

# 《資料結構》

一、請參考圖1：

- (一)由a點出發，做depth-first traversal（深度優先拜訪），請問那些節點（node）不會被訪問到？（3分）
- (二)由a點出發，做breadth-first traversal（寬度優先拜訪），請問那些節點（node）不會被訪問到？（2分）
- (三)假設圖1代表heap上各個節點（node）及其相互指向的關係。p、q、r三節點代表全域變數（global variables），其他的節點代表heap上的記憶體區塊。如果p→a的指標被消除，那些節點會變成無用的垃圾節點？你必須詳細描述尋找垃圾節點的方法及所需之資料結構。你的演算法只能從a節點出發，它必須指出所有的垃圾節點，並且你的演算法只能在每一個節點儲存很少量的資料。請問你的演算法必須在每一個節點儲存那些資料？（15分）

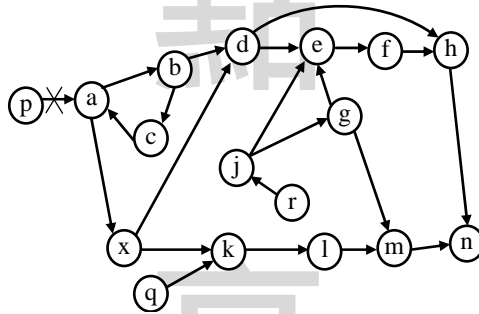


圖1 一個有向圖 (a directed graph)

<b>試題評析</b>	本題使用圖形的追蹤，DFS 與BFS 即可解出。
<b>考點命中</b>	《高上資料結構》，王致強編撰，頁8-22~28。

**答：**

- (一)g,j,q,r
- (二)g,j,q,r
- (三)如果p→a的指標被消除,a、b、c、d、x會變成垃圾節點。

演算法如下：

先將全部的節點都作記號；

For 每一個存在的節點 do 進行追蹤，DFS或BFS皆可，將追蹤到的節點記號消除；

最後，記號未消除的節點皆為垃圾節點。

二、定義如下的函數F： **【版權所有，重製必究！】**

如果x是偶數，則 $F(x) = x/2$ ；

否則 $F(x) = F(F(3x+1))$

(一)請問 $F(11) = ?$  (5分)

(二)請證明對於任何正整數w，我們都可以在有限時間內計算 $F(w)$ 。(提示：每個奇數可以寫成 $(2i+1)2^k-1$ 的形式，再採用數學歸納法來證明。)(15分)

<b>試題評析</b>	測驗考生的遞迴程式的推演能力，屬於進階問題。
<b>考點命中</b>	《高上資料結構》，王致強編撰，頁5-58精選例題38。

**答：**

(1)  $F(11)=F(F(34))=F(17)=F(F(52))=F(26)=13$

(2) 取  $w = (2i + 1)2^k - 1$ ，使用 Induction 證明即可。

Basis：當  $k=0$  時， $F(x)=F(2i)=i$ 。

Hypothesis：當  $k < n$  時，對任意的  $i$  值， $F(x)$  之值皆可以計算出來。

Induction：證明  $k=n$  時，亦成立。

$$\begin{aligned} & F((2i+1)2^n-1) \\ &= F(F(3(2i+1)2^n-3+1)) \\ &= F(F((3(2i+1)2^{n-1}-1)2)) \\ &= F((6i+3)2^{n-1}-1) \\ &= F((2(3i+1)+1)2^{n-1}-1) \\ &= F((2m+1)2^{n-1}-1) \dots \text{ 是可以計算出來的。} \end{aligned}$$

三、Knuth, Morris及Pratt發明了一個快速的字串比對方法 (string pattern matching)。他們的方法採用一個失敗函數 (failure function)。失敗函數其實就是一個輔助的資料結構，用來加速比對。請依他們的方法計算下列字串的失敗函數。你必須說明失敗函數的定義為何，以及失敗函數如何加速比對。(15分)

index	0	1	2	3	4	5	6	7	8	9
pattern	a	b	b	a	b	c	a	b	b	a
failure	?	?	?	?	?	?	?	?	?	?

**試題評析** 測驗字串比對之KMP演算法。

**考點命中** 《高上資料結構》，王致強編撰，頁3-72 ~74。

**答：**

失敗函數定義如下：

$f(j) =$  最大的  $i$  值(其中，存在有  $j > i \geq 0$ ，而且  $p_0p_1 \dots p_i = p_jp_{j+1} \dots p_i$ )；

$f(j) = -1$ ，其他情況。

失敗函數如下：

index	0	1	2	3	4	5	6	7	8	9
pattern	a	b	b	a	b	c	a	b	b	a
failure	-1	-1	-1	0	1	-1	0	1	2	3

當比對到  $s_i \neq p_j$  時，取  $j \leftarrow f(j-1)+1$ ，然後繼續比對  $s_i$  與  $p_j$  下去，可避免須從頭比較的時間耗費。

四、請參考圖2。每一條線段上的數字代表兩節點間的距離。請找出a節點到k節點的最短路徑的長度。並請說明你的方法如何應用在非常大型的圖裡。(15分)

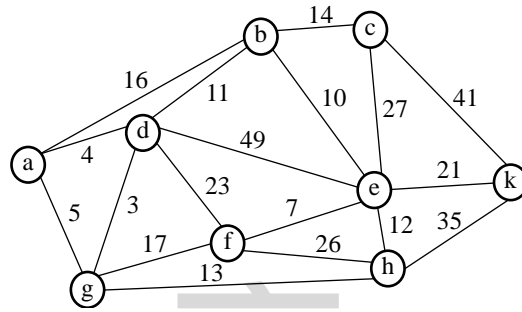


圖2 最短路徑

<b>試題評析</b>	測驗Dijkstra 的單起點最短路徑演算法。
<b>考點命中</b>	《高上資料結構》，王致強編撰，頁8-60~62。

**答：**

使用Dijkstra 最短路徑演算法，可求得 a 到 k 的最短路徑：a→d→b→e→k，總長度為46。

在大型的圖裡，可以用鄰接串列來表示圖形，再使用 Fibonacci-Heap 的delete-smallest-key 找出距離最小的頂點，利用其Decrease-key 的運算來修改各頂點的距離，可以使得整個 Dijkstra algorithm 時間複雜度為  $O(n\log n + e)$ 。演算法如下：

- (1) procedure shortestpath( $v_0$ , cost, dist, pred);
- (2) begin
- (3)  $S \leftarrow \{v_0\}$ ;  $dist[v_0] \leftarrow 0$ ;  $prev[v_0] \leftarrow 0$ ; //設定起點初值
- (4) for each  $u \in V - \{v_0\}$  do // 設定其他頂點的相關初值
- (5) begin
- (6)  $dist[u] \leftarrow cost[v_0, u]$ ; // 路徑長度初值
- (7) if  $cost[v_0, u] \neq \infty$  then  $pred[u] \leftarrow v_0$ ; // 記錄前行節點
- (8) else  $pred[u] \leftarrow 0$ ;
- (9) end;
- (10) while  $|S| < |V|$  do
- (11) begin
- (12)  $u \leftarrow$  選出  $S$  之外  $dist$  最小的頂點;
- (13) // 用 F-heaps 的 delete-min 運算，每次 amortized
- (14) // cost 為  $O(\log |V|)$ ，總時間為  $O(|V| \log |V|)$ 。
- (15)  $S \leftarrow S \cup \{u\}$ ;
- (16) for each  $(u, w) \in E(G)$  do
- (17) if  $w \notin S$  then
- (18) if  $dist[u] + cost[u, w] < dist[w]$  then
- (19) begin
- (20)  $pred[w] \leftarrow u$ ;
- (21)  $dist[w] \leftarrow dist[u] + cost[u, w]$ ;
- (22) // 使用 F-heaps 的 decrease-key()
- (23) // amortized cost  $O(1)$
- (24) end;
- (25) end;
- (26) end;

【版權所有，重製必究！】

五、請參考圖3。圖3是一個activity-on-edge網路。在activity-on-edge網路中，一項計畫可以分成很多件工作，每一件工作由一條線段代表，線段上的數字代表該工作所需的時間（以工作日為單位），線段的箭頭代表工作的先後關係。例如在圖3中，ab及db線段代表的工作完成之後，bc、be、及bf線段代表的工作才可以開始進行，其他的先後關係依此類推。a節點是起點，k節點是全部工作的完成點。請找出k節點的最早完成時間及關鍵路線（critical path）。並請說明你的方法如何應用在非常大型的圖裡。（15分）

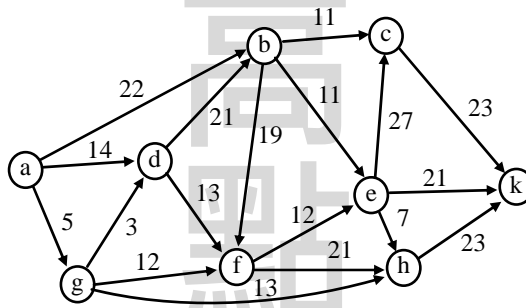


圖3 Activity-on-edge網路

<b>試題評析</b>	測驗 AOE 邊工作網路之計算。
<b>考點命中</b>	《高上資料結構》，王致強編撰，頁8-89~94。

**答：**

- (一)k節點最早完成的時間為116，關鍵路線為 ad、db、bf、fe、ec、ck。
- (二)在大型的圖裡，圖形以Adjacency List 表示，以stack 記錄可以輸出的節點，以forward 求每個頂點的 ee(Earliest Event Occurrence Time)。再用Reverse Adjacency List，backward 方向求每個頂點的 le(Latest Event Occurrence Time)，其中也是以stack 記錄可以輸出的節點。最後再計算，每個線段的 e 與 l，如果 e=l的線段即為關鍵活動。

(1)先以forward 方式由 a 往 k 方向求 ee。

v	a	b	c	d	e	f	g	h	k
ee(v)	0	35	93	14	66	54	5	75	116

(2)再以backward 方式由 k 往 a 方向求 le。

v	a	b	c	d	e	f	g	h	k
le(v)	0	35	93	14	66	54	11	93	116

(3)最後計算每個線段的 e 與 l。

線段	e	l	關鍵活動
ab	0	13	
ad	0	0	v
ag	0	6	
bc	35	82	
be	35	55	
bf	35	35	v
ck	93	93	v
db	14	14	v
df	14	41	
ec	66	66	v

eh	66	86	
ek	66	95	
fe	54	54	v
fh	54	72	
gd	5	11	
gf	5	42	
gh	5	80	
hk	75	93	

六、在一個二元樹裡有許多節點 (nodes)。假設每一個節點的資料結構如下圖：

LEFT	DATA	RIGHT
------	------	-------

其中DATA欄位為該節點的資料。LEFT欄位為指向左方子樹的指標變數。RIGHT欄位為指向右方子樹的指標變數。

如果節點p沒有左方子樹，其LEFT欄位為空指標 (null pointer)。同理，如果節點p沒有右方子樹，其RIGHT欄位為空指標 (null pointer)。

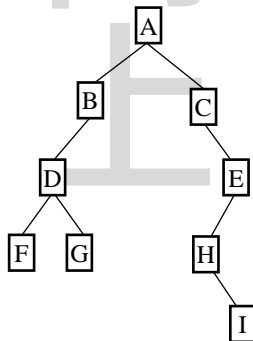
(一)如果一個二元樹有n個節點，那麼它有幾個空指標？(5分)

(二)我們可以利用原本是空指標的欄位來儲存引線 (threads)。二元樹加上引線的結果稱為引線樹 (threaded trees)。當然我們必須在各節點再加上兩個欄位LTAG及RTAG，共5個欄位，如下圖所示：

LTAG	LEFT	DATA	RIGHT	RTAG
------	------	------	-------	------

如果LEFT欄位代表一般的節點指標，則LTAG=0。如果LEFT欄位代表引線指標，則LTAG=1。同理，如果RIGHT欄位代表一般的節點指標，則RTAG=0。如果RIGHT欄位代表引線指標，則RTAG=1。

請將下圖的二元樹加上適當的引線指標，讓它變成引線樹，並請繪圖標出A到I共9個節點中所有引線指標指向的節點。(10分)

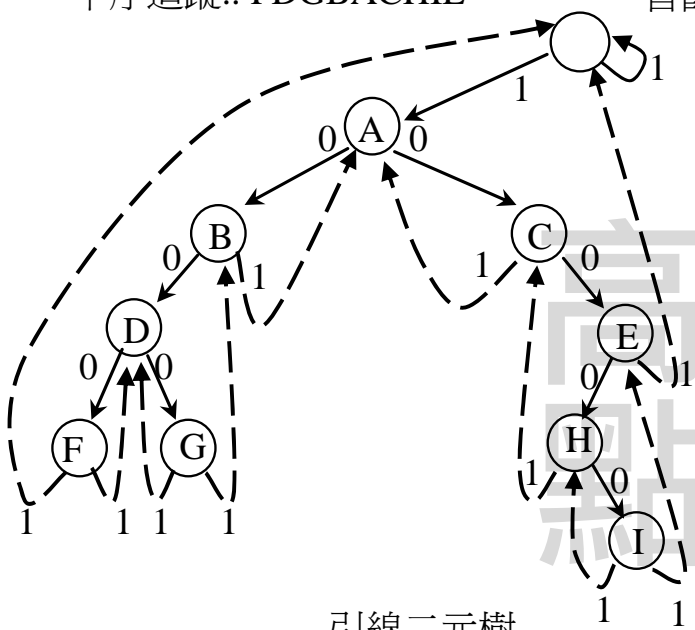


試題評析	測驗引線二元樹的建置。
考點命中	《高上資料結構》，王致強編撰，頁6-34~39。

答：

中序追蹤.: FDGBACHIE

首節點



引線二元樹

高上

【版權所有，重製必究！】