經濟部所屬事業機構105年新進職員甄試試題

類別:統計資訊 **節次:第三節**

科目:1.資料庫及資料探勘 2.程式設計

注意事項	1. 本試題共 3 頁(A3 紙 1 張)。 2. 可使用本甄試簡章規定之電子計算器。 3. 本試題分 6 大題，每題配分於題目後標明，共 100 分。須用藍、黑色鋼筆或原子筆在答案卷指定範圍內作答，不提供額外之答案卷，作答時須詳列解答過程，於本試題或其他紙張作答者不予計分。 4. 本試題採雙面印刷，請注意正、背面試題。 5. 考試結束前離場者，試題須隨答案卷繳回，俟本節考試結束後，始得至原試場或適當處所索取。 6. 考試時間：120 分鐘。

一、決策樹(Decision Tree)是一個常用於解決分類(Classification)問題的方法，【表 1】所列之收入、年齡、信用為資料的屬性，而每一個屬性都有二種可能的值，分別為 *l*(low)與 *h*(high)，而類別標籤則代表資料的類別，可以是 T (True) 或是 F (False)。請建立【表 1】的決策樹，並利用 $\mathbf{1}-\sum_{j=1}^{n}\boldsymbol{p}_j^2$(gini index, CART 演算法所使用)作為屬性選擇的根據，在公式中 *n* 代表資料類別的個數，$\boldsymbol{p}_j$代表類別 *j* 在資料集中出現的頻率，請回答下列問題(需寫出運算過程)。

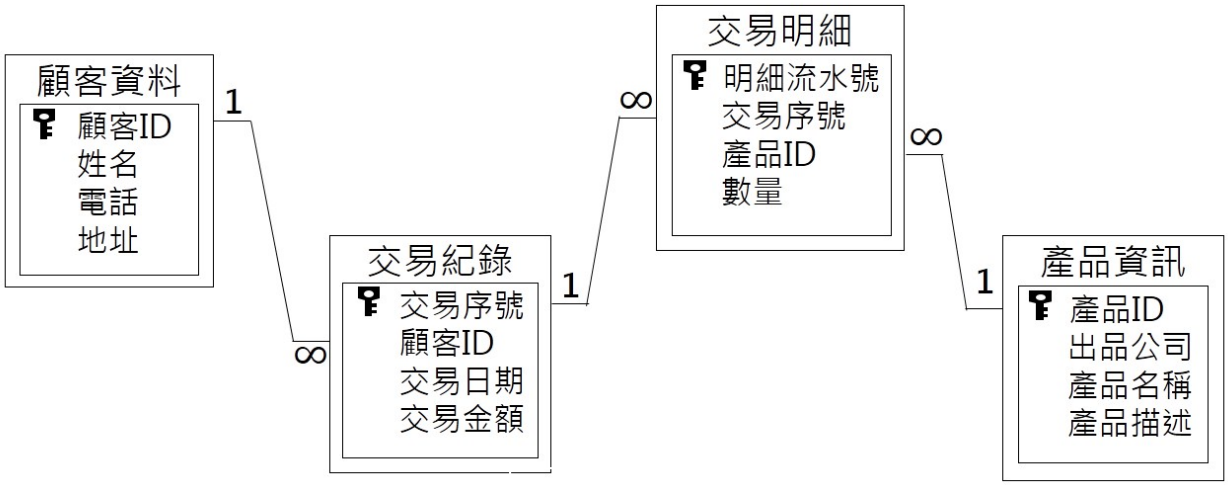
(一) 第一個被選出成為決策樹的根節點的屬性為何？（5 分）

(二) 請畫出此決策樹，並預測新進資料(收入= *l*、年齡= *l*、信用= *h*)的資料類別。（10 分）

【表 1】

收入	年齡	信用	類別標籤
l	*h*	*h*	T
l	*l*	*l*	F
l	*h*	*l*	F
h	*h*	*h*	T
h	*l*	*l*	T
h	*h*	*l*	T

二、【圖 1】為某公司的顧客交易資料庫架構，請使用 SQL 語法回答下列查詢。



【圖 1】

(一) 請列出交易明細表中，交易序號為 T0001 的所有資料。（5 分）

(二) 請列出所有顧客 ID 與其累積消費總金額，並由多至少排列。（5 分）

(三) 請列出產品 ID 為 P0001 與 P0002 的產品，曾經一起被購買的次數。（5 分）

三、頻繁樣式探勘(Frequent Pattern Mining)的議題，最早於 1994 年由 Agrawa 與 Srikant 提出，目的在於透過分析顧客交易紀錄，了解產品被購買的規律性，並對此問題提出 Apriori Algorithm，利用頻繁樣式(frequent itemset)的向下封閉性質(downward closure property)，有效解決產品間各種排列組合關係造成的高複雜度計算。

(一) 請簡述頻繁樣式的向下封閉性質。（5 分）

(二) 請利用 Apriori Algorithm 分析【表 2】顧客交易紀錄，詳列頻繁樣式探勘之過程與結果（假定最小支持(minimal support)為 2 次）。（15 分）

【表 2】

TID	items
T100	I1, I2, I3, I5
T200	I2, I6
T300	I2, I3, I5
T400	I2, I4, I6
T500	I4, I5, I6
T600	I1, I2, I5
T700	I1, I2, I3
T800	I2, I6

四、請依算術運算式(((9+1)*3)/((7-6)+5))-((2*(3-2))+1)，回答下列問題：

(一) 請繪出此算術運算式之二元樹，其終端節點均為運算元（1、2、3、5、..），非終端節點均為運算子（+、-、*、/）。（5 分）

(二) 為求得運算式之值，可採「中序(infix)」、「前序(prefix)」或「後序(postfix)」等表示法，請從記憶體耗用、程式複雜度觀點，比較此 3 種表示法何者較佳？為什麼？（6 分）

(三) 請將此運算式，改為後序表示法(postfix expression)。（5 分）

(四) 欲使用堆疊(stack)來求得此算術運算式之解，請畫出該堆疊的資料歷程變化。（6 分）

五、「遞迴」與「迴圈」是程式設計重要的手法，請回答下列問題：

(一) 兩設計手法相比，「遞迴」的優點、缺點為何？（8 分）

(二) 下列左右兩邊之程式碼，左邊以「遞迴」手法撰寫，右邊擬將之改為以「迴圈」手法撰寫，請於右邊程式空白處填入正確程式碼。（9 分）

遞迴

```
int fib(int n){
        if(n == 0) return(0);
        else if(n == 1) return(1);
        else return(fib(n-1)+fib(n-2));
}
```

迴圈

```
int fib(int n){
        int i;
        int fib_n;
        int fib_n_1;
        int fib_n_2;

        if(n == 0) return(0);
        else if(n ==1) return(1);
        else{
                fib_n_2=0;
                fib_n_1=1;
                for(i=2; i<=n; i++){
```

（空白處）

```
                }
                return(fib_n);
        }
}
```

六、處理巨量資料時，分析人員常需面對龐大資料，且資料量遠大於記憶體容量。今有一循序檔 data.txt，內含 9 筆資料如下，欲對該檔進行排序。惟受限於記憶體容量，讀入 data.txt 資料時，每次最多只能 6 筆。在考量磁碟處理速度遠低於記憶體情況下，請以敘述表示法，設計一可兼顧減少磁碟存取次數及提高排序效率之排序演算法。（11 分）

data.txt

6000	800000	500	2	40	9000000	3	1.5	70000