《資料結構》

試題評析	今年考題重點著重在基本觀念與應用。 第一題：雜湊表的觀念，對考生而言並不難，但能否完整詳述雜湊表的各項要點，變成是取分的關鍵。 第二題：本題爲應用題，需要使用到遊戲樹的觀念，如果設計過人工智慧遊戲，則本題不致於太難。 第三題：互斥集合是資料結構中，一種著名的結構，其應用於最低成本伸展樹，亦爲重要的應用範例，因此考生本題應可取得不錯的分數。 第四題：使用陣列來模擬指標或鏈結串列的效果，爲基本的做法，取分亦不難。 綜觀今年考題，大多偏向申論式的問題，未有演算法或程式性的問題，考驗考生的組織能力，是否能完整詳述整體的做法，成爲分數高低的關鍵。預估一般考生應可取得 60~70 分，準備完整且組織能力佳者，應可得到 80 分或更高的分數。
	1.王致強「資料結構」上課講義第六回 PP. 10~13。 2.王致強「資料結構」上課講義第三回 PP. 50~52。 3.王致強「資料結構」上課講義第三回 PP. 44~45、第四回 P. 21。 4.王致強「資料結構」上課講義第二回 P. 61。

一、(一)請解釋 Hash function。其主要功能及設計考量點為何？（15 分）
(二)我們可以用那一種資料結構來實現它？（10 分）

【擬答】

(一)

1.Hash function 是採用一個計算位址的函數，將關鍵值以參數代入，用來計算出資料所應儲存的位置或索引。

2.其功能爲可以在 O(1)(常數時間)存取資料，包括插入新資料，搜尋資料等。

3.設計要點：

(1)碰撞機率：選擇適當的 Hash function 以減少碰撞機率。

(2)表格空間利用率：配置較大空間可以降低碰撞機率，但浪費空間；較小的的表格空間，較不會浪費空間，但碰撞機率相對會較高。

(3)滿溢處理(overflow handling)：當插入新資料時，若 bucket 已滿，如何找到空間存放新資料，分爲開放位址(open addressing)與獨立串列(separate chaining)兩類。

(4)減少群聚(clustering)：滿溢處理要注意儘量不要有 primary clustering 與 secondary clustering 現象發生。

(二)Hash function 使用陣列來實作，一個陣列分以分成 m 個 buckets，每個 bucket 可以包含數個 slots，每個 slot 可以儲存一筆資料。以 hash function 計算出 bucket 編號，然後存取 bucket，插入資料時，若 bucket 已滿，則使用滿溢處理以儲存資料：

1.開放定址法：就在表格中，以特定規則找到別一個未滿的 bucket 儲存資料。

2.獨立串列：則以鏈結串列將新資料建立一個節點存放。

二、在圍棋程式中，最常用的三種資料結構為何？請說明其用途。（30 分）

【擬答】

(一)圖形：記錄已落子的棋子位置，通常用鄰接矩陣(2-D 陣列)記錄棋子位置即可，此一陣列用記錄盤面狀況，來檢查死活，計算佔領區域的大小，以判斷勝負。

(二)遊戲樹：在人工智慧部份用來找尋電腦的最佳落子位置，通常使用 mini-max 演算法或 Alpha-beta pruning 演算法，來找尋最佳落子位置。

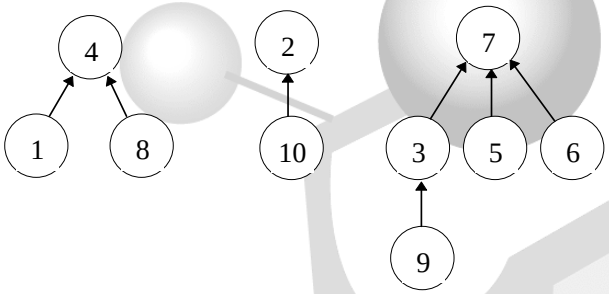
(三)堆疊或佇列：用 backtracking 搜尋遊戲樹時，需要用到堆疊；用廣度搜尋法時，使用佇列；若使用最佳優先搜尋法(best-first search)時，使用優先權佇列(priority queue)。

三、(一)請解釋 disjoint-set data structure。(10 分)
(二)請舉出一個應用的例子。我們可以用那一種資料結構來實現它？(15 分)

【擬答】

(一)disjoint-set 是一群集合，兩兩之間的交集皆爲空集合，則這群集合稱爲互斥集合。通常需要兩個運算。
1.聯集(union)：將兩個互斥集合合併爲一個集合。
2.找尋(find)：找尋元素所屬的集合。

(二)應用範例：在 Kruskal 演算法中，找尋最低成本伸展樹，可以用 disjoint- set 來協助檢查是有迴圈產生。disjoint-set 可以使用 trees 來實現，例. A={1,4,8} B={2,10} C={3,5,6,7,9}爲三個互斥集合，每一個互斥集合用一棵樹來代表。



資料結構
(1)parent[i]≧0：代表指向父親的指標。
(2)parent[i]＜0：表示 node i 爲 root，且整棵樹共有 –parent[i] 個 nodes。

運算：
(1)集合的聯集(Union)：合併兩棵樹,O(1)

```
1.union(x, y)
2.{
3.if (parent[x]<=parent[y])
4.{
5. parent[x]←parent[x]+parent[y];
6. parent[y]←x;
7. }
8. else
9. {
10. parent[y]←parent[y]+parent[x];
11. parent[x]←y;
12.}
13.}
```

(2)元素所屬於集合之找尋(Find)：O(logn)

```
1.find(x)
2.{
3. i←x;
4. while (parent[i]>=0)
5. i←parent[i];
6.// Collapsing
7. j←x;
8.while (parent[j]>=0)
9.{
10. k←parent[j];
11. parent[j]←i;
12. j←k;
```

```
13. }
14. return i;
15.}
```

四、我們如何在一個沒有支援 pointer 的程式語言中，利用那一種資料結構來實現 pointer？請舉例說明。(20 分)

【擬答】

以陣列來實作：使用兩個陣列，一個存放資料，另一個指出下一項資料的位置(註標)。

```
Datatype DATA[MaxItem];
int      LINK[MaxItem];
```