

《資料結構》

試題評析	<p>第一題：字串表示法與字串比對問題，為鏈結串列應用。</p> <p>第二題：大富翁遊戲的資料結構設計，屬於應用題。</p> <p>第三題：程式語言物件導向的類別定義，本題應屬於程式語言範圍。</p> <p>第四題：CPU 排程所需資料結構，屬於應用題。</p> <p>綜合而言，今年題目並未考到較複雜的資料結構，大多題目偏向應用題型，有些必須具備該應用領域相關知識，才能了解題意，選出適當的資料結構來配合。一般考生的分數可能在 50~60 左右，具備相關應用領域知識者，應可拿到 70~80 分。</p>
考點命中	<p>第一題：《資料結構》，王致強編撰，高點出版，頁 3-72~3-76。</p> <p>第二題：《資料結構》，王致強編撰，高點出版，第 3 章鏈結串列。</p> <p>第三題：《資料結構》，王致強編撰，高點出版，頁 3-2~3-3。</p> <p>第四題：《資料結構》，王致強編撰，高點出版，頁 4-17~7-18、頁 7-3~7-6。</p>

一、「字串比對」是要找出某一有興趣的字串是否包含於另外一個較大的字串或文章中，在字串比對演算法中，我們需要怎麼樣的資料結構來幫助我們求得該字串是否出現？請詳述之。(25分)

【擬答】

字串比對可以使用 **KMP 演算法(Knuth-Morris-Pratt)**效率較佳，配合使用的資料結構為連續記憶體(如一維陣列)較佳，以方便配合 **KML 演算法**。

	x ₁	x ₂		x _n	
s[i]	s[i+1]		s[i+n-1]		

(1) 先對 pattern 找出其 prefix function(或稱為 failure function) f 。

f(j)的求法：

$f(j) =$ 最大的 $i \leq j$, 使得 $p_1 p_2 \dots p_i = p_{j-i+1} p_{j-i+2} \dots p_j$, 存在有 $i \geq 1$;

$f(j) = 0$, 其他情况。

(2) KMP 演算法進行字串比對

 $q=0;$

```
for (i=1;i<=n;i++)
```

```
{ while (q>0 && p[q+1]!=s[i]) q=f[q];
```

```
if (p[q+1]==s[i]) q++;
```

```
if (q==m) { printf("pattern occurs at %d\n", i-m+1);
```

$$q=f[q];$$

}

}

時間複雜度為 $O(m+n)$ ，其中較長的字串長度為 n ；較短字串長度為 m 。

二、在大富翁的遊戲中，我們需要那些資料結構來幫助我們設計該遊戲？請詳述之。(25 分)

說明：大富翁（Monopoly）是一種多人策略圖版遊戲。參賽者分得遊戲金錢，憑運氣（擲骰子）及交易策略，買地、建樓以賺取租金。

【擬答】

- (一) Monopoly 有一個盤面，通常是一個方形環狀的道路，此一盤面使用環狀的鏈結串列來實作，讓代表玩家的棋子沿著鏈結串列循環前進即可。
- (二) 每個位置有路名、所屬玩家名稱、是否有建房屋等等資訊，記錄在鏈結串列的節點中，每個串列節點有資料欄位記錄前述資料。
- (三) 機會與命運兩疊卡片，分別各用一個鏈結串列來表示，每次翻卡片時，就由串列開頭取得第一個節點。
- (四) 骰子則以亂數函數實作即可。

三、Point p 是 XY 平面中的某一點，由 x 及 y 的座標所組成如下： (x, y) ，請用物件的方式寫出 Class Point 並利用 Class Point 進一步定義出 Class Line，Line 是由 XY 平面中的兩點所組成的線段。(25 分)

【擬答】

```
class Point {
    private:
        float x, y;
    public:
        Point(int x, int y)
        {   this.x=x;   this.y=y;   }
}

class Line {
    private:
        Point *point1, *point2;
    public:
        Line(point *p1, point *p2)
        {   point1=new Point(p1->x,p1->y);
            point2=new Point(p2->x,p2->y);
        }
}
```

四、在作業系統資源管理程式中，我們需要那些資料結構來幫助我們適當的分配 CPU 的執行時間？請詳述之。(25 分)

【擬答】

CPU 排程(scheduling)需視系統採用何種排程演算法，才能決定使用那一種資料結構，以下列舉各種排程法說明：

- (一) 若使用先到先服務(FCFS, First-Come-First-Serve)，可以使用佇列(queue)的 FIFO 資料結構，即可以配合。
- (二) 若使用最短工作優先(SJF, Shortest Job First)，可以使用優先權佇列(priority queue)資料結構，實作上是以最小堆積(Min-Heap)，提供新到達工作加入隊伍的插入(Insertion) 與選取最短工作的 Delete-Min 兩個運算，時間複雜度皆為 $O(\log n)$ 。
- (三) 若使用優先權排程(Priority Scheduling)，也是使用優先權佇列(priority queue)資料結構，實作與 SJF 類似。
- (四) 若使用循環式排程(Round-Robin Scheduling)，佇列(queue)的 FIFO 資料結構，即可以配合。