

113 年公務人員特種考試國家安全局國家安全情報人員 考試試題

考試別：國家安全情報人員考試

等 別：三等考試

類科組別：資訊組 (選試英文)

科 目：計算機概論

一、請將以下 IEEE-754 單倍精確度 (32 bit) 浮點數格式 (二進位)，轉「十進位浮點小數」。

Sign (1 bit)	Exponent (8 bits, bias=127)	Fraction (23 bits)
0	10000011	10100000000000000000000

再請針對十進位浮點小數-17.125 表達成 IEEE-754 格式後，寫出其 Exponent 部位「最後 4 bits 的二進位值」。最後請將「十六進位資料 C0F80000」，表示成 IEEE-754 單倍精確度浮點數格式，寫出 Fraction 部位「最前面 4 bits 的二進位值」，和此「十六進位資料」的「十進位浮點小數」。(25 分)

- | |
|---|
| 1. 《考題難易》：★★ (最難 5 顆星) |
| 2. 《解題關鍵》：IEEE-754 單倍精確度 (32 bit) 浮點數計算 |

【擬答】：

(一) S=0 為正

$$E=10000011=131$$

$$\therefore \text{真正指數為 } 131-127=4$$

$$\text{故為 } 1.101 \times 2^4 = 1.625 \times 2^4 = 26$$

(二) $17.125=10001.001=1.0001001 \times 2^4$

$$\therefore \text{Exponent}=4+127=131=10000011$$

故最後 4bits 為 0011

(三) $C0F80000=11000000111110000000000000000000$

$$S=1$$

$$E=10000001=129$$

$$F=111100000000000000000000$$

$$\therefore \text{Fraction 前 4bits 為 } 1111$$

$$\text{又其真正指數為 } 129-127=2$$

$$\text{故十進位數為 } -1.1111 \times 2^2 = -7.75$$

二、臺灣科技公司規劃網路，其中財務部門使用 C 等級 (class C) 的網路系統並手動設定 IP，子網路遮罩為 255.255.255.224。其中，財務主管小明的 IP 為 140.123.100.89，員工佳佳的 IP 為 140.123.100.182。請寫出小明的網路位址、廣播位址 (以十進位表示)；以及說明財務部門的網路可以切割成幾個子網路、每個子網路設定可用主機電腦數量。最後，請說明手動設定電腦 IP 位址時，子網路遮罩 (Subnet Mask) 與預設開道的功能。(25 分)

- | |
|-------------------------|
| 1. 《考題難易》：★★★ (最難 5 顆星) |
| 2. 《解題關鍵》：IP 位址分析 |

【擬答】：

$$\text{小明的 IP 為 } 140.123.100.89=10001100.01111011.01100100.01011001$$

$$\text{子網路遮罩：} 255.255.255.224=11111111.11111111.11111111.11100000$$

公職王歷屆試題 (113 國安局特考)

10001100.01111011.01100100.01011001
and 11111111.11111111.11111111.11100000

10001100.01111011.01100100.01000000
∴網路位址為 140.123.100.64

10001100.01111011.01100100.01011001
Or 00000000.00000000.00000000.00011111

10001100.01111011.01100100.01011111

∴廣播位址為 140.123.100.95

子網位數為 3 位元，因此可以劃分 $2^3=8$ 個子網路

每個子網路可以容納 $2^5-2=30$ 台主機

子網路遮：

用來區分 IP 位址的網路部分和主機部分，並可判斷 IP 是否贊同一子網路下。

預設閘道：

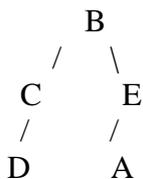
當主機發送的封包目標位址不在同一區段時，封包會發送到預設閘道，由它來轉封包到其他區段或外部網路

三、某城市有景點 A~E，其連接的道路均為單行道，單行道起點、終點與距離的資訊為：A->B (20)、A->C (25)、A->D (11)、B->C (25)、B->E (18)、C->D (27)、C->E (12)、D->B (24)、E->A(22)。請畫出"起始點為 B"之廣度優先走訪樹狀圖，其中每一個景點只出現一次。另外請畫出根節點"起始點為 B"，其他景點為葉節點的展開樹，找出 B->D 景點的所有路徑與最短路徑，以及所經過之景點及其距離。最後，請說明上面最短路徑走訪方法的演算步驟。(25 分)

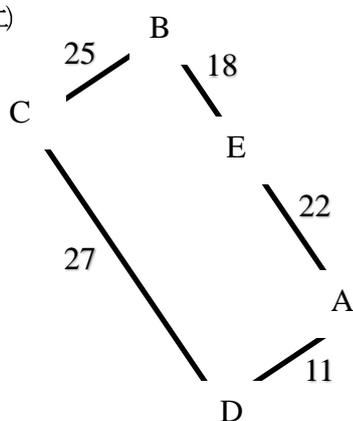
- 1. 《考題難易》：★★★ (最難 5 顆星)
- 2. 《解題關鍵》：最短路徑演算法

【擬答】：

(一) BFS



(二)



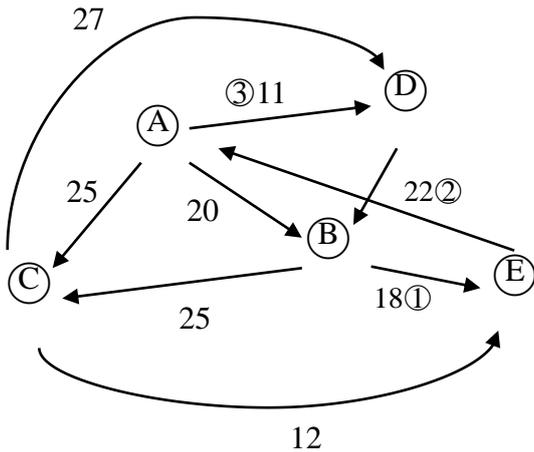
B→D

所有路徑：

1. B→C→D 距離為 25+27=52

2. B→E→A→D 距離為 18+22+11=51

最短路徑為 B→E→A→D



Step1

$B \rightarrow C \Rightarrow 25$

$B \rightarrow E \Rightarrow 18 \quad \therefore B \rightarrow E \Rightarrow 18$

Step2

$E \rightarrow A \Rightarrow 22$

Step3

$A \rightarrow C \rightarrow D \Rightarrow 25+27=52$

$A \rightarrow D \Rightarrow 11 \quad \therefore A \rightarrow D \Rightarrow 11$

故 $B \rightarrow D$ 最短路徑為

$B \rightarrow E \rightarrow A \rightarrow D$

四、請完成計算中位數與眾數兩個方法(I)~(V)空格程式，使程式正常運作，以輸出 Mean=1, Mode=5。其中，中位數是將一組數值資料由小到大排列後，取最中間的數值為中位數。若資料個數為奇數，則取最中間的數值為中位數，例如 1, 2, 3, 3, 4, 6, 7, 7, 9 的中位數是 4；若資料個數為偶數，則取最中間兩個數值的平均為中位數，例如 1, 2, 3, 3, 4, 6, 7, 7, 8, 9 的中位數是 $(4+6)/2=5$ 。眾數，則是指一組數字中出現次數最頻繁的數字。例如，2、3、3、5、7、10 的眾數是 3。最後請說明此程式是否有不安全的地方，以及其原因。(25 分)

<pre> #include <iostream> #include <algorithm> using namespace std; class Compute { public: void setData(int *, int); double getMedian(); double getMode(); private: int *data, d_size; // 取得 key 在 map 中的索引 int getIndex(int d[2][20], int key, int c) { for (int i=0; i<c; i++) if (key==d[0][i]) return i; return -1; } //取得最大頻率值的 key int getMaxValue(int d[2][20], int c) { int maxKey = d[0][0]; int maxValue = d[1][0]; for (int i=1; i<c; i++) if (maxValue<d[1][i]) { maxKey = d[0][i]; maxValue = d[1][i]; } return maxKey; } }; //設定資料並完成排序 void Compute::setData(int *d, int s) { data = d; d_size = s; sort(data, data+d_size); } </pre>	<pre> // 求中位數，若數量為奇數，回傳中間值，否則回傳中間平均值 double Compute::getMedian() { if (d_size%2==<u>(I)</u>) // (I) return data[<u>(II)</u>]; // (II) else return (data[(d_size/2)-1]+<u>(III)</u>)/2.0; // (III) } //求眾數，統計數列各值出現頻率，找最大 double Compute::getMode() { //d_map[0]紀錄 Key, d_map[1]紀錄頻率值 int d_map[2][20]={0}; //d_map 數量 int map_count=0; for (int i=0; i<d_size; i++) { int index = getIndex(d_map, data[i], map_count); if (index==<u>(IV)</u>) { // (IV) d_map[0][map_count]=data[i]; d_map[1][map_count++]=1; } else d_map[1][<u>(V)</u>]=d_map[1][index]+1; // (V) } return getMaxValue(d_map, map_count); } int main() { int data[10] = {5, 3, -4, 3, -1, 5, -1, -3, 5, -2}; Compute p; p.setData(data, 10); cout<<"Mean="<<p.getMedian()<<" "; cout<<"Mode="<<p.getMode()<<endl; } </pre>
--	--

1. 《考題難易》：★★★ (最難 5 顆星)

2. 《解題關鍵》：程式語言

【擬答】：

(一) (I) 為 1；(II) 為 $d_size / 2$ ；(III) 為 $data[d_size / 2]$ ；(IV) 為 -1；(V) 為 index

(二) 不安全的地方為缺乏對無效輸入的檢查。

公
職
王