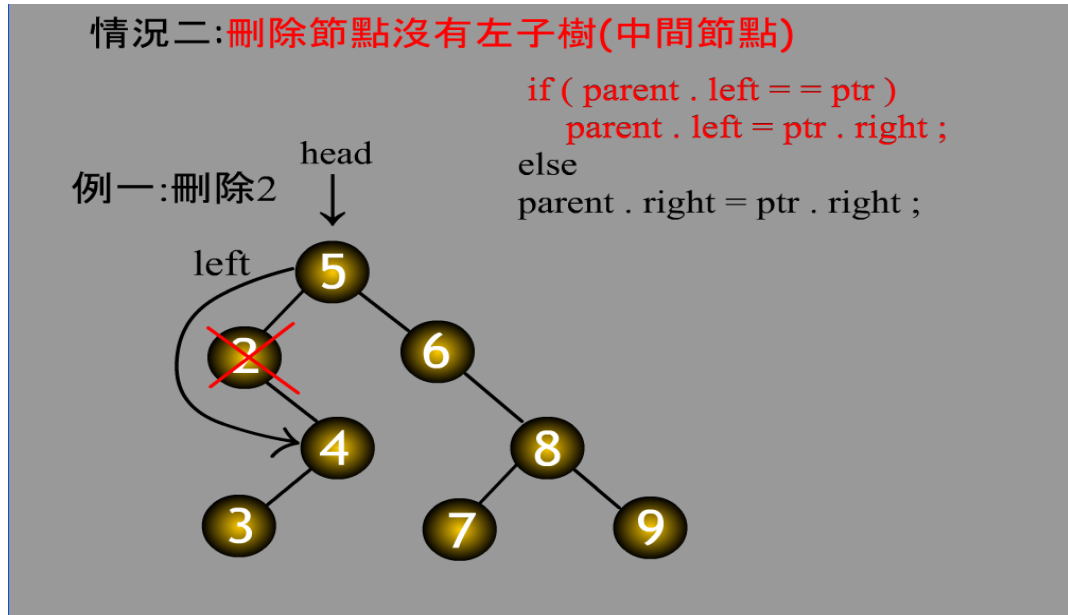


圖 21-5 二元搜尋樹的節點刪除教學動畫 圖形表示

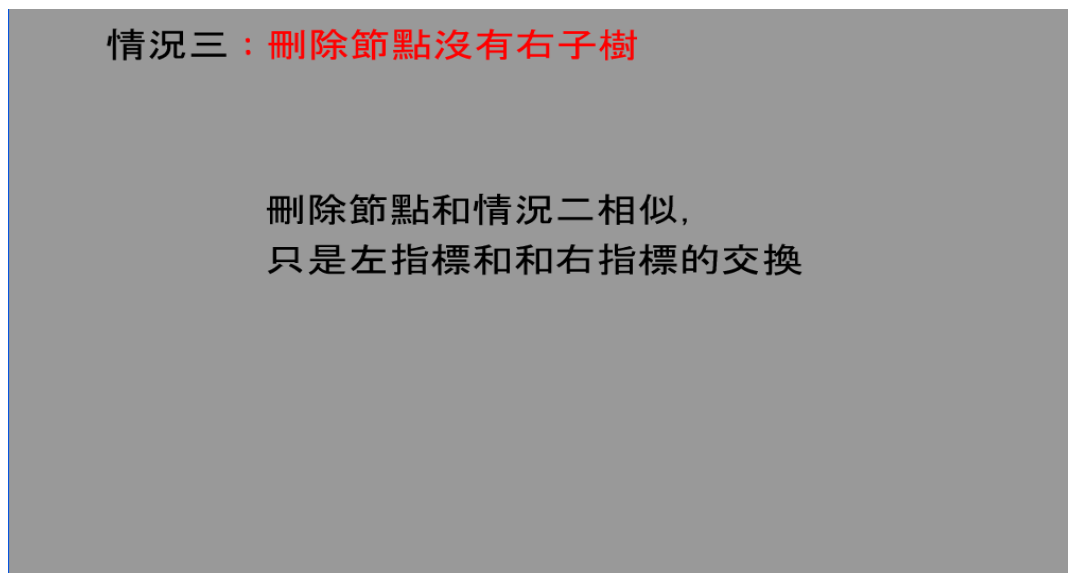


圖 21-6 二元搜尋樹的節點刪除教學動畫-文字呈現

四、二元樹的應用運算式處理教學動畫

從樹的觀念而言，樹可以處理各種階層關係的問題，如將資料建立成二元搜尋樹，就成為一種很好的資料搜尋方法。也可用二元樹也處理運算式。例如：將中序運算式轉換成二元樹，如下所示：

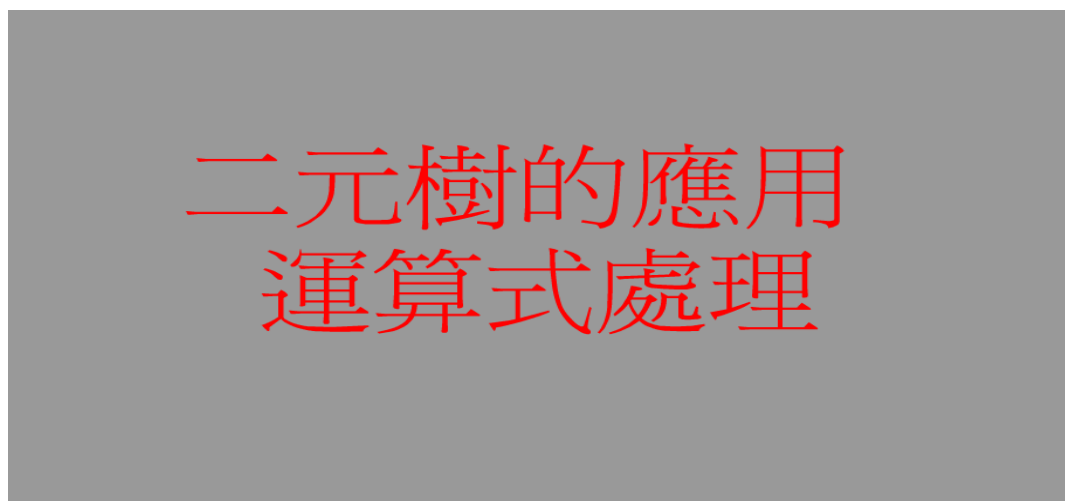


圖 22-1 二元樹的應用運算式處理教學動畫 初始畫面



圖 22-2 中序運算式的運算元 圖形表示

將運算元轉為二元樹的葉節點，運算子是非終端節點，因考量運算子的優先順序，乘號大於加號，所以前後 2 個乘號運算子先處理，即可建立二棵二元樹，如下圖所示：

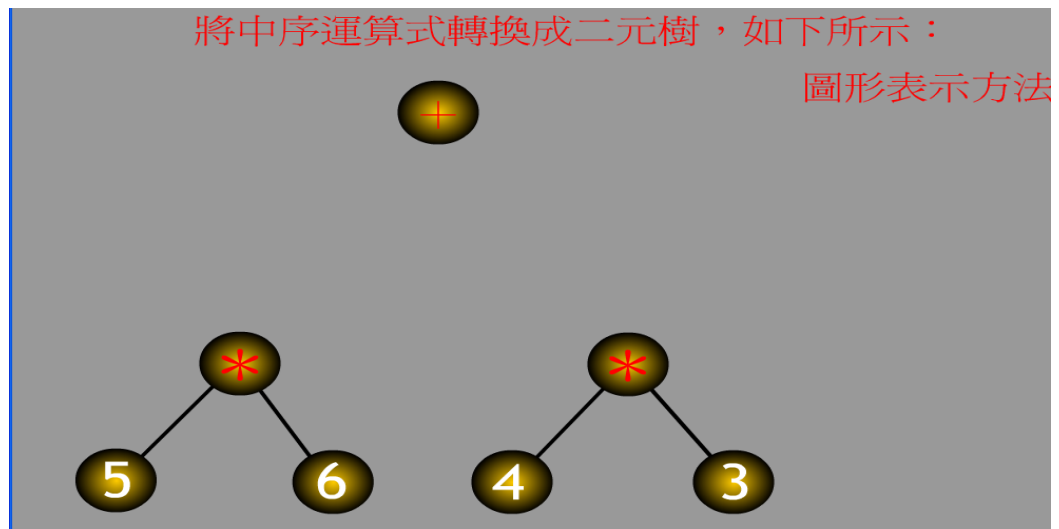


圖 22-3 二元樹的應用運算式處理教學動畫 圖形運算

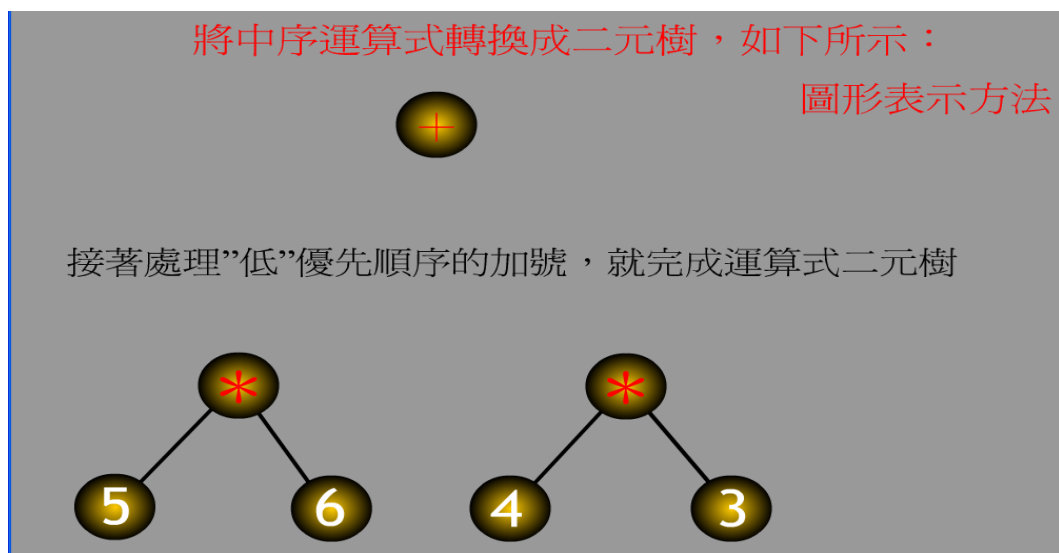


圖 22-4 二元樹的應用運算式處理教學動畫 文字呈現

將中序運算式轉換成二元樹，如下所示：

圖形表示方法

運算式結果

$$\begin{aligned} 5 * 6 &= 30 \\ 3 * 4 &= 12 \\ 30 + 12 &= 42 \end{aligned}$$

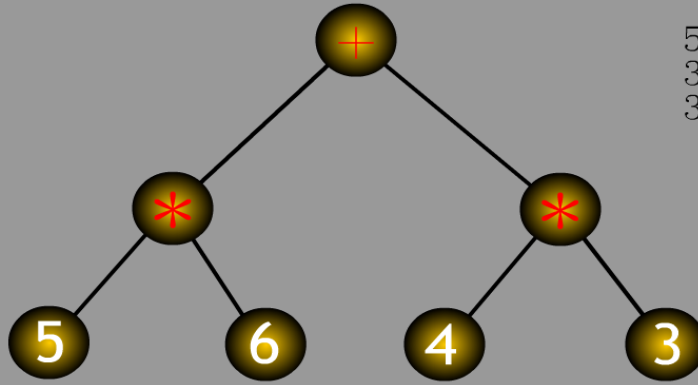


圖 22-5 二元樹的應用運算式處理教學動畫 文字呈現

佇列部分介紹鏈結串列建立佇列在這種資料結構中如何去做到建立、刪除、插入.等等的動作。在環狀鏈結佇列方面，有介紹了存入、取出及判斷佇是否已滿或是為空佇列。希望藉由這些選擇動作能讓使用者了解其中的不同處。在明白這些動作的原理之後，能夠一步一步的去瞭解程式流程都是為了滿足這些鏈結串列的各種動作，加強使用者對此課程的理解。

動態鏈結佇列

圖 23-1 動態鏈結佇列首頁圖

Front 和 Rear 成員變數分別為佇列前端和尾端的指標，Front 指標是指向單向串列的第 1 個節點，其初始值為 null，表示目前的佇列是空的。

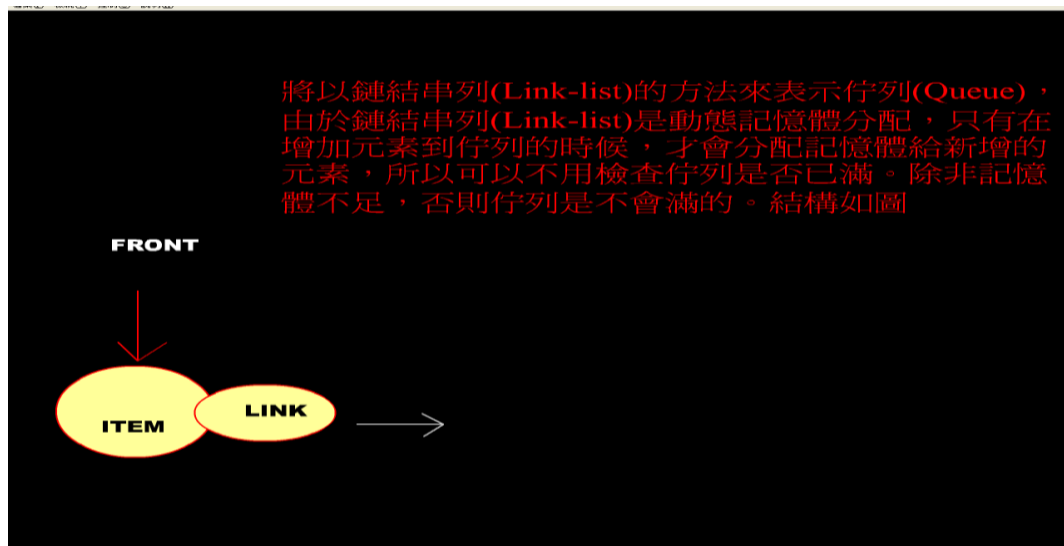


圖 23-2 鏈結串列建立佇列架構圖

將以鏈結串列(Link-list)的方法來表示佇列(Queue)，由於鏈結串列(Link-list)是動態記憶體分配，只有在增加元素到佇列的時候，才會分配記憶體給新增的元素，所以可以不用檢查佇列是否已滿。除非記憶體不足，否則佇列是不會滿的。結構如圖

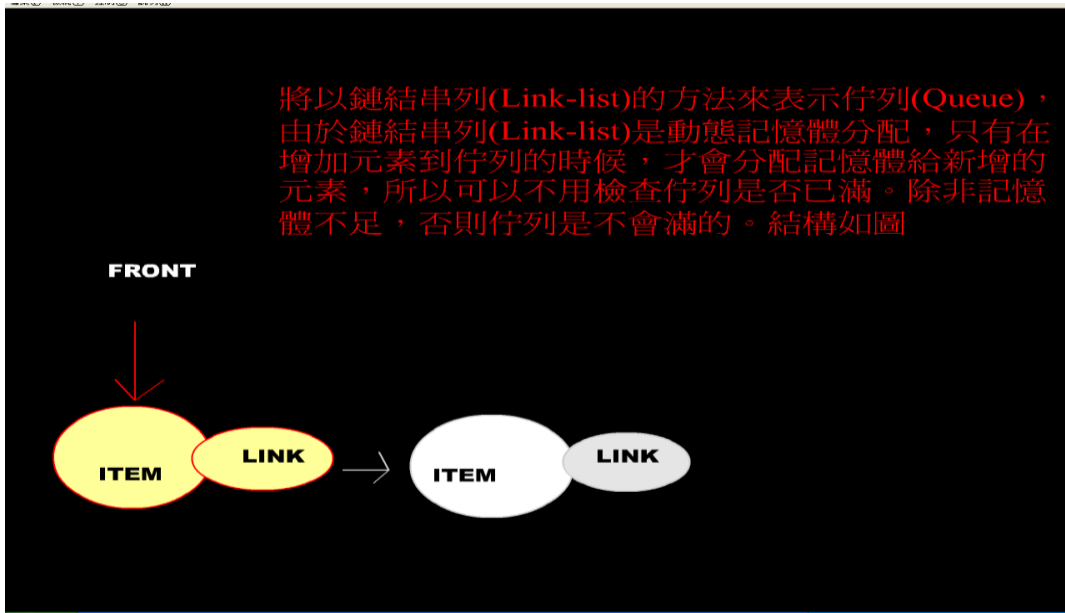


圖 23-3 鏈結串列建立佇列架構圖

將以鏈結串列(Link-list)的方法來表示佇列(Queue)，由於鏈結串列(Link-list)是動態記憶體分配，只有在增加元素到佇列的時候，才會分配記憶體給新增的元素，所以可以不用檢查佇列是否已滿。除非記憶體不足，否則佇列是不會滿的。結構如圖

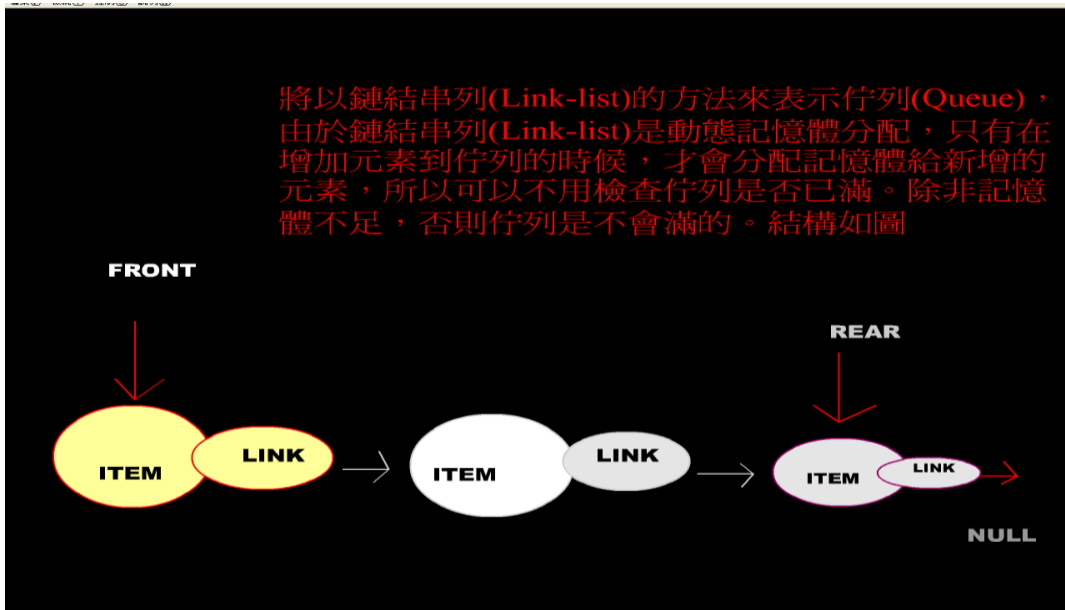


圖 23-4 鏈結串列建立佇列架構圖

環狀佇狀也是使用一維陣列實作的有限元素數佇列，其差異只在使用特殊技巧來處理陣列索引值，將陣列視為一個環狀結構。如下圖所示：

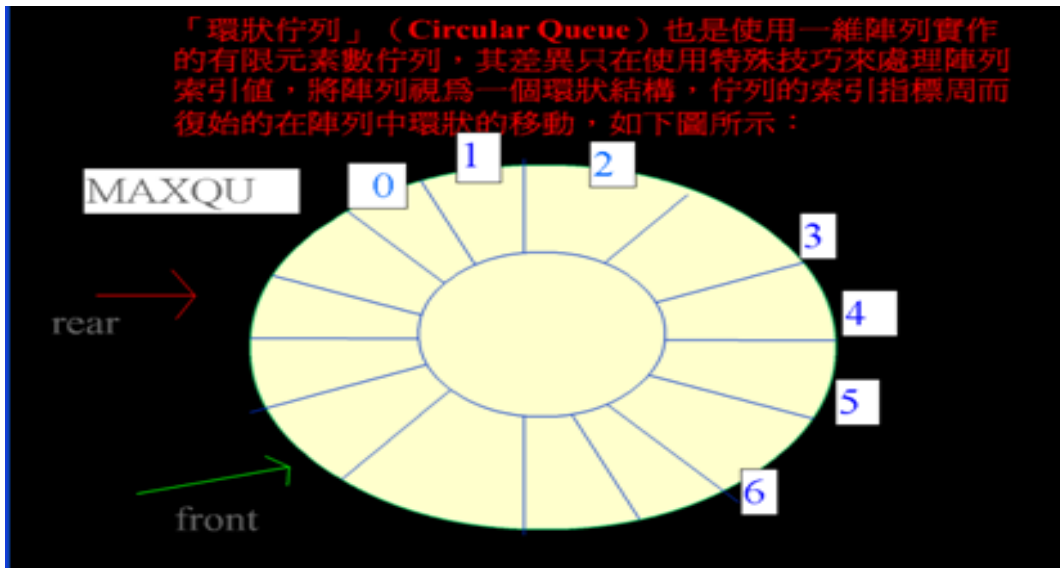


圖 24-1 環狀佇列基本架構圖

一個陣列大小 `queue.length` 為 4 的環狀佇列，當 `rear=3` 時到達陣列邊界，此時再新增佇列元素 5，`rear++` 等於 4，超過陣列尺寸，所以需要將它歸 0，如下所示：

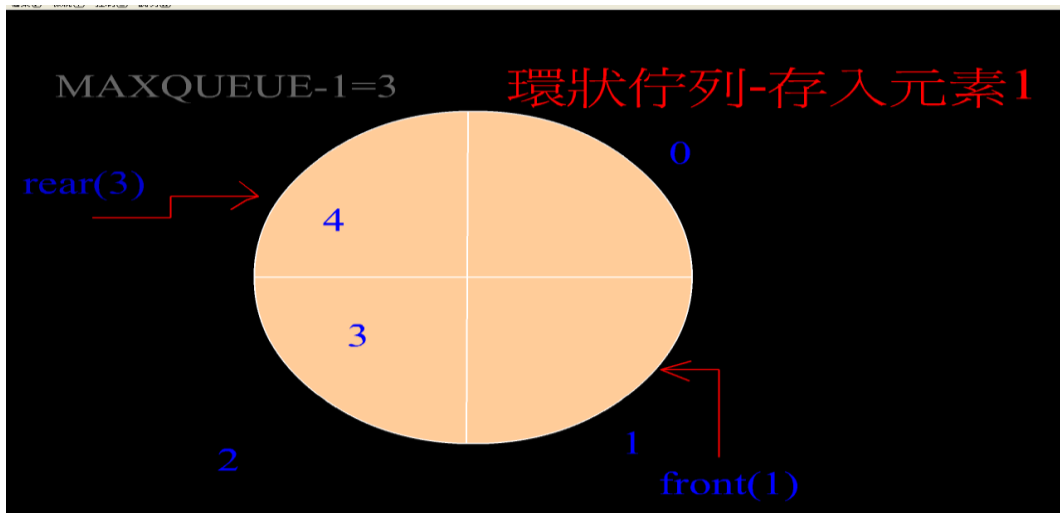


圖 24-2 環狀佇列-存入元素圖

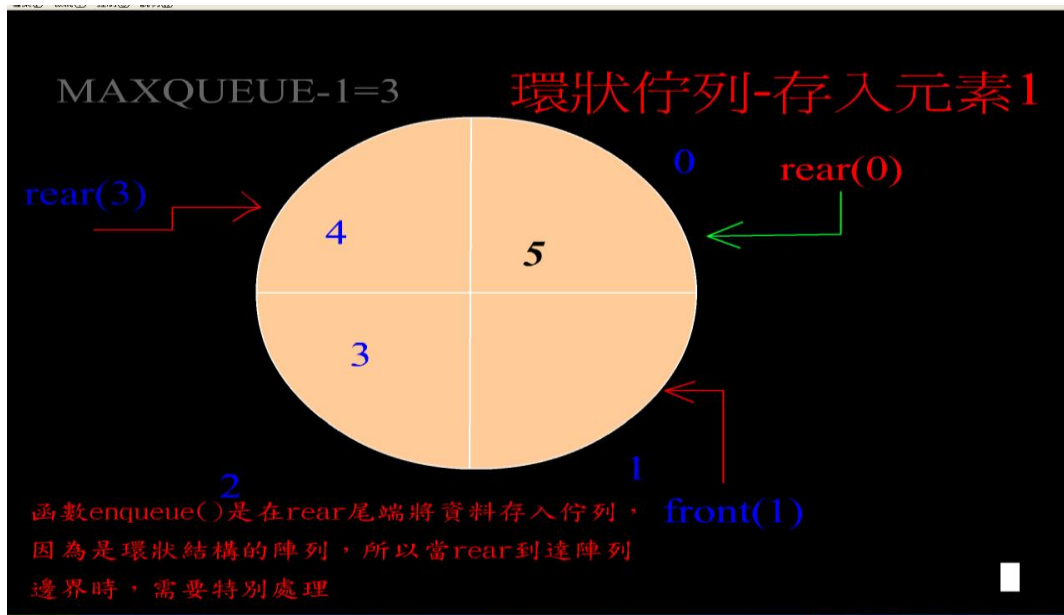


圖 24-3 環狀佇列-存入元素圖

MAXQUEUE為4的環狀佇列，當rear = 3時到達陣列邊界，
此時再新增佇列元素5，rear++等於4，超過陣列尺寸，所以需要將它
歸0，如下所示：

```
rear++;
if ( rear == MAXQUEUE ) rear = 0;
```

?:條件運算子，如下所示：

```
rear = ( rear+1 == MAXQUEUE ) ? 0 : rear+1;
```

使用餘數運算，如下所示：

```
rear = (rear+1) % MAXQUEUE;
```

圖 24-4 環狀佇列-存入元素圖

一個陣列大小 queue.length 為 4 的環狀佇列，當 rear=3 時到達陣列
邊界，此時再取出佇列元素 3，front++等於 4，超過陣列尺寸，所以
需要將它歸 0，如下所示：

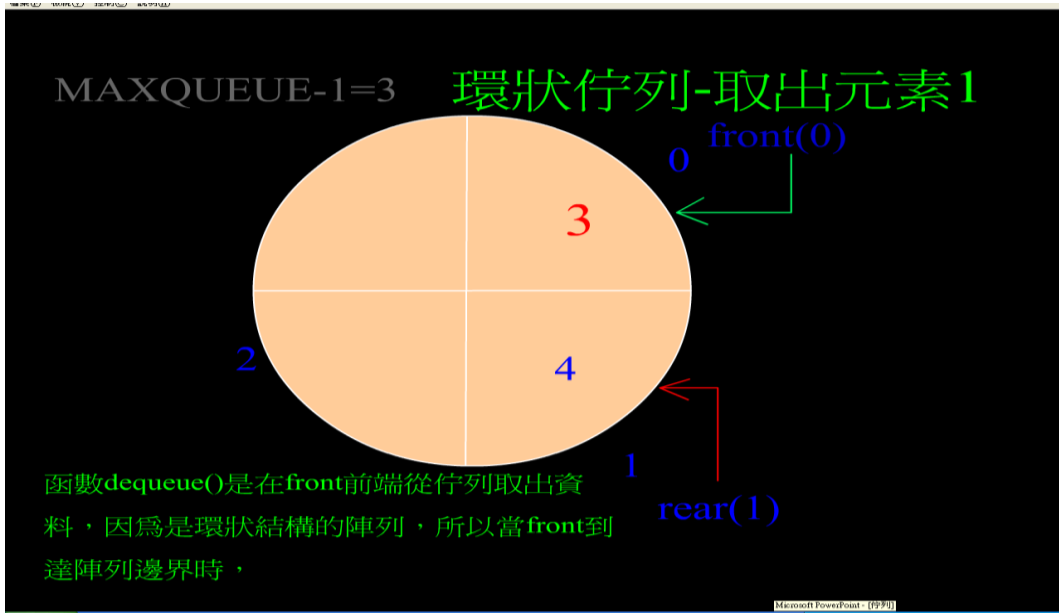


圖 24-5 環狀佇列-取出元素圖

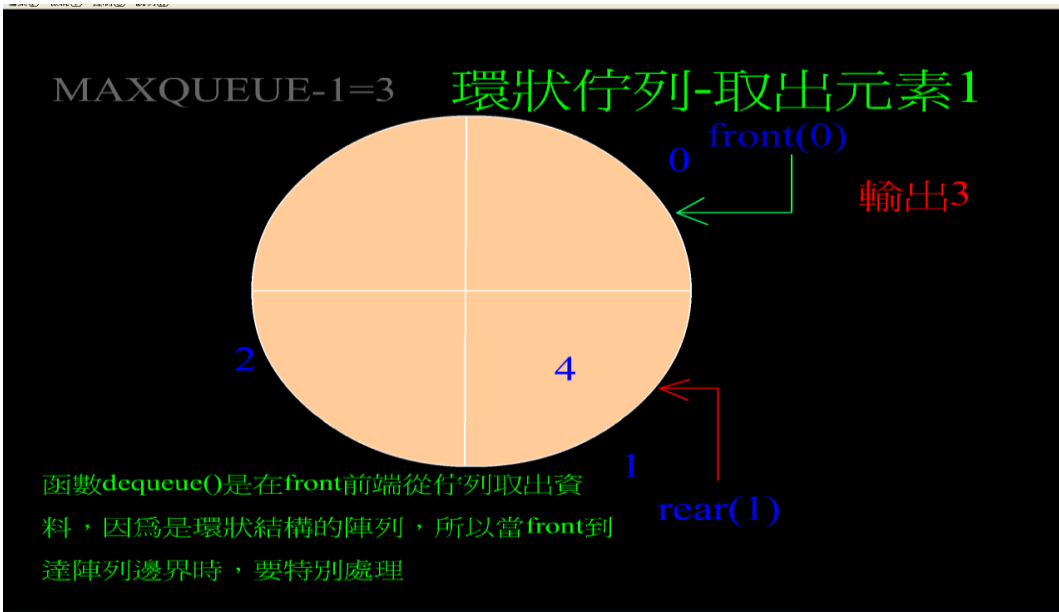


圖 24-6 環狀佇列-取出元素圖

```
MAXQUEUE為4的環狀佇列，當front = 3時到達陣列邊  
界，此時再從佇列取出元素3，front++等於4，超過陣列尺  
寸，所以需要將它歸0，如下所示：  
front++;  
if ( front == MAXQUEUE ) front = 0;  
?:條件運算子，如下所示：  
front = ( front+1 == MAXQUEUE ) ? 0 : front+1;  
使用餘數運算，如下所示：  
front = (front+1) % MAXQUEUE;
```

圖 24-7 環狀佇列-取出元素圖

isQueueFull()方法:環狀佇列是否已滿，現在環狀佇列尚有 1 個空間
沒有存入元素，如下所示:

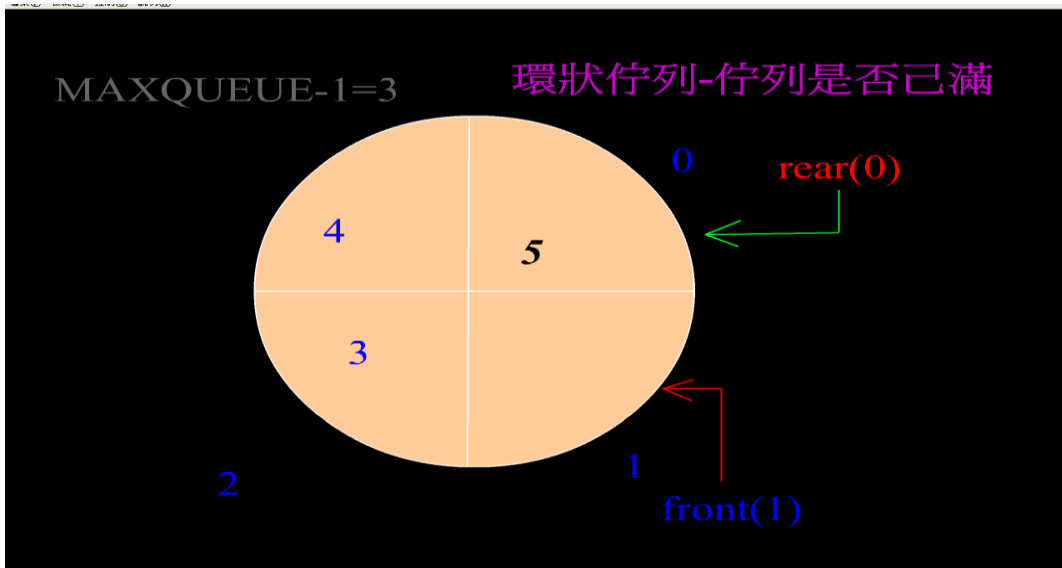


圖 24-8 環狀佇列-確認佇列是否已滿圖

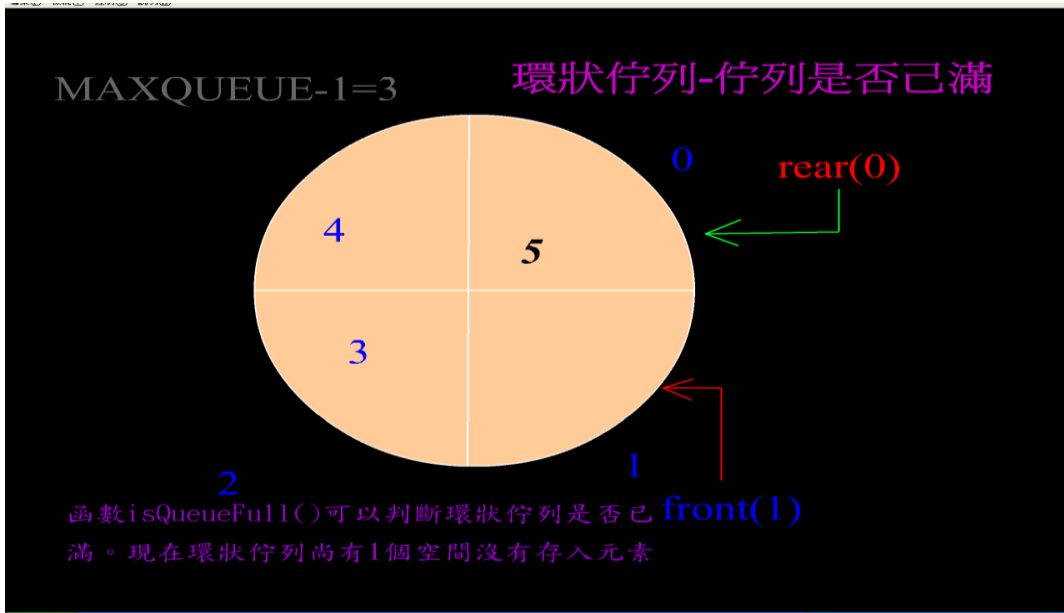


圖 24-9 環狀佇列-確認佇列是否已滿圖

再執行一次dequeue()取出最後1個元素4，可以發現front和rear指標相等，換句話說，只需判斷兩個指標是否相等，就可以判斷環狀佇列是否已經空了，如下所示：

```

if ( front == rear ) return 1;
else      return 0;

```

圖 24-10 環狀佇列-確認佇列是否已滿圖

isQueueEmpty()方法:環狀佇列是否已空，現在環狀佇列尚餘1個元

素，如下所示：

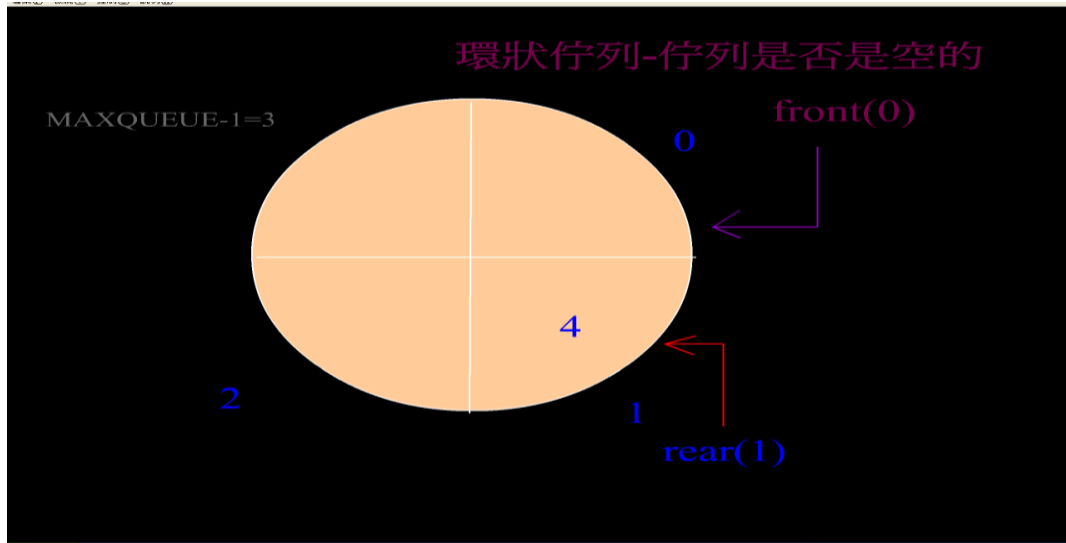


圖 24-11 環狀佇列-確認是為空佇列圖

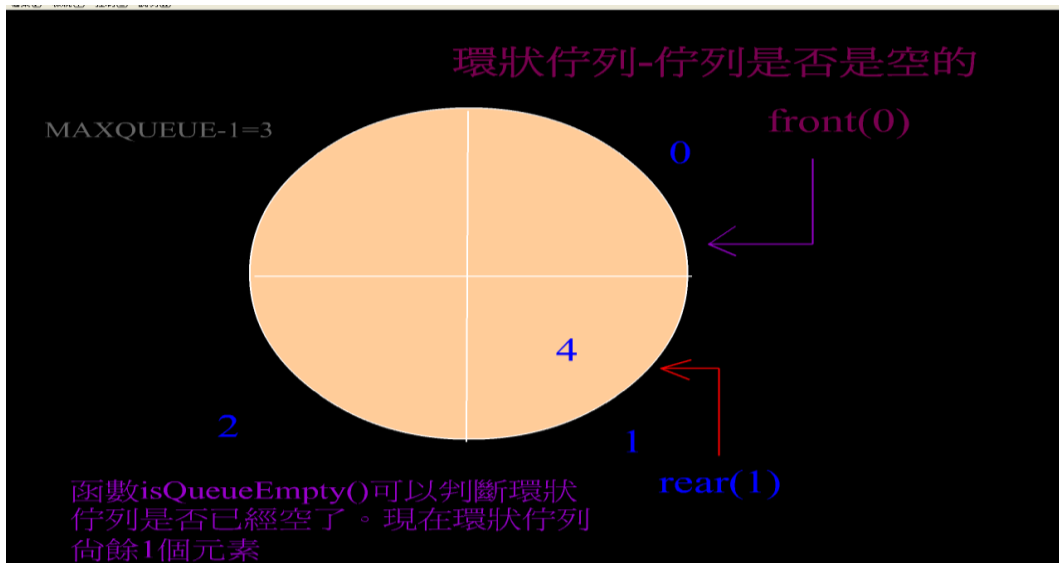


圖 24-12 環狀佇列-確認是為空佇列圖

再執行enqueue(6)新增元素6，可以發現front和rear指標相等，沒有辦法判斷環狀佇列是已空和全滿，因為兩個指標都是指向相同索引值1。所以，環狀佇列全滿就是指標rear和front相隔一個空間，換句話說，為了分辨環狀佇列是已空和全滿，其實際的儲存空間是陣列尺寸減1，如下所示：

```
int pos;  
pos = (rear+1) % MAXQUEUE;  
if ( front == pos ) return 1;  
else return 0;
```

圖 24-13 環狀佇列-確認是為空佇列圖

第二節 專題實作遇到之問題與解決

因為我們電腦輔助教學主要教導法基本架構是建立在透過電腦的應用，與操作等功能。所以，我們要更要盡量避免使用者會發生的許多無法預期的個人反應，因此我們必須仔細考慮使用者對於基礎知識的深淺，為了不讓使用者對此課程感到厭煩的情緒，或反抗拒絕的學習心態，我們必須在教材的編寫上要更注意章節的順序：由淺入深，引領使用者進入個別單元學習。另外即我們認定的重要性重點並不一定符合使用者的要求或需要，因此在上述幾點，是我們在編寫教材中格外要小心且注意的地方。從一開始定好題目之後，我們一直尋找如何做好這個專題的方法，真的是很茫然，沒有方向感，我們思考著到底什麼在這個專題裡才是最重要的，經過了討論之後，我們決定先

從如何去呈現出資料結構的整個架構下手，而在此時林正平老師建議我們運用時下最熱門，也是最好用的動畫軟體之一「Flash」來製做我們課程內容的部分，剛開始挑了一本比較內容較簡單的書，但是單純只是閱讀並沒有得到好的學習成效。而在小組討論中，建邦提出他在「Flash」學習上更有效率的方法，就是挑自己會用到的地方學習教材，並且馬上應用於實作的資料結構動畫教學上能更快熟悉「Flash」的軟體應用。這也同時讓我們在專題的製作上得到更好的學習與應用。

第五章 結論

第一節 線上輔助教學未來的發展性

隨著知識經濟時代革新的瞬息萬變，企業主意識到永續生存的關鍵在於人才培育，紛紛思考如何以更有效率的方式，提升員工的工作能力；所以網際網路的興起，逐漸在人們生活型態發生變化，培育人才的方法也多了一種選擇，一種「運用網路傳遞、擷取學習資訊及內容的學習方式」，透過學習入口網站、線上圖書館、學習社群、虛擬教室等方法，在不受時間、地點限制的情況下有效率地學習各種知識與技能。而線上教學更不只侷限於學術研習上，在企業裡更是不可替代的因為無論從事員工訓練及知識分享都是極佳的選擇，所以在未來的數位科技中依然有其重要性。

第二節 專題的心得與感想

一個好的電腦輔助教學系統，必須要有良好的規劃及設計，所有的教學內容都必須經過完善的設計，而且針對不同的年齡層而有不同的設計才行，因此我們在設計我們的教材時，是經過多次的修改才定案的，主要是為了讓使用者的學習更加有效率。在製作專題的過程

中，難免會有意見分歧的時候，但是我們總是會經過良好的互動，藉由不斷的溝通和討論，來產生我們都認同的結果；藉由這些溝通，使得我們更加了解對方，也讓我們相處得更為融洽，也更能去瞭解彼此的差異，無論在研究專題的動畫製作或是文字上的呈現都能分工合作努力完成它，所以這次的經驗可真是獲益良多，不僅在專題上有所獲，也更加增進彼此的友誼。而在製作專題的過程中，每週三都固定和老師的面會，他總是提供一些意見來引導我們讓此專題的輔助教學系統更加完整，每次的指導都讓我們受益良多，難能可貴是無論假期或是學期中老師都不會缺席，持之以恆不辭辛勞的為我們解惑。所以在此更要感謝林正平老師辛苦地教導我們，使得我們獲益良多，所以在此要特別地感謝他的教導，謝謝。

參考文獻

陳會安(2004) 資料結構理論與實務以 JAVA2 語言實作 學貫行銷股份有限公司

趙英傑(2006) 碼上就會-Flash 8 動畫設計寶典 上奇科技

恩光技術團(2006) Flash 8 動畫寶貝 基峰資訊

施威銘研究室(2006) Flash 8 躍動的網頁 旗標出版

高職進修網站 Flash MX 教學

<http://elearning.stut.edu.tw/flash/index.htm>

Flash 8 動態網頁教學

<http://web.chsh.chc.edu.tw/computer/flash/>

國立中正網路大學 數位學習平台

http://server4.webedu.ccu.edu.tw/resource_plan.htm

附錄、成果光碟