

A / KHÁI NIỆM CHUNG

I / KHAI NIỆM VỮ QUI :

Một đối tượng gọi là có tính đệ qui nếu nó đ-ợc định nghĩa thông qua chính nó .

Một hàm , một thủ tục có tính đệ qui nếu trong thân ch-ơng trình của hàm , thủ tục này lại có lời gọi tới chính nó .

Thí dụ 1:

Định nghĩa giai thừa của một số nguyên không âm là định nghĩa có tính đệ qui. Thật vậy:

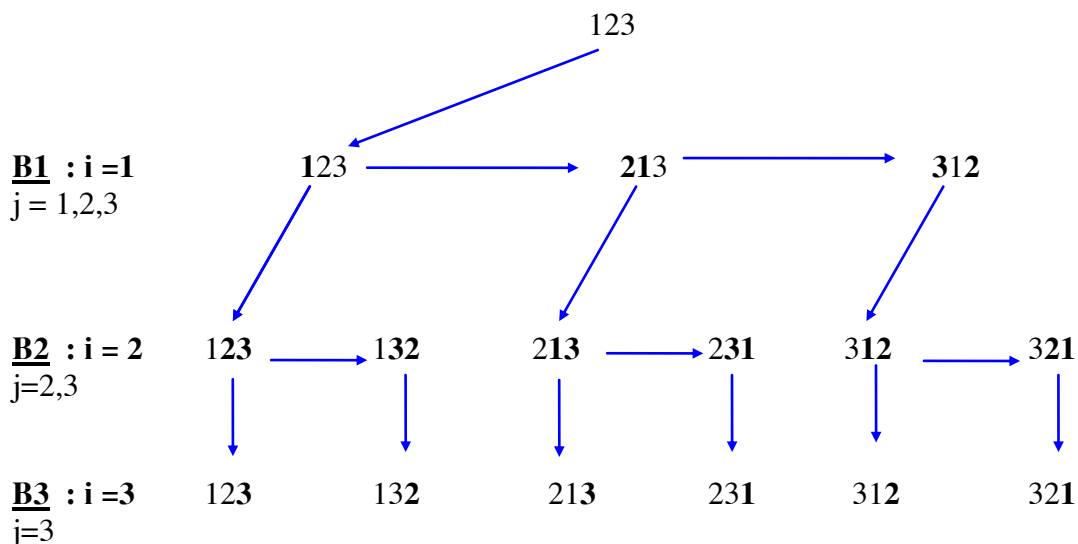
$$(N)! = \begin{cases} 1 & \text{Nếu } N=0 \\ N * (N-1)! & \text{Nếu } N>0 \end{cases}$$

Để định nghĩa N giai thừa , phải thông qua định nghĩa giai thừa (của $N-1$).

Thí dụ 2:

Xây dựng hoán vị của N phần tử cũng có tính chất đệ qui . Thật vậy :

Giả sử có 1 hoán vị là $S(A_1, A_2, \dots, A_{i-1}, A_i, \dots, A_{n-1}, A_n)$, sau đó đổi chỗ 2 phần tử $S[i]$ và $S[j]$ của hoán vị đó ta sẽ đ-ợc một hoán vị mới .Sau đây là sơ đồ hình thành dần các hoán vị tiếp theo nhau của hoán vị $S(1,2,3)$



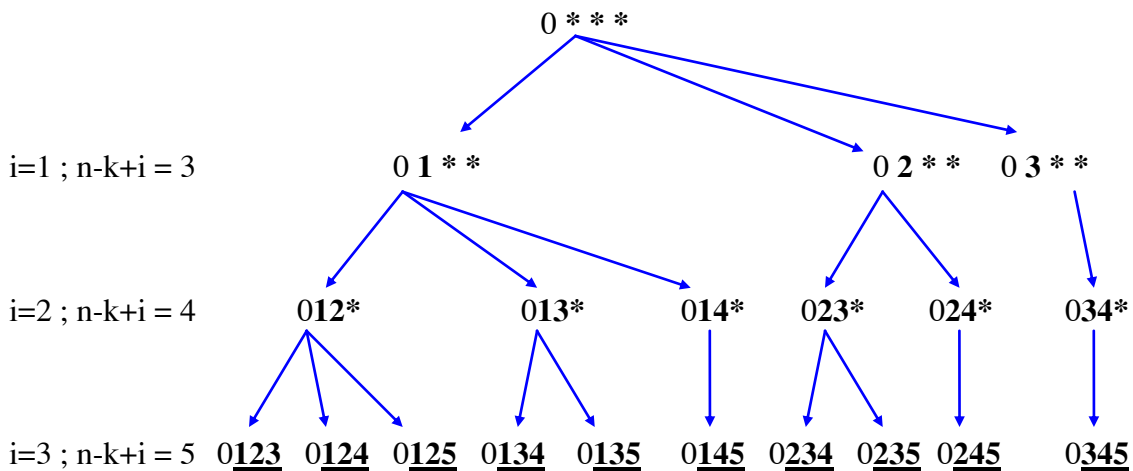
Vậy để xây dựng các hoán vị sau ta phải dựa vào các hoán vị đã sinh ra trước đó.

Thí dụ 3: Xây dựng tổ hợp chập K của N phần tử 1,2,3,...,N cũng theo phương thức đệ qui :

Ta sẽ xây dựng dần từng phần tử từ vị trí thứ 1 đến vị trí thứ K của tổ hợp. Để xây dựng phần tử thứ i (sau khi đã xây dựng xong các phần tử từ 1 đến i-1 của tổ hợp này), ta sẽ cho phần tử thứ i nhận 1 trong các giá trị từ $(A_{i-1} + 1)$ đến giá trị cao nhất có thể được của nó đó là giá trị $(N-K)+i$ vì sau phần tử thứ i này còn $(K-i)$ phần tử, do đó nếu phần tử thứ i nhận giá trị cao nhất là $(N-K)+i$ thì các phần tử tiếp theo vẫn còn khả năng nhận các giá trị : $(N-K)+i+1$, $(N-K)+i+2$, ..., $(N-K)+i+(K-i) = N$.

Vậy để xây dựng phần tử thứ i của 1 tổ hợp, ta phải dựa vào kết quả đã xây dựng tới phần tử thứ i-1. Tất nhiên để xây dựng phần tử thứ 1, ta phải dựa vào 'phần tử hàng rào' là phần tử ở vị trí thứ '0', ta gán cho phần tử này giá trị nào cho phù hợp qui luật nêu trên? rõ ràng đó là giá trị 0, nhằm cho nó quyền được bình đẳng như mọi phần tử khác. Phần tử 0 này chịu một trách nhiệm rất nặng nề, bắt đầu từ nó mới xây dựng dần được các phần tử tiếp theo của mọi tổ hợp, song ta cũng đừng quên nó phải 'ngâm ngùi' vì 'không được đứng trong tổ hợp'.

Sau đây là sơ đồ minh họa việc xây dựng tổ hợp chập 3 của 5 phần tử 1,2,3,4,5



II / L- U □ VŨ TH □ T □ C VÀ HÀM □ □ QUI :

L- u ý 1 + Trong thủ tục và hàm đệ qui cần chứa các lệnh thể hiện tính dừng của đệ qui. Nghĩa là các thủ tục, hàm đệ qui chỉ gọi tới chính nó một số hữu hạn lần rồi gặp điều kiện thoát (để nó không gọi tới chính nó nữa)

Thí dụ 1 :

```
Function Giaithua(N: Byte) : LongInt;
Begin
    If N=0 then giaithua := 1
    Else
        Giaithua := N*Giaithua(N-1);
End;
```

Trong hàm Giaithua , điều kiện dừng là $0! = 1$, vì mỗi lần gọi tới hàm Giaithua thì N giảm đi 1 đơn vị nên sẽ dẫn tới trường hợp $N=0$.

Thí dụ 2 :

```
Function Fibonaci(N : Integer) : LongInt;
Begin
  If (N=1) or (N=2) then Fibonaci := 1
  Else
    Fibonaci:= Fibonaci(N-1)+ Fibonaci(N-2);
End;
```

Trong hàm Fibonaci , điều kiện dừng là :

If (N=1) or (N=2) then Fibonaci := 1

vì mỗi lần gọi tới hàm Fibonaci thì N giảm đi 1 , sẽ dẫn tới tình trạng $N=3$
 $\Rightarrow \text{Fibonaci}(3) = \text{Fibonaci}(2) + \text{Fibonaci}(1) = 1 + 1 = 2$.

L- u ý 2 Thủ tục và hàm đệ qui phải thể hiện tính đệ qui : Nó gọi tới chính nó
 Trong 2 thí dụ nêu trên các lệnh

Giaithua := N*Giaithua(N-1); { Thí dụ 1 }

hoặc

Fibonaci:= Fibonaci(N-1)+ Fibonaci(N-2); { Thí dụ 2 }

thể hiện tính đệ qui .

III / MỘT SỐ BÀI TẬP CỐ BỔN :

Bài 1 : Xây dựng các hoán vị của tập N phần tử $1,2,3,...,N$ bằng đệ qui :

Bài 2 : Xây dựng các tổ hợp chập K của N phần tử $1,2,3,...,N$ ($0 < K < N$)

Bài 3 : Xây dựng các chỉnh hợp chập K của N phần tử $1,2,3,...,N$ ($0 < K < N$)

Bài 4 : Xây dựng các chỉnh hợp lặp chập K của N phần tử $1,2,3,...,N$ ($0 < K < N$) (còn gọi là bộ mẫu N phần tử)

IV / BÀI TẬP VỮNG NHÀ

Bài 5 : Tạo xâu kí tự có độ dài không quá 20 , chỉ chứa 3 kí tự A,B,C có tính chất : Không có 2 xâu con liên nhau bằng nhau

Gợi ý :

+ Xây dựng hàm KT kiểm tra 2 xâu con liên nhau có bằng nhau không ?

+ Giả sử đã tạo được xâu A có $i-1$ kí tự , chọn kí tự thứ i là 1 trong 3 kí tự A,B,C nối thêm vào xâu A mà A vẫn thoả mãn KT thì tìm tiếp kí tự $i+1$, nếu không thoả mãn thì xâu A trở lại nh- tr- ớc (có $i-1$ kí tự cũ) để chọn kí tự thứ i của xâu là 1 trong 2 kí tự còn lại

Bài 6 :

Lập trình thể hiện trò chơi Tháp Hà Nội : Trên cọc 1 có N đĩa và xếp đĩa nhỏ ở trên đĩa lớn ; cọc 2 và cọc 3 ch- a có đĩa . Hãy chuyển hết đĩa ở cọc 1 sang cọc 3 theo qui luật sau :

Chuyển từng đĩa ở trên cùng của một trong 3 cọc sang cọc khác sao cho đĩa lớn không đặt trên đĩa nhỏ .

Gợi ý :

- + Nếu cọc 1 chỉ có 1 đĩa thì chuyển nó sang cọc 3
- + Giả sử đã giải đ- ợc bài toán trong tr- ờng hợp có N-1 đĩa ; không mất tính chất tổng quát ,ta giả sử cọc 2 chứa N-1 đĩa (đĩa nhỏ trên đĩa lớn) và sẽ chuyển hết đ- ợc sang cọc 3 nhờ cọc trung gian là cọc 1 .Ta sẽ chứng minh bài toán cho N đĩa xếp ở cọc 1 , chuyển sang cọc 3 nhờ cọc trung gian là cọc 2 sẽ giải đ- ợc. Thật vậy :
 - a) Tìm cách chuyển N-1 đĩa từ cọc 1 sang cọc 2 (cọc phụ : 3);
 - b) Chuyển 1 đĩa còn lại (đĩa lớn nhất) ở cọc 1 sang cọc 3
 - c) Tìm cách chuyển N-1 đĩa từ cọc 2 sang cọc 3 (cọc phụ là cọc 1)

Bài 7 :

Lập trình bài toán : Tính số cách chia M vật thành N phần theo qui luật :

$$S_1 \geq S_2 \geq \dots \geq S_{N-1} \geq S_N \geq 0 \quad (S_i \text{ là số vật của phần thứ } i)$$

$$\sum_{i=1}^N S_i = M$$

Gợi ý : + Nếu số đồ vật $M=0$ thì coi nh- có 1 cách chia : đó là cách chia mỗi ng- ời không đ- ợc vật nào .

+ Nếu số ng- ời $N=0$ thì không thể chia đ- ợc

+ Nếu $0 < M < N$ thì trong mọi cách chia , luôn có ít nhất N-M ng- ời không đ- ợc chia , do vậy các cách chia khác nhau ở chỗ : chia có khác nhau cho M ng- ời còn lại hay không ? Nói cách khác số cách chia trong tr- ờng hợp này bằng số cách chia của bài toán chia M vật cho M ng- ời .

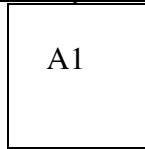
+ Nếu $M \geq N > 0$ thì các cách chia thuộc 2 loại :

Loại 1 : Mọi ng- ời đều có phần , vậy mọi cách chia có chỗ giống nhau là mọi ng- ời đều có ít nhất 1 vật , các cách chia chỉ khác nhau ở chỗ phân chia M-N vật còn lại cho N ng- ời nh- thế nào ?

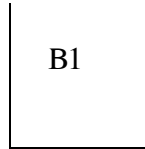
Loại 2 : Có 1 ng- ời không đ- ợc chia vật nào . Nghĩa là chỉ chia M vật cho N-1 ng- ời

Bài 8 : Vẽ các đ- ờng HilBert cấp 5 , biết các đ- ờng HilBert cấp 1, cấp 2, cấp 3 nh- hình vẽ d- ới đây :

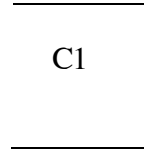
Các đ- ờng cấp 1



A1



B1



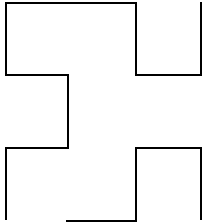
C1



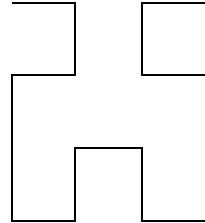
D1

Các đ-ờng cấp 2

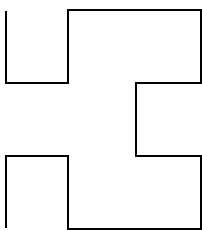
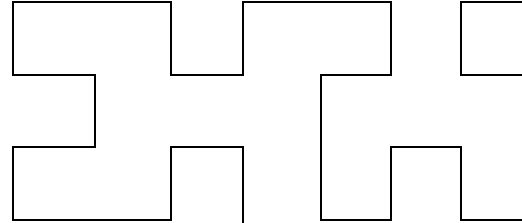
Đ-ờng A3



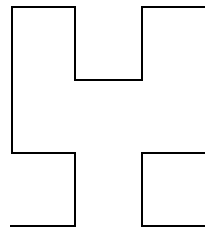
A2



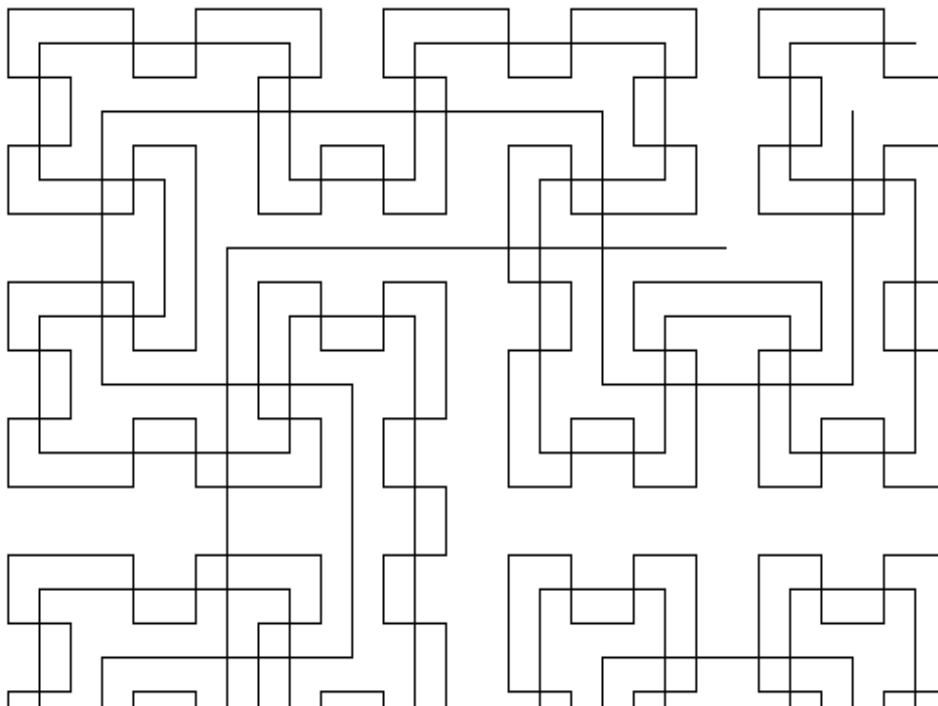
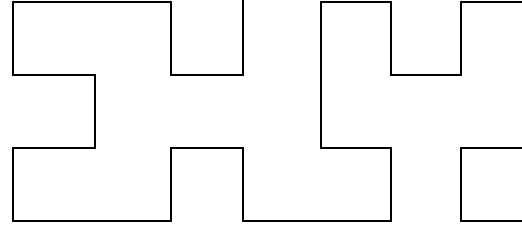
B2



C2



D2



Đ-ờng A5

Bài 1:

```
Uses  Crt;
Const  N      = 8;
       TF     = 'hoanvi.txt';
Type   TS     = String[N];
Var    S      : TS;
       d,Lt   : Longint;
       F      : Text;
       T      : LongInt Absolute $0000:$046C;
Procedure Doi(Var a,b : Char);
  Var p : Char;
  Begin
    p := a; a := b; b := p;
  End;
Procedure Hien(S : TS);
  Begin
    Inc(d); Write(F,S,' ');
    If (d mod 10 = 0) then Writeln(F);
  End;
Procedure Tao(S : String;i : Byte);
  Var  j      : Byte;
       p      : Char;
  Begin
    If i=N then Hien(S);
    For j:=i to N do
      Begin
        Doi(S[i],S[j]);
```

```

        Tao(S,i+1);
    End;
End;
BEGIN
    Clrscr;
    S := '123456789';
    S := Copy(S,1,N);
    d := 0;
    LT := T;
    Assign(F,TF);
    ReWrite(F);
    Tao(S,1);
    Close(F);
    Writeln(#13#10,'So hoan vi la  ',d);
    Writeln('Mat thoi gian la  ',((T-Lt)/18.2):10:2,' giay');
    Readln;
END.

```

Ch-ong trình trên chạy trên máy DX2-486 , N =8 , mất thời gian khoảng 4 giây .
N= 9 , mất khoảng 37 giây .

Bài 2 :

```

Uses Crt;
Var    X      : Array[0..20] of Byte;
        K,N    : Byte;
        C      : LongInt;
Procedure Init;
    Begin
        Write('k,n = ');
        Readln(k,n);
        X[0]   := 0;
        C      := 0;
    End;
Procedure Inkq;
    Var i : Byte;
    Begin
        Inc(C);
        Write(C:5,' : ');
        For i:=1 to k do Write(x[i]:3);
        Writeln;
    End;
Procedure Thu(i : Byte);
    Var j : Byte;
    Begin
        For j:= x[i-1]+1 to n-k+i do
            Begin
                x[i] := j;
                If i= k then Inkq Else Thu(i+1);
            End;
    End;

```

```

        End;
BEGIN
    Clrscr;
    Init;
    Thu(1);
    Readln;
END.

```

Bài 3 :

```

Uses Crt;
Var
    Cx      : Array [1..10] of Boolean;
    A       : Array [1..10] of Byte;
    N,k     : Byte;
    dem     : LongInt;
Procedure Nhap;
Begin
    Write('NHap N,k : ');
    Readln(N,k);
End;
Procedure Tao;
Begin
    Fillchar(Cx,Sizeof(Cx),True);
    dem := 0;
End;
Procedure Hien;
    Var j : Byte;
Begin
    Inc(dem);Write(dem:5,' : ');
    For j:=1 to k do Write(a[j]:3);
    Writeln;
End;
Procedure Try(i : Byte);
    Var j : Byte;
Begin
    For j:=1 to n do
        If Cx[j] then
            Begin
                A[i]:=j;
                Cx[j]:=False;
                If i=k then Hien Else Try(i+1);
                Cx[j]:=True;
            End;
    End;
End;

Begin
    Clrscr;
    Nhap;

```



```

    Tao;
    Try(1);
    Readln;
End.

```

Bài 4 :

```

Uses Crt;
Const Max = 20;
Var X      : Array[0..Max] of Byte;
    K,N    : Byte;
    dem    : LongInt;
Procedure Init;
Begin
    Write('k,n (k<=n) = ');
    Readln(k,n);
    X[0] := 0;
    dem := 0;
End;
Procedure Inkq;
Var i : Byte;
Begin
    Inc(dem);
    Write(dem:10,' : ');
    For i:=1 to k do Write(x[i]:2);
    Writeln;
End;
Procedure Thu(i : Byte);
Var j : Byte;
Begin
    For j:= 1 to n do
        Begin
            x[i] := j;
            If i = k then Inkq Else Thu(i+1);
        End;
    End;
END.
BEGIN
    Clrscr;
    Init;
    Thu(1);
    Readln;
END.

```

Bài 5 :

```

Uses Crt;
Const N = 20;
Var S : String;
Function Kt(S : String) : Boolean;
    Var i,j : Byte;

```

```

Begin
  Kt := True;
  For i:=1 to Length(S) div 2 do
    For j:=1 to Length(S)- 2*i+1 do
      If Copy(S,j,i)=Copy(S,j+i,i) then
        Begin
          Kt := False;
          Exit;
        End;
    End;
  End;
Procedure Tao(S : String);
  Var ch : Char;
  Begin
    If Length(S)=N then
      Begin
        Writeln(S);

        Readln;
        Halt;
      End;
    For ch:='A' to 'C' do    { Khởi tạo mọi khả năng }
      Begin
        S := S+ch;    { Thử chọn 1 khả năng }
        If Kt(S) then Tao(S) {Nếu thoả mãn điều kiện thì tìm tiếp }
        Else Delete(S,Length(S),1); {Nếu không thì trả về trạng thái cũ}
      End;
    End;
BEGIN
  Clrscr;
  S := "";
  Tao(S);
END.

```

Bài 6 :

```

Uses  Crt;
Const  C1    = '1';
        C2    = '2';
        C3    = '3';
        Max   = 20;
Var    Sodia,i,h1,h2,h3 : Byte;
        A,B,C : Array[1..100] of Byte;
Procedure Khoitri;
  Begin
    Write('Nhap so luong dia (<=20) : ');
    Repeat
      {$I-} Readln(Sodia); {$I+}
    Until (IoResult=0) and (sodia<=Max) and (Sodia>0);
  End;

```

```

Textcolor(14);
For i:=sodia downto 1 do
  Begin
    Gotoxy(40,24-i);
    Writeln('*');
  End;
Textcolor(12);
For i:=sodia downto 1 do
  Begin
    Gotoxy(50,24-i);
    Writeln('*');
  End;
Textcolor(9);
For i:=sodia downto 1 do
  Begin
    Gotoxy(60,24-i);
    Writeln('*');
  End;
{ Readln; }
Textcolor(15);
For i:=sodia downto 1 do
  Begin
    Gotoxy(40,24-i);
    Writeln((sodia-i+1):2);
    A[i] := sodia-i+1;
    B[i] := 0;
    C[i] := 0;
  End;
{ Readln; }
h1 := sodia;
h2 := 0;
h3 := 0;
End;
Procedure Hien(X,Y : Char);
Begin
  Case X of
    '1' : Begin
      Gotoxy(40,24-h1);
      Textcolor(14);Write('*');Textcolor(15);
      Case Y of
        '2' : Begin
          Inc(h2);B[h2] :=A[h1];
          Gotoxy(50,24-h2); Write(B[h2]:2);
        End;
        '3' : Begin
          Inc(h3);C[h3] := A[h1];
          Gotoxy(60,24-h3); Write(C[h3]:2);
        End;
      End;
    End;
  End;
End;

```

```

        End;
        Dec(h1);
    End;
'2' : Begin
    Gotoxy(50,24-h2);
    Textcolor(12);Write('*');Textcolor(15);
    Case Y of
        '1': Begin
            Inc(h1);A[h1] := B[h2];
            Gotoxy(40,24-h1); Write(A[h1]:2);
        End;
        '3' : Begin
            Inc(h3);C[h3] := B[h2];
            Gotoxy(60,24-h3); Write(C[h3]:2);
        End;
    End;
    Dec(h2);
End;
'3' : Begin
    Gotoxy(60,24-h3);
    Textcolor(9);Write('*');Textcolor(15);
    Case Y of
        '1': Begin
            Inc(h1);A[h1] := C[h3];
            Gotoxy(40,24-h1); Write(A[h1]:2);
        End;
        '2' : Begin
            Inc(h2);B[h2] :=C[h3];
            Gotoxy(50,24-h2); Write(B[h2]:2);
        End;
    End;
    Dec(h3);
End;
End;
End;
Procedure Chuyen(N : Byte;A,B,C : Char);
Begin
    If N=1 then { Writeln('Chuyen ',A,' --> ',C); }
    Begin Hien(A,C);{ Readln; }End
    Else
    Begin
        Chuyen(N-1,A,C,B);
        Chuyen(1,A,B,C);
        Chuyen(N-1,B,A,C);
    End;
End;
BEGIN
    Repeat

```

```

    Clrscr;
    Khoitri;
    Chuyen(sodia,C1,C2,C3);
    Gotoxy(1,24);Writeln('ESC : thoat ');
    Until ReadKey=#27;
END.

```

Bài 7 :

```

Uses  Crt;
Var   M,N,sc : LongInt;
Procedure Nhap;
    Begin
        Write('Nhap so do vat : ');
        Readln(M);

        Write('Nhap so nguoi : ');
        Readln(N);
    End;
Function Chia(M,N : LongInt) : LongInt;
    Begin
        If M=0 then Chia := 1
        Else {M>0}
            If N=0 then Chia := 0
            Else {N>0}
                If M<N then Chia := Chia(M,M)
                Else
                    Chia := Chia(M-N,N)+Chia(M,N-1);
    End;
BEGIN
    Clrscr;
    Nhap;
    sc := Chia(M,N);
    If sc=0 then
        Begin
            Writeln('Khong the chia cho 0 nguoi ');
            Readln;
            Halt;
        End
    Else Writeln('So cach chia la : ',sc);
    Readln
END.

```

Bài 8 :

```

Uses  Crt,graph;
Const N      = 4;
      h0     = 512;

```

```

Var   i,h,x,y,x0,y0  : Integer;
      Gd, Gm          : Integer;
Procedure D(i:integer);forward;
Procedure B(i:integer);forward;
Procedure C(i:integer);forward;
Procedure A(i:integer);forward;
Procedure A;
  Begin
    If i>0 then
      Begin
        D(i-1); x:=x-h; lineto(x,y);
        A(i-1); y:=y-h; lineto(x,y);
        A(i-1); x:=x+h; lineto(x,y);
        B(i-1);
      End
    End;
  End;

```

```

Procedure B;
  Begin
    If i>0 then
      Begin
        C(i-1); y:=y+h; lineto(x,y);
        B(i-1); x:=x+h; lineto(x,y);
        B(i-1); y:=y-h; lineto(x,y);
        A(i-1);
      End
    End;
  End;

```

```

Procedure C;
  Begin
    If i>0 then
      Begin
        B(i-1); x:=x+h; lineto(x,y);
        C(i-1); y:=y+h; lineto(x,y);
        C(i-1); x:=x-h; lineto(x,y);
        D(i-1);
      End
    End;
  End;

```

```

Procedure D;
  Begin
    If i>0 then
      Begin
        A(i-1); y:=y-h; lineto(x,y);
        D(i-1); x:=x-h; lineto(x,y);
        D(i-1); y:=y+h; lineto(x,y);
        C(i-1);
      End
    End;
  End;

```

```
BEGIN
  Gd := Detect; InitGraph(Gd, Gm, 'C:\tp97\tp\bgi');
  If GraphResult <> grOk then Halt(1);
  i:=0;
  h:=h0;
  x0:=h div 2;
  y0:=x0;
  Repeat
    inc(i);
    h:=h div 2;
    x0:=x0+(h div 2);
    y0:=y0+(h div 2);
    x:=x0;
    y:=y0;
    Moveto(x,y);
    A(i);

  Until i=n;
  Readln;
  CloseGraph;
END.
```

Chú ý : Ch- ơng trình trên dùng đệ qui gián tiếp (với từ *ForWard*)

Thủ tục *D* gọi tới các thủ tục *A* và *C* ở d- ới nó

Thủ tục *B* gọi tới các thủ tục *C* và *A* ở d- ới nó

Ngoài ra , để dùng các lệnh vẽ (chế độ đồ hoạ) ta sử dụng *Unit Graph* .

B / QUAY LUI + VẾT CẠN + LỰA CHỌN TỐI ƯU
KẾT HỢP ĐỆ QUI

I / Ý nghĩa :

Trong nhiều trường hợp , nghiệm của bài toán là dãy các phần tử được xác định không theo một luật tính toán nhất định, muốn tìm nghiệm phải thực hiện từng bước , tìm kiếm dần từng phần tử của nghiệm .Để tìm mỗi phần tử ,phải kiểm tra “đúng,sai” các khả năng có thể chấp nhận của phần tử này.

+ Nếu khả năng nào đó không dẫn tới giá trị chấp nhận được của phần tử đang xét thì phải loại bỏ khả năng đó , chuyển sang chọn khả năng khác (ch- a được chọn) . Chú ý : mỗi khi chọn một khả năng cho một phần tử thì thông tin trạng thái bài toán sẽ thay đổi vì thế khi chuyển sang chọn khả năng khác , phải trả lại trạng thái nh- tr-ớc khi chọn khả năng vừa loại bỏ (nghĩa là phải quay lui lại trạng thái cũ).

+ Nếu có 1 khả năng chấp nhận được (nghĩa là gán được giá trị cho phần tử đang xét của nghiệm) và ch- a là phần tử cuối cùng thì tìm tiếp phần tử tiếp theo .

+ Nếu bài toán yêu cầu chỉ tìm 1 nghiệm thì sau khi chọn được 1 khả năng cho 1 phần tử của nghiệm , ta kiểm tra phần tử này đã là phần tử cuối cùng của 1 nghiệm hay ch- a (*gọi là lệnh kiểm tra kết thúc 1 nghiệm*). Nếu đúng là phần tử cuối cùng của nghiệm thì : Hiện nghiệm và thoát hẳn khỏi thủ tục đệ qui bằng lệnh **Halt**;

Nếu bài toán yêu cầu tìm tất cả các nghiệm thì không có lệnh kiểm tra kết thúc 1 nghiệm

+ Trong việc thử mọi khả năng của 1 phần tử của nghiệm , nếu biết tìm những điều kiện để nhanh chóng loại bỏ những khả năng không thể chấp nhận được thì việc thử sẽ nhanh chóng hơn. Việc thử mọi khả năng của 1 phần tử của nghiệm cũng giống như một người đi đường , mỗi khi đến ngã N-đ-ờng , lần lượt chọn 1 đường thích hợp trong các con đường của ngã N-đ-ờng đó , nếu biết chắc chắn những đường nào đó trong các đường của ngã N-đ-ờng là đường “cụt” không thể đi tới đích thì người đi đường sẽ loại ngay những đường đó ; hoặc ngược lại nếu nhìn thấy trước những điều kiện cho phép chỉ cần đi theo một số con đường nhất định trong N đ-ờng mà vẫn tới đích nhanh chóng thì người đi đường sẽ dùng những điều kiện ấy như “la bàn “ chỉ phương hướng đi của mình .Tất nhiên khi khẳng định điều này là “đúng” ,điều kia là “sai” phải hết sức thận trọng.Nếu những khẳng định “ chắc chắn” chỉ là điều “ngộ nhận” thì có thể bỏ sót một số con đường tới đích, hoặc chệch hướng không thể tới đích . Một trí khôn vừa “táo bạo” vừa “chắc chắn” là trí khôn của một chương trình sáng giá !

+ Nếu tìm 1 nghiệm tốt nhất (theo điều kiện) thì mỗi khi tìm được 1 nghiệm , ta so sánh với nghiệm tốt nhất đã tìm được cho đến lúc này(gọi là nghiệm tối - u) . Nếu nghiệm vừa tìm được tốt hơn nghiệm tối - u thì gán lại nghiệm tối - u là nghiệm mới . Quá trình tiếp diễn cho đến khi duyệt hết các nghiệm của bài toán ta sẽ được nghiệm tối - u của bài toán .

Tóm lại thuật toán “duyet trên cơ sở tìm kiếm và quay lui ” - Thuật toán BackTracking - có chứa các nội dung sau :

+ Vết cạn mọi nghiệm bằng tìm kiếm tiến dần về đích đồng thời biết quay lui khi không thể tiến

- + Có thể đặt các “mắt lọc” để việc tìm kiếm nhanh chóng hơn : hoặc loại bỏ hoặc chỉ chọn một số h- ớng .
- + Có thể so sánh các nghiệm để có nghiệm tối - u
- + Tùy theo yêu cầu , có thể chỉ tìm 1 nghiệm , cũng có thể tìm mọi nghiệm

Do thuật toán BackTracking xây dựng trên cơ sở tìm kiếm dần ,kết quả sau hình thành từ kết quả tr- ớc, nên có thể dùng các hàm, thủ tục đệ qui để thực hiện thuật toán Cụ thể có 3 dạng dần bài th- ờng gặp sau đây :

II / Ba dạng đệ qui th- ờng gặp để thực hiện thuật toán BackTracking

DẠNG 1 : Tìm mọi nghiệm

```

Procedure Tim(k : Integer);
Begin
  Vòng lặp đệ cử mọi khả năng của b- ớc thứ k trong tìm kiếm 1 nghiệm
  Begin
    + Thử chọn 1 đề cử cho b- ớc k
    + Nếu đề cử này chấp nhận đ- ợc thì
      Begin
        * Ghi nhận giá trị đề cử;
        * L- u trạng thái mới của bài toán sau đề cử;
        * Nếu ch- a phải b- ớc cuối cùng thì Tim(K+1)
        Else {là b- ớc cuối cùng} thì Hiện Nghiệm;
        * Trả lại trạng thái của bài toán tr- ớc khi đề cử;
      End;
    End;
  End;
End;

```

Cũng có thể viết d- ới dạng sau :

```

Procedure Tim(k : Integer);
Begin
  Nếu b- ớc k là b- ớc sau b- ớc cuối cùng thì Hiện nghiệm ;
  Vòng lặp đệ cử mọi khả năng của b- ớc thứ k trong tìm kiếm 1 nghiệm
  Begin
    + Thử chọn 1 đề cử cho b- ớc k
    + Nếu đề cử này thoả mãn bài toán thì
      Begin
        * Ghi nhận giá trị đề cử;
        * L- u trạng thái mới của bài toán sau đề cử;
        * Tim(k+1);
        * Trả lại trạng thái của bài toán tr- ớc khi đề cử;
      End;
    End;
  End;
End;

```

End;
End;

Thí dụ : Bài toán con mã đi tuần (Hiện tất cả các nghiệm)

Cách 1 :

```

Program Madequy;
Uses      Crt;
Const     Max   = 8;
          Fi     = 'madq.inp';
          D      : Array [1..8] of -2..2 = (-2,-2,-1,1,2,2,1,-1);
          C      : Array [1..8] of -2..2 = (-1,1,2,2,1,-1,-2,-2);

Var
  F      : Text;
  T1,T2  : longint;
  A      : Array[1..Max,1..Max] of Integer;
  x,y,k,dem,n,nsq : Integer;
Procedure DocFi;
Begin
  Assign(F,Fi);
  {$I-} Reset(F); {$I+}
  If Ioresult<>0 then
    Begin Writeln('Loi File '); Readln; Halt; End;
  Readln(F,N);
  Nsq := N*N;
  Readln(F,x,y);
  Close(F);
End;
Procedure Hien;
Var i,j : Integer;
Begin
  Inc(dem);
  Assign(F,Fi);
  Append(F); { Ghi nghiệm ngay cuối File dữ liệu Input }
  Writeln(F,'Nghiem thu ',dem);
  For i:=1 to N do
    Begin
      For j:=1 to N do
        Write(F,A[i,j]:3);
        Writeln(F);
      End;
    Close(F);
  End;
Procedure Try(k:Integer;x,y: Integer);
Var i,j,u,v : Integer;
Begin
  If k > nsq then Hien Else

```

```

    For i:=1 to 8 do
    Begin
        u:=x+D[i]; v:=y+C[i];
        If (u in [1..n]) and (v in [1..n]) and (A[u,v]=0) then
            Begin
                A[u,v]:=k;
                try(k+1,u,v);
                A[u,v]:=0;
            End;
        End;
    End;
End;
BEGIN
    Clrscr;
    Fillchar(A,Sizeof(A),0);
    dem:=0;
    DocFi;
    A[x,y]:=1;
    Try(2,x,y);
END.

```

Cách 2 : (Chuyển mảng 2 chiều sang 1 chiều , hiệu suất hơn)

```

Uses  Crt;
Const N      = 12;
Type  Mt     = Array[1..(n+4)*(n+4)] of Integer;
Var   x      : Mt;
      K      : Array[1..8] of Integer;
      db,spt,d,c,L,z : Integer; { db :so o dau bang }
Procedure Khoitao;
    Var i,j,all : Integer;
    Begin
        db := 2*(L+4)+2;
        all := (L+4)*(L+4);
        For i:=1 to all do X[i] := 1;
        For i:=1 to L do
            For j:=1 to L do
                X[db+(i-1)*(L+4)+j] := 0;
            X[db+(d-1)*(L+4)+c] := 1;
            K[1] := 2*L+9;      K[2] := 2*L+7;
            K[3] := L+6;       K[4] := L+2;
            K[5] := -K[4];     K[6] := -K[3];
            K[7] := -K[2];     K[8] := -K[1];
            z := 0; { So nghiem }
            spt:= L*L;
        End;
    End;
Procedure Hien;
    Var i,j : Integer;
    Begin

```

```

    Inc(z);
    Writeln('Nghiem : ',z);
    For i:=3 to L+2 do
    Begin
        For j:=3 to L+2 do
            Write(X[(i-1)*(L+4)+j]:3);
            Writeln;
        End;
    End;
    Procedure Tim(t,p : Integer);{ Di toi o thu t,ma dang o o thu p cua x }
    Var i : Integer;
    Begin
        If t=spt then Hien ;
        For i:=1 to 8 do
            If x[p-k[i]]=0 then
                Begin
                    x[p-k[i]] := t+1;
                    Tim(t+1,p-k[i]);
                    x[p-k[i]] := 0;
                End;
            End;
    End;
BEGIN
    Clrscr;
    Write('Kich thuoc ban co : ');
    Readln(L);
    Write('Nhap 2 toa do o xuat phat : ');
    Readln(d,c);
    Khoitao;
    Tim(1,db+(d-1)*(L+4)+c);
    If z=0 then Writeln('Khong co nghiem ');
END.

```

DẶNG 2 : Tìm một nghiệm :

```

Procedure Tim(k : Integer);
Begin
    Vòng lặp để cử mọi khả năng của b- ớc thứ k trong tìm kiếm 1 nghiệm
    Begin
        + Thử chọn 1 đề cử
        + Nếu đề cử này chấp nhận đ- ọc thì
            Begin
                * Ghi nhận giá trị đề cử
                * L- u trạng thái mới của bài toán sau đề cử
                * Nếu là b- ớc cuối cùng thì
                    Begin
                        Hiện Nghiệm
                        Thoát
                    End
                * Trả lại trạng thái tr- ớc khi đề cử
            End
        End;
    End;
End;

```

Hoặc có thể viết d- ới dạng sau :

```

Procedure Tim(k : Integer);
Begin
    Nếu là b- ớc sau b- ớc cuối cùng thì
        Begin
            Hiện Nghiệm
            Thoát
        End
    Còn không :
        Tạo vòng lặp để cử mọi khả năng của b- ớc thứ k trong tìm kiếm 1 nghiệm
        Begin
            + Thử chọn 1 đề cử
            + Nếu đề cử này thoả mãn bài toán thì
                Begin
                    * Ghi nhận giá trị đề cử
                    * L- u trạng thái mới của bài toán sau đề cử
                    * Nếu ch- a phải b- ớc cuối cùng thì Tim(K+1)
                    * Trả lại trạng thái của bài toán tr- ớc khi đề cử
                End
            End;
        End;
    End;
End;

```

Trong bài toán tìm 1 nghiệm , ng- ời ta th- ờng đ- a thêm vào các điều kiện đối với các khả năng đề cử để bỏ bớt đi 1 số khả năng đề cử hoặc làm cho khả năng đề cử thu hẹp lại

Thí dụ :

+ Điều kiện cần để một khả năng đ- ọc chấp nhận ở b- ớc thứ i là b- ớc i+1 cũng có khả năng chấp nhận một đề cử của nó và b- ớc thứ i ch- a phải b- ớc cuối cùng . Vì vậy có thể nhanh chóng tới đích nếu đ- a ra qui luật chọn đề cử của b- ớc thứ i nh- sau :

ở b- ớc thứ i ta sẽ chọn đề cử nào mà theo nó đ- a ta tới b- ớc i+1 có ít khả năng chấp nhận nhất (nghĩa là b- ớc thứ i+1 vẫn có khả năng đề cử của nó , nh- ng số đề cử ít)

+ Một cách khác : Khi chấp nhận một khả năng đề cử cho b- ớc thứ i , có thể sẽ tác động tới trạng thái bài toán . Vì vậy ta tính toán tr- ớc nếu chọn đề cử này thì trạng thái bài toán có thay đổi quá mức giới hạn cho phép hay không ? .Nghĩa là có v- ợt qua cận trên hoặc cận d- ới của bài toán hay không ? Nếu v- ợt qua thì ta không chọn đề cử ấy Trong nhiều bài toán những cận này cũng thu hẹp dần theo từng b- ớc , nếu ta tìm đ- ợc sự thay đổi của cận theo từng b- ớc thì các khả năng đề cử ngày càng hẹp dần , bài toán nhanh chóng kết thúc .

Trở lại bài toán con mã đi tuần nh- ng với yêu cầu chỉ hiện 1 nghiệm

Cách 1 : (Thông th- ờng)

Uses Crt;

Const Max = 7;

Fi = 'madq.inp';

D : Array [1..8] of -2..2 = (-2,-2,-1,1,2,2,1,-1);

C : Array [1..8] of -2..2 = (-1,1,2,2,1,-1,-2,-2);

Var

F : Text;

T1,T2 : longint;

A : Array[1..Max,1..Max] of Integer;

x,y,Lx,Ly,k,dem,n,nsq : Integer;

Procedure DocFi;

Begin

Assign(F,Fi);

{ \$I- } Reset(F); { \$I+ }

If Ioresult<>0 then

Begin

Writeln('Loi File ');

Readln;

Halt;

End;

Readln(F,N);

Nsq := N*N;

Readln(F,x,y);

Lx := x;

Ly := y;

```

        Close(F);
    End;
Procedure Hien;
    Var i,j : Integer;
    Begin
        Inc(dem);
        Assign(F,Fi);
        Append(F);
        Writeln(F,'Nghiem thu ',dem);
        For i:=1 to N do
            Begin
                For j:=1 to N do
                    Write(F,A[i,j]:3);
                    Writeln(F);
            End;
        Close(F);
    End;
Procedure Try(k:Integer;x,y: Integer);
    Var i,j,u,v : Integer;
    Begin
        If k>nsq then Hien Else
            Begin
                If dem=1 then
                    Begin
                        Writeln('Da xong . Moi an phim Enter ');
                        Readln;
                        Halt;
                    End;
                For i:=1 to 8 do
                    Begin
                        u:=x+D[i];
                        v:=y+C[i];
                        { Writeln(u,' ',v); }
                        If (u in [1..n]) and (v in [1..n]) and (A[u,v]=0) then
                            Begin
                                A[u,v]:=k;
                                try(k+1,u,v);
                                A[u,v]:=0;
                            End;
                    End;
                End;
                If (u=Lx) and (v=Ly) then
                    Begin
                        Writeln('Vo nghiem ');
                        Readln;
                        Halt;
                    End
            End
    End

```

```

    End;
  End;
BEGIN
  Clrscr;
  Fillchar(A,Sizeof(A),0);
  dem:=0;
  DocFi;
  A[x,y]:=1;
  k:=1;
  Try(2,x,y);
END.

```

Cách 2 :{ Đặt mắt chon h- ớng đi nhanh chóng tới đích là chon ô có bậc thấp nhất }
{Hiệu suất ch- ơng trình tăng đáng kể - Lời giải : Tr- ơng Vũ H- ng 12CT 1996}

```

{$A+,B-,D+,E+,F-,G-,I+,L+,N-,O-,P-,Q-,R+,S+,T-,V+,X+}
{$M 16384,0,655360}
Uses crt;
Const
  Max    = 20;
  dx     : Array[1..8] of integer=(-2,-1,1,2, 2, 1,-1,-2);
  dy     : Array[1..8] of integer=( 1, 2, 2,1,-1,-2,-2,-1);
Var
  N,x,y  : Byte;
  A      : Array[-1..max+2,-1..max+2] of Integer;
Procedure Nhap;
Begin
  Write('Nhap kích thước bàn cờ = ');
  Readln(n);
  Write('Nhap tọa độ xuất phát x,y = ');
  Readln(x,y);
End;
Procedure Hien;
Var
  i,j : Integer;
Begin
  For i:=1 to n do
    Begin
      For j:=1 to n do write(a[i,j]:4);
      Writeln;
    End;
  End;
Procedure Hangrao;
Var
  i,j : Integer;
Begin
  Fillchar(a,sizeof(a),0);
  For i:=-1 to n+2 do
    For j:=1 to 2 do

```



```

    Begin
        A[i,1-j]:=-1;
        A[i,n+j]:=-1;
        A[1-j,i]:=-1;
        A[n+j,i]:=-1;
    End;
End;
Function Bac(x,y:integer) : Integer;
    Var i,dem : Byte;
    Begin
        dem:=0;
        For i:=1 to 8 do
            If a[x+dx[i],y+dy[i]]=0 then inc(dem);
            Bac:=dem;
        End;
Procedure Vet(so,i,j:integer);
    Var k,lk ,Ldem,p : Byte;
    Begin
        If so>n*n then
            Begin
                Clrscr;
                Hien;
                Readln;
                Halt;
            End;
        Ldem:=9;
        For k:=1 to 8 do
            If A[i+dx[k],j+dy[k]]=0 then
                Begin
                    P := Bac(i+dx[k],j+dy[k]);
                    If { ( P>=0 ) and } ( Ldem>P ) then
                        Begin
                            Lk := k;
                            Ldem := p;
                        End;
                End;
        If Ldem = 9 then exit; { Ldem =9: ô (i,j) tắc nghẽn, nên Exit }
        { Ldem<9 : Sẽ chọn đề cử là ô có bậc nhỏ nhất }
        A[i+dx[Lk],j+dy[Lk]] := So;
        Vet(so+1,i+dx[Lk],j+dy[Lk]);
        A[i+dx[Lk],j+dy[Lk]] := 0;
    End;
Procedure Lam;
    Begin
        Hangrao;
        A[x,y]:=1;
        Vet(2,x,y);
    End;

```

```

End;
BEGIN
  Clrscr;
  Nhap;
  Lam;
END.

```

Lời bình : Ngoài việc sử dụng đệ quy kết hợp quay lui , ch-ơng trình còn dựa trên thuật toán “Hầu ăn” : có lợi thì làm để nhanh chóng đạt đích . Cụ thể là ở mỗi b-ớc SO sẽ chọn ô của b-ớc $(SO+1)$ tiếp theo nếu từ ô ấy có ít h-ớng đi tiếp tới ô khác của b-ớc $(SO+2)$. Cây phân nhánh sẽ ít nhánh đi đáng kể . Tất nhiên phải chứng minh rằng, với cách thức đi nh- thể vẫn bảo đảm có ít nhất 1 nghiệm.

Ta thấy :Bằng cách chọn ô có bậc thấp và phải xuất phát từ ô (1,1) nên cứ đi vòng quanh bàn cờ dần vào trong luôn có đ-ờng đi vào trong ruột bàn cờ , vì bậc các ô bên ngoài lớn hơn bậc các ô bên trong, và bậc các ô bên trong còn lớn hơn 1 khi mã ch-a vào sâu trong bàn cờ .Chỉ khi gần kết thúc mới nảy sinh vấn đề : có đ-ờng đi tiếp nữa hay không (còn ô có bậc lớn hơn 1 hay không) , nghĩa là khi đó ta mới biết cách đi này có đúng đắn không ? (Các em hãy tự chứng minh , hoặc ít nhất hãy thử nghiệm với các giá trị $N=5,6,7,8,..20$ nếu vẫn có nghiệm thì rõ ràng cách đi nh- thể đã đúng với các tr-ờng hợp này) và nh- thể kết quả thu đ-ợc cũng đã quá bất ngờ so với lập trình bình th-ờng .Vây ‘Hầu ăn’ nhiều khi cũng có lợi lắm đấy .

*

Một khó khăn khác của loại toán hiện 1 nghiệm là : tr-ờng hợp bài toán vô nghiệm cần viết ch-ơng trình nh- thể nào ? Phải duyệt hết mọi khả năng mới rõ kết luận vô nghiệm hay không vô nghiệm . Nghĩa là đã đi theo mọi nhánh nh-ng nhánh nào cũng đều không tới đích ,do đó theo quy luật cứ quay lui mãi để tìm kiếm thì đến lúc nào đó dẫn đến tình trạng phải trở về ô xuất phát .Vây khi gặp ô đề cử mới trùng với ô xuất phát thì bài toán vô nghiệm .(xem lại bài giải trang 330) .

Ta chỉ cần thêm vào mẫu 1 (Dạng tìm mọi nghiệm) một chút “gia vị” là có ngay dạng t-ơng ứng với bài toán vô nghiệm :

```

Procedure Tim(k : Integer);

```

```

Begin

```

```

  Vòng lặp đề cử mọi khả năng của b-ớc thứ k trong tìm kiếm 1 nghiệm

```

```

    Begin

```

```

      + Thử chọn 1 đề cử cho b-ớc k

```

```

      + Nếu đề cử này chấp nhận đ-ợc thì

```

```

        Begin

```

```

          * Ghi nhận giá trị đề cử;

```

```

          * L- u trạng thái mới của bài toán sau đề cử;

```

```

          * Nếu ch- a phải b-ớc cuối cùng thì Tim(K+1)

```

```

          Else {là b-ớc cuối cùng} thì Hiện Nghiệm;

```

```

          * Trả lại trạng thái của bài toán tr-ớc khi đề cử;

```

```

        End;

```

```

    End;

```

```

  Nếu đề cử cuối cùng ra khỏi vòng lặp trùng với giá trị của b-ớc thứ nhất thì

```

```

    Begin

```

```

      Thông báo vô nghiệm

```

Thoát

End;

End;

Cũng có thể viết dưới dạng sau :

Procedure Tim(k : Integer);

Begin

Nếu b-ớc k là b-ớc sau b-ớc cuối cùng thì Hiện nghiệm ;

Vòng lặp để thử mọi khả năng của b-ớc thứ k trong tìm kiếm 1 nghiệm

Begin

+ Thử chọn 1 đề cử cho b-ớc k

+ Nếu đề cử này thoả mãn bài toán thì

Begin

* Ghi nhận giá trị đề cử;

* L- u trạng thái mới của bài toán sau đề cử;

* Tim(k+1);

* Trả lại trạng thái của bài toán trước khi đề cử;

End;

End;

Nếu đề cử cuối cùng ra khỏi vòng lặp trùng với giá trị của b-ớc thứ nhất

thì

Begin

Thông báo vô nghiệm

Thoát

End;

End;

Hoặc có thể xử lý bài toán vô nghiệm như- chương trình sau :

Uses Crt;

Const N =5; nsq=n*n;

A : Array[1..8] of integer=(2,1,-1,-2,-2,-1,1,2);

B : Array[1..8] of integer=(1,2,2,1,-1,-2,-2,-1);

Type Index=1..n;

Var i,j : Index;

q : Boolean;

h : Array[index,index] of integer;

Procedure Try(i:integer;x,y:index;Var q:Boolean);

Var k,u,v : Integer;

q1 : Boolean;

Begin

k:=0;

Repeat

Inc(k);

q1:=false;

u :=x+a[k];

v :=y+b[k];

If (1<=u) and (u<=n) and (1<=v) and (v<=n) then

If h[u,v]=0 then

```

        Begin
            h[u,v]:=i;
            If i< nsq then
                Begin
                    Try(i+1,u,v,q1);
                    If not q1 then h[u,v]:=0;
                End
            Else q1:=true;
        End
    Until q1 or (k=8);
    q:=q1;
End;
BEGIN
    Clrscr;
    q:=False;
    For i:=1 to n do
        For j:=1 to n do h[i,j]:=0;
    h[1,1]:=1;
    Try(2,1,1,q);
    If q then
        For i:=1 to n do
            Begin
                For j:=1 to n do Write(h[i,j]:5);
                Writeln;
            End
        Else Writeln(' Không có nghiệm ');
    END.

```

Ng- ời lập trình đã đ- a thêm vào thủ tục đệ qui một tham biến q với chức năng làm nhiệm vụ thông báo tình trạng đã có nghiệm hay ch- a ? q chỉ nhận giá trị **TRUE** khi b- ớc tiếp theo là b- ớc cuối cùng . Do đó nếu sau khi đã vét cạn mọi khả năng vẫn không đi tới b- ớc cuối cùng , tham biến q sau khi thoát khỏi thủ tục đệ qui Try sẽ có giá trị **FALSE** ban đầu . Vậy sau thủ tục đệ qui Try , nếu $q=\text{TRUE}$ thì có nghiệm , nếu $q=\text{FALSE}$ là vô nghiệm .Nhiệm vụ của q nh- cái gây dò dẫm tìm đ- ờng vậy ! Có thể tăng độ dài của gây lên không, để nó thông báo kết thúc sớm hơn không ? (Các em hãy chạy ch- ơng trình với $N=4$).

ĐẶNG 3 : Tìm nghiệm tối - u

Có 3 cách th- ờng dùng :

Cách 1 :

```

Procedure Tim(k : Integer);
  Begin
    Nếu b- ớc k là b- ớc sau b- ớc cuối cùng thì
      Begin
        Nếu tìm đ- ợc nghiệm mới thì So sánh nghiệm mới với nghiệm
        l- u tối - u tr- ớc để chọn lại nghiệm l- u tối - u
      End;
      Vòng lặp đề cử mọi khả năng của b- ớc thứ k trong tìm kiếm 1 nghiệm
      ( Chú ý nên kết hợp với nghiệm l- u tối - u đã có để thu hẹp diện đề cử )
      Begin
        + Thử chọn 1 đề cử cho b- ớc k
        + Nếu đề cử này thoả mãn bài toán thì
          Begin
            * Ghi nhận giá trị đề cử;
            * L- u trạng thái mới của bài toán sau đề cử;
            * Tim(k+1);
            * Trả lại trạng thái của bài toán tr- ớc khi đề cử;
          End;
        End;
      End;
  End;

```

Thí dụ trong bài toán du lịch : Tìm đ- ờng đi qua N thành phố , mỗi thành phố chỉ qua 1 lần , sao cho tốn ít chi phí vận chuyển nhất . Mỗi nghiệm của bài toán là 1 véc tơ N thành phần đó là dãy tên có thứ tự chọn của N thành phố . Giả sử đã tìm đ- ợc 1 số nghiệm , và trong đó nghiệm tốt nhất có chí phí t- ơng ứng là CPM_{ax} đồng , bây giờ tìm tiếp các nghiệm còn lại .Đặt tình huống ta đang xây dựng tối thành phần thứ i (i<N) của nghiệm tiếp theo ,gọi CP₂ là tổng chi phí tối thiểu của N-i thành phố còn lại , CP₁ là tổng chi phí qua i thành phố đã chọn

Nếu một đề cử nào đó của b- ớc i mà $CP_1 + CP_2 > CPM_{\max}$ thì đề cử này bị loại .
Nh- vậy biết kết hợp với nghiệm tối - u của các nghiệm tr- ớc đó thì việc tìm kiếm nghiệm tiếp theo đ- ợc nhanh chóng hơn .

Cách 2 :

```

Procedure Tim(k : Integer);
  Begin
    Vòng lặp đề cử mọi khả năng của b- ớc thứ k trong tìm kiếm 1 nghiệm
    ( Chú ý nên kết hợp với nghiệm l- u tối - u đã có để thu hẹp diện đề cử )
    Begin
      + Thử chọn 1 đề cử cho b- ớc k
      + Nếu đề cử này chấp nhận đ- ợc thì
        Begin
          * Ghi nhận giá trị đề cử;
          * L- u trạng thái mới của bài toán sau đề cử;
          * Nếu ch- a phải b- ớc cuối cùng thì Tim(K+1)
          Else {là b- ớc cuối cùng} thì
            Begin
              So sánh nghiệm mới với nghiệm tối - u
              tr- ớc để chọn lại nghiệm tối - u
            End;
          * Trả lại trạng thái của bài toán tr- ớc khi đề cử
        End;
      End;
    End;
  End;

```

Cách 3 : Th- ờng dùng trong các bài toán chọn một số phần tử trong N phần tử cho tr- ớc để tạo thành 1 nghiệm .Thủ tục d- ới đây thực hiện thử chọn dần phần tử i cho nghiệm tốt nhất , S : điều kiện chấp nhận của các phần tử i sẽ chọn , F là cận trên của hàm mục tiêu cần tối - u (Xem lời giải bài toán cái túi - Trang 343)

```

Procedure Tim( i : Integer; S ,F: LongInt)
Begin
    * Nếu phần tử i thỏa mãn điều kiện chấp nhận S thì
    Begin
        + Ghi phần tử thứ i vào tập nghiệm
        + Nếu i ch- a phải phần tử cuối cùng then Tim(i+1,S_mới ,F)
        Còn không :
        Nếu cận trên còn lớn hơn so với L- u cận là LF thì
            Begin LF := F; L- uNghiệm := Nghiệm ; End;
        + Trả lại trạng thái cũ : Loại bỏ phần tử i khỏi tập nghiệm .
    End;
    * Giảm Cận trên của hàm mục tiêu : chọn cận mới là F_mới
    * Nếu F_Mới > LF thì
    Begin
        Nếu i ch- a là phần tử cuối cùng thì Tim(i+1,S,F_Mới)
        Còn không :
        Begin L- uF := F_Mới; L- uNghiệm := Nghiệm; End;
    End;
End;

```

Bài toán 1:

Bài toán ng-òì du lịch : Cho N thành phố , giá c-óc phí vận chuyển từ thành phố i tới thành phố j là C_{ij} . Yêu cầu :

File dữ liệu vào là 'DULICH.INP' như sau

Dòng đầu là N , XP , Dich (N số thành phố , XP : th/ phố xuất phát , Dich : th/phố đích)

N dòng tiếp theo :

Số đầu dòng là i , các cặp số tiếp theo là j và C_{ij} của ma trận $C(N,N)$

File dữ liệu ra là 'DULICH.OUT'

Dòng đầu : Liệt kê hành trình tốn ít chi phí nhất , lần l-ợt qua N thành phố (Mỗi thành phố chỉ 1 lần)

Dòng tiếp theo : Tổng chi phí .

TEST :**DULICH.INP**

```

10 1 8
1 2 3 5 2 7 3 9 3 10 7
2 5 1 6 6 10 3
3 1 7 8 1 10 7
4 1 3 2 2 5 3 9 7
5 1 2 3 7 4 5 6 1 7 8 8 2 9 3
6 1 8 2 7 3 5 7 6 8 1 10 8
7 1 1 3 3 5 2 6 5 8 6 10 1
8 2 2 3 7 6 4 9 2
9 2 5 6 1
10 2 1 4 6 5 2 7 3 8 6

```

DULICH.OUT

```

1 5 8
6

```

Bài chữa : Bài toán du lịch

```

Uses  Crt;
Const MN  = 100;
      TF1  = 'DULICH.INP';
      TF2  = 'DULICH.OUT';
Var    F    : Text;
      C     : Array[1..MN,1..MN] of Integer;
      KQ,LKQ : Array[1..MN] of Byte;
      D     : Array[1..MN] of Boolean;
      N,Lcs,cs,xp,Dich : Byte;
      Tong,LTong : LongInt;
Procedure Batdau;
Begin
  FillChar(C,Sizeof(C),0);
  FillChar(D,Sizeof(D),False);
  FillChar(KQ,Sizeof(KQ),0);
  FillChar(LKQ,Sizeof(LKQ),0);
End;

Procedure TaoF;
Var    F    : Text;
      i,j,k : Byte;
Begin
  Write('Nhap so thanh pho    : ');Readln(N);
  Write('Nhap thanh pho xuất phát : ');Readln(xp);
  Write('Nhap thanh pho sẽ tới   : ');Readln(Dich);
  Assign(F,TF1);
  Rewrite(F);
  Writeln(F,N,' ',xp,' ',Dich);
  Randomize;
  For i:=1 to N do
    Begin
      Write(F,i:4);
      For j:=1 to N do
        Begin
          k := Random(2);
          If i=j then k:=0;
          If k=1 then Write(F,j:4,(Random(8)+1):2);
        End;
      Writeln(F);
    End;
  Close(F);
End;

Procedure DocF;
Var    i,j    : Byte;

```



```

        F      : Text;
Begin
    Assign(F,TF1);
    Reset(F);
    Readln(F,N,XP,Dich);
    While Not SeekEof(F) do
        Begin
            Read(F,i);
            While Not Eoln(F) do
                Begin
                    Read(F,j);
                    Read(F,C[i,j]);
                End;
            End;
        Close(F);
        Tong := 0;
        LTong:= MaxInt div 2;
        cs   := 1;
        KQ[cs] := xp;
        D[xp] := True;
    End;

```

Procedure Hien;

```

    Var i,j : Byte;
    Begin
        For i:=1 to n do
            Begin
                For j:=1 to N do
                    If C[i,j]>0 then Write(C[i,j]:2)
                    Else Write('*':2);
                Writeln;
            End;
        End;
    End;

```

Procedure Tim (i: Byte;Tong : LongInt);

```

    Var      j : Byte;
    Begin
        For j:=1 to N do
            If (Not D[j]) and (i<>j) then
                If (C[i,j]>0) and (Ltong-Tong>=C[i,j]) then
                    Begin
                        Inc(cs);
                        KQ[cs] := j;
                        D[j]  := True;
                        Tong  := Tong + C[i,j];
                        If (j<>dich) then Tim(j,Tong)
                    Else
                        If (Tong<Ltong) or ((Tong=Ltong) and (cs<Lcs)) then

```

```

Begin
    Ltong := Tong;
    LKQ := KQ;
    Lcs := cs;
End;
Dec(cs);
D[j] := False;
Tong := Tong - C[i,j];
End;
End;
Procedure HienKQ;
Var i : Byte;
Begin
    For i:=1 to Lcs do
        Write(LKQ[i]:4);
        Writeln;
        Writeln('Tong chi phi la : ',LTong);
    End;
BEGIN
    Clrscr; { TaoF; }
    Batdau; DocF; Nhonhat := Min;
    If XP= Dich then
        Begin Writeln(Xp); Writeln('Khong di chuyen ');Readln;Halt;End;
    Tim(xp,Tong); { Hien;Chi gọi khi N<=10}
    Writeln;
    HienKq;
    Readln;
END.

```

Bài toán 2 (Bài toán cái túi) :



Tìm cách chọn các đồ vật trong N đồ vật (mỗi loại đồ vật chỉ chọn 1), xếp vào va li sao cho tổng giá trị của các đồ vật trong va li là lớn nhất nh-ng tổng trọng l- ợng của chúng không v- ợt quá giới hạn qui định là LimW. Giả sử N, Wi , Vi đều nguyên d- ợng (Wi : trọng l- ợng vật i , Vi : giá trị vật i)

Dữ liệu vào : cho trong File ‘VALY.INP’ tổ chức nh- sau

Dòng đầu : 2 số N LimW

N dòng tiếp theo : Mỗi dòng 2 số Wi Vi

Dữ liệu ra : File ‘VALY.OUT’

Dòng đầu : số LimW

Các dòng tiếp theo : Mỗi dòng 3 số : i Wi Vi là số thứ tự ,trọng l- ợng,giá trị của các đồ vật đ- ợc chọn vào va li.

Bài giải

```

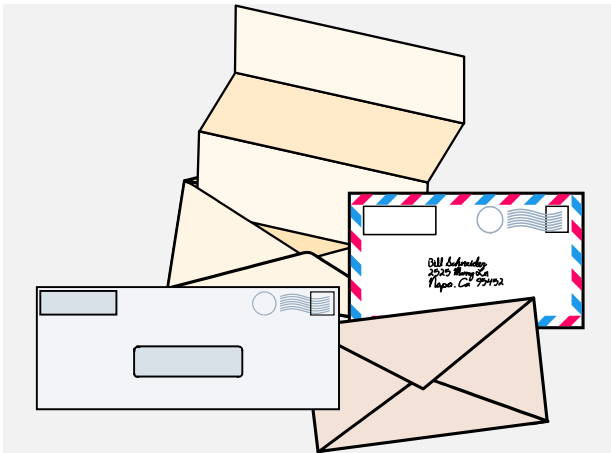
Uses  Crt;
Const  MN   = 30;
       TF   = 'Valy.inp';
       TF2  = 'Valy.out';
Type   Index = 1..MN;
       Dovat = Record
           W,V : Integer;  { W Trong luong ,V Gia tri }
       End;
Var    i,N   : Index;
       A     : Array[Index] of Dovat;
       KQ,LKQ : Set of Index;
       LimW,LCanV,CanV : Integer;
Procedure DocF;
    Var  i      : Index;
         F      : Text;
    Begin
        Assign(F,TF);
        Reset(F);
        Readln(F,N,LimW);
        For i:=1 to N do
            With A[i] do
                Begin
                    Readln(F,W,V);
                    CanV := CanV+V;
                End;
            Close(F);
        End;
End;
Procedure Try(i : Index;Tw,CanV : Integer);
    Var  CanV1 : Integer;
    Begin
        If Tw + A[i].w <= LimW then
            Begin
                KQ := KQ+[i];
                If i<N then Try(i+1,Tw+ A[i].w,Canv)
            Else
                If CanV > LCanV then
                    Begin
                        LCanV := Canv;
                        LKQ := KQ;
                    End;
                KQ := KQ-[i];
            End;
        CanV1:= CanV - A[i].v;
        If CanV1>LCanV then
            Begin
                If i<N then Try(i+1,Tw,CanV1)
            Else
                Begin

```

```

                LCanV := CanV1;
                LKQ := KQ;
            End;
        End;
    End;
Procedure GhiF;
    Var      i      : Index;
            F      : Text;
    Begin
        Assign(F,TF2);
        ReWrite(F);
        Writeln(F,'Gioi han trong luong : ',LimW);
        For i:=1 to N do
            If i in LKQ then
                With A[i] do
                    Writeln(F,i:4,' : TrLG = ',W:4,', GT = ',V:4);
                Close(F);
            End;
    BEGIN
        DocF;
        LCanV := 0;
        Try(1,0,CanV);
        GhiF;
        Writeln('Da xong ');
        Readln;
    END.

```



BÀI TẬP ĐỆ QUI CÔNG THUẬT TOÁN TÌM KIẾM BÔNG VÉT CƠN VÀ QUAY LUI BACKTRACKING

C11-B-01 Lập trình đặt 8 quân hậu lên bàn cờ sao cho không quân nào ăn đ-ợc quân nào (Bài toán t- ơng đ- ơng : 8 quân hậu không chế hết các ô của bàn cờ)

C11-B-02 Điền các số từ 1 đến $N \times N$ vào các ô của hình vuông $N \times N$ ($N \leq 5$) ô vuông theo qui cách : Nếu ô (x,y) có số k thì hoặc ô $(x+2,y-2)$ hoặc ô $(x+2,y+2)$ hoặc ô $(x-2,y+2)$ hoặc ô $(x-2,y-2)$ hoặc ô $(x+3,y)$ hoặc ô $(x-3,y)$ hoặc ô $(x,y+3)$ hoặc ô $(x,y-3)$ chứa số $K+1$. Nhập từ bàn phím số N và toạ độ x,y của ô xuất phát Hiện các cách sắp xếp theo dạng ma trận vuông trên màn hình , và tổng số cách sắp xếp .

C11-B-03 Trong hình vuông 4×4 ô vuông hãy sắp xếp 16 chữ cái : 4 chữ a, 4 chữ b, 4 chữ c , 4 chữ d sao cho mỗi dòng cũng nh- mỗi cột , mỗi chữ cái chỉ có mặt đúng 1 lần .

C11-B-04 (Tìm đ- ờng trong mê cung)

Mê cung gồm N phòng ($N < 100$) có các hành lang nối với nhau đó là nơi trú ngụ của quái vật Minotau (Nửa bò , nửa ng- ời) . Ban ngày quái vật th- ờng ra khỏi mê cung phun lửa giết chóc tàn phá với sức mạnh không ai địch nổi . Ban đêm quái vật ngủ trong mê cung và hòn than lửa của nó được cất ở phòng “Dich”; ai lấy được hòn than lửa ấy thì chinh phục đ- ợc quái vật. Theo lời thỉnh cầu của công chúa Arian , anh hùng Têđê nhận lời sẽ vào mê cung thu phục quái vật . Têđê xuất phát từ phòng XP và quyết định dùng thuật toán tìm kiếm bằng vét cạn và quay lui (cùng cuộn chỉ của nàng Arian tặng chàng để quay lui thuận tiện) . Trong mê cung tối om đầy đặc phòng và hành lang - chàng đã tìm được được phòng “Dich” và thu phục quái vật .

Em hãy lập trình hiện đ- ờng đi của Têđê .

Dữ liệu vào : File ‘MECUNG.TXT’ tổ chức như sau :

+ Dòng đầu là 3 số N XP Dich

+ N dòng tiếp theo :

Dòng thứ i : Đầu tiên là số i ($1 \leq i \leq N$) tiếp theo là các số j (hai số liên nhau cách nhau ít nhất 1 khoảng trống) thể hiện có hành lang một chiều từ phòng i sang phòng j .

Thông tin ra :

Đ- ờng đi của Têđê : liệt kê lần l- ợt các phòng chàng sẽ đi qua (không kể những đoạn phải quay lại)

C11-B-05 Trong biểu thức $(...(1?2)?3)?4)?5)...?N$, hãy thay các dấu ? bằng 1 trong 4 phép tính sau : + , - , * , / sao cho giá trị của biểu thức đã cho bằng S . Gọi số lượng các biểu thức tạo ra là d .

Yêu cầu :

Dữ liệu vào (gọi là dữ liệu Input) :

Nạp từ bàn phím số N và S nguyên dương thỏa mãn $1 < N < 255$; $-10^9 < S < 10^9$

Dữ liệu ra (gọi là dữ liệu Output) :

File 'BIEUTHUC.TXT'

+ Nếu $d=0$ thì dòng đầu ghi số 0

+ Nếu $d>0$ thì

Ghi d dòng , mỗi dòng là 1 biểu thức tìm được

Dòng cuối cùng là số d

Thí dụ :

Vào : N=5 S=1

Ra :

$((((1+2)-3)-4)+5)$

$((((1+2)*3)-4)/5)$

$((((1+2)/3)+4)/5)$

$((((1-2)+3)+4)-5)$

$((((1*2)-3)*4)+5)$

$((((1/2)*3)*4)-5)$

6

C11-B-06

Nhập phân số T/M ($0 < T < M < 969696$; T,M nguyên) . Lập trình thực hiện các yêu cầu :

a) Biểu diễn phân số dưới dạng phân số tối giản.

b) Biểu diễn phân số này dưới dạng tổng các phân số có tử số bằng 1 . Tổng càng ít số hạng càng tốt .

(Đề thi Olympic sinh viên Việt Nam - khối không chuyên 1996)

C11-B-07

Cho N quả cân có các khối lượng tương ứng là : d_1, d_2, \dots, d_N (nguyên) và có 1 cân 2 đĩa (khi cân có thể đặt một số quả cân trên đĩa nào cũng được)

a) Bộ quả cân đó có thể cân được những vật có khối lượng bao nhiêu ?

b) Cho vật có khối lượng M , cân nó bằng những quả cân nào ?

C11-B-08

Bài toán đổi tiền : Cho biết trong kho còn những loại tiền lẻ L_1, L_2, \dots, L_K với số lượng tương ứng là S_1, S_2, \dots, S_K tờ mỗi loại . Tìm cách đổi số tiền ST thành các loại tiền lẻ có trong kho . Giả thiết các số $L_1, L_2, \dots, L_K, S_1, S_2, \dots, S_K$ nguyên dương.

C11-B-09

Bài toán khôi phục hiện trạng cũ : Xét một ô đất hình chữ nhật $M \times N$ ô vuông . Mỗi ô đất có thể có 1 ngôi nhà đã xây hoặc chưa có ngôi nhà nào . Người ta mô tả miếng đất này bằng 1 bảng hình chữ nhật $M \times N$ ô vuông , mỗi ô chứa 1 số nguyên bằng tổng số nhà đã xây ở các ô xung quanh nó (các ô có chung đỉnh hoặc cạnh) . Hãy nêu rõ bản đồ về tình

trạng các nhà đã xây ở khu đất đó : Ô nào có nhà thì ghi số 1 ô nào ch- a có nhà thì ghi số 0 .

Thí dụ :

Khu đất với số liệu mô tả ban đầu

Khu đất đ- ọc khôi phục lại số liệu

1	1	1	2	0	1	0	0
1	3	3	3	1	1	0	0
0	2	1	3	2	2	2	1
0	3	3	5	2	2	3	1
1	4	4	5	4	3	3	3
0	4	5	6	5	3	3	1
1	4	5	7	5	3	3	2
0	2	3	5	4	4	1	1

0	1	1	0	1	0	0	0
0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	0	1	1	0	1
0	0	1	1	0	0	1	0
1	0	1	1	0	0	0	1
0	0	1	1	1	1	0	0
0	0	1	1	1	0	1	0

C11-B-10

Bài toán du lịch qua đủ N thành phố (mỗi thành phố chỉ qua 1 lần , trừ thành phố xuất phát) rồi quay trở lại thành phố xuất phát

Coi nh- đ- ờng đi 2 chiều. Tìm đ- ờng đi tốn ít c- ớc phí nhất và càng ngắn càng tốt (c- ớc phí là - u tiên số một) .

File dữ liệu : 'Dulich2.inp'

Dòng đầu N , XP

Các dòng tiếp theo :

Số đầu của 1 dòng là i , các số tiếp theo : tạo thành từng nhóm 3 số j,Cij ,Hij (j>i) và có ý nghĩa : Từ i có thể đi tới j với c- ớc phí Cij và khoảng cách là Hij

File dữ liệu ra : 'Dulich2.out'

Một số dòng đầu : các mã số các thành phố nêu hành trình

Dòng tiếp : 2 số : Tổng chi phí , Tổng đ- ờng dài của hành trình .

C11-B-11

Bài toán phát hành tem :

Trong một n- ớc ng- ời ta phát hành N loại tem khác nhau về giá trị (chẳng hạn loại tem 1 đồng , 3 đồng , . . .) Ng- ời ta không cho phép dán trên mỗi vật phẩm quá M con tem (có thể dán tem cùng loại) . Giá c- ớc mỗi vật phẩm là một số nguyên đồng . Nhập M,N từ bàn phím . Xác định tất cả các bộ giá trị của các loại tem cần phát hành sao cho dãy giá c- ớc của các vật phẩm đ- ọc gửi là một dãy dài các số nguyên liên tiếp dài nhất 1,2,3...,s

Thí dụ :

Số loại tem : N = 4

Số tem nhiều nhất trên 1 vật phẩm : M = 5

thì dãy giá c- ớc gửi đ- ọc dài nhất là 1,2,3, . . . , S = 71 với bộ tem {1,4,12,21} hoặc bộ {1,5,12,28 }

C11-B-12

Bài toán điều hành ô tô buýt :

Ông A ở bến ô tô buýt ghi lại thời điểm các ô tô đến bến thành 1 dãy số . Biết có nhiều tuyến xe cùng đến bến này . Hai ô tô liên tiếp của cùng 1 tuyến luôn cách nhau một khoảng thời gian cố định và mỗi tuyến có ô tô chạy đều đặn trong khoảng cả giờ (tính theo đơn vị nguyên phút , từ 0 phút đến 59 phút). Tại cùng một thời điểm có thể có nhiều ô tô của các tuyến khác nhau tới bến , cũng có thể khoảng thời gian cố định của 2 xe ô tô liên tiếp trên 2 tuyến nào đó nh- nhau

Hãy tìm số tuyến xe ít nhất theo dãy số của ông A

Yêu cầu :

File dữ liệu vào gồm 1 dòng là dãy số của ông A

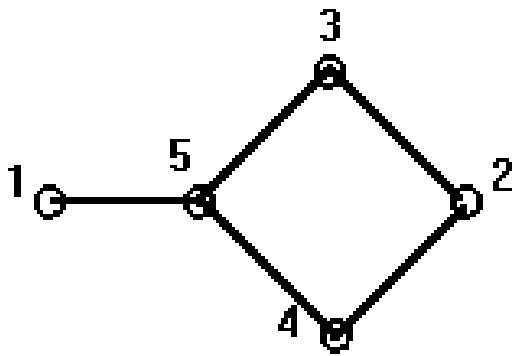
File dữ liệu ra đặt tên là 'OTO.OUT' mỗi dòng là 1 tuyến ô tô gồm 2 con số : thời điểm ô tô đầu tiên tuyến tới bến , sau đó là khoảng thời gian cố định của 2 xe ô tô liên tiếp của tuyến này .

C11-B-13

Bài toán tô màu

Trên mặt phẳng cho N điểm , một số điểm trong chúng đ-ợc nối với nhau bởi các đoạn thẳng. Hãy dùng số màu ít nhất để tô màu các điểm theo qui luật : 2 điểm có chung đoạn thẳng nối chúng với nhau thì đ-ợc tô bằng 2 màu khác nhau .

Thí dụ :



Điểm 2 và 5 sẽ tô màu số 1
 Điểm 1,3,4 sẽ tô màu số 2
 Vậy số màu cần dùng là : 2

C11-B-14

Bài toán giao thông

Tại một đầu mối giao thông ng-ời ta quản lý các tuyến đ-ờng qua nó . Ta coi 1 tuyến đ-ờng nh- 1 điểm trên mặt phẳng . Nếu 2 tuyến không đ-ợc đồng thời cùng thông đ-ờng (nghĩa là không cùng cho xe chạy một lúc) thì 2 điểm t-ơng ứng đ-ợc nối với nhau bằng 1 đoạn thẳng . Các điểm đ-ợc tô màu theo qui tắc : 2 tuyến không cùng thông đ-ờng đ-ợc tô bằng 2 màu khác nhau ,nghĩa là 2 điểm có chung đoạn thẳng nối chúng thì khác màu nhau . Hãy tô màu các điểm sao cho số màu dùng ít nhất . (Việc tô màu các

điểm , t-ong đ-ong với việc dựng cột đèn màu tại đầu mỗi giao thông này với số màu ít nhất , để số tuyến đ-ọc cùng thông đ-ờng càng nhiều càng ít tắc nghẽn giao thông)

Thí dụ :

Trong hình vẽ d-ối đây tuyến EC là đ-ờng 1 chiều ,còn lại các tuyến khác là đ-ờng 2 chiều

Tuyến số : 1 2 3 4 5 6 7 8 9 10 11 12 13
 Tên tuyến : AB AC AD BA BC BD DA DB DC EA EB EC ED
 Mạng tuyến đ-ờng này đ-ọc mô tả trong File GT.DAT nh- sau :

13

1 4 5 6 7 10

2 4 6 7 8 10 11

3 4 7 8 9 10 11 12

4 1 2 3 8 11

5 1 8 11

6 1 2 7 8 9 11 12

7 1 2 3 6 8 11 12 13

8 2 5 6 12 13

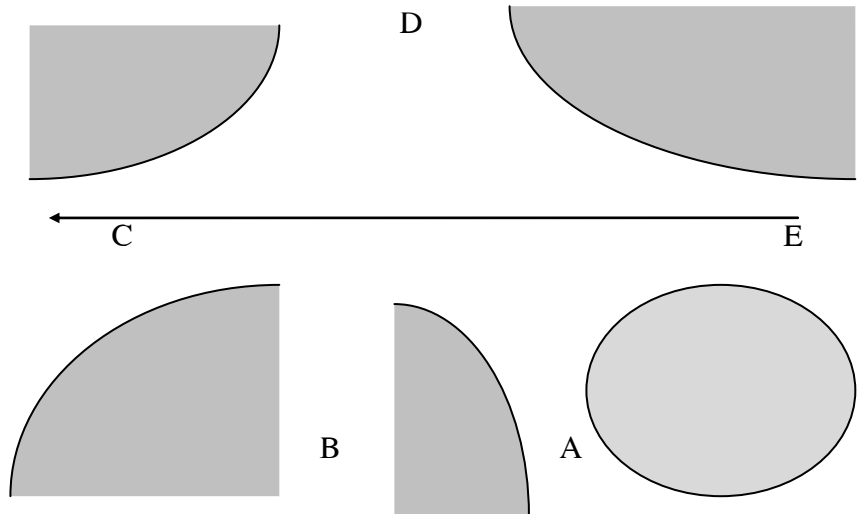
9 3 6 13

10 1 2 3

11 2 3 4 5 6 7

12 3 6 7 8

13 7 8 9



Dòng 1 là số tuyến : 13
 tuyến

Các dòng tiếp theo : số ở đầu dòng là tuyến không cùng thông đ-ờng với các tuyến số tiếp theo cùng dòng . Thí dụ dòng 6 : 5 1 8 11 có ý nghĩa tuyến 5 không cùng thông đ-ờng với các tuyến 1,8,11

Yêu cầu kết quả trên màn hình :

Dòng đầu : số màu ít nhất

Các dòng tiếp theo : mỗi dòng 1 tuyến gồm 2 con số : số của tuyến , màu của tuyến

Thí dụ với dữ liệu vào nh- trên , thì dữ liệu ra trên màn hình là :

4

1 1

2 1

3 1

4 2

5 2

6 2

7 3

8 3

9 3

10 2

11 4

12 4

13 2

C11-B-15 Bài toán ghép cặp

Có N thợ và N công việc. Mỗi thợ yêu thích từng công việc với mức độ khác nhau, mức yêu thích cho bằng điểm từ 1 đến N . Ng- ọc lại mỗi công việc sẽ đạt hiệu quả với các mức độ khác nhau, khi giao cho từng ng- ời thợ làm công việc ấy (mức hiệu quả cũng cho bằng điểm từ 1 đến N). Hãy phân công sao cho mỗi thợ 1 việc mà tổng hiệu quả công việc lớn nhất, đồng thời hạn chế 2 tình trạng éo le:

Tình trạng 1 : Công việc V1 sẽ giao cho thợ T1, nh- ng thợ T2 làm V1 hiệu quả hơn

Tình trạng 2 : Công việc V1 sẽ giao cho thợ T1, nh- ng thợ T1 thích V2 hơn.

C11-B-16

Cho M, N là 2 số tự nhiên ($M, N \leq 15$). Cho một bảng M dòng, N cột, chứa $M \cdot N$ số nguyên có giá trị từ 0 đến 99. Cho một số k . Tìm k phần tử trong bảng nói trên để tổng các phần tử đ- ọc lấy ra là lớn nhất với điều kiện trên mỗi hàng, mỗi cột chỉ đ- ọc chọn nhiều nhất 1 phần tử.

Dữ liệu vào : File 'TONGK.INP'

Dòng đầu 3 số M, N, K

M dòng tiếp theo : mỗi dòng là 1 dòng của bảng (gồm N số)

Dữ liệu ra : File 'TONGK.OUT'

Dòng đầu 2 số K, T (T là tổng các số đ- ọc chọn)

K dòng tiếp theo: Mỗi dòng 3 số : i, j, A_{ij} (i, j : chỉ số dòng, cột của số A_{ij} lấy ra từ bảng)

Thí dụ :

File 'TONGK.INP'

15 20 12

23	36	8	7	74	43	81	96	69	15	30	70	4	66	58	99	58	77	73	25
58	45	27	46	39	7	62	34	39	42	94	22	67	28	12	34	22	15	4	41
55	61	98	72	37	34	71	48	39	76	83	36	25	95	19	50	69	55	5	71
7	51	3	10	15	80	75	26	27	30	70	63	95	96	25	79	64	94	37	39
41	95	78	8	45	29	6	39	2	1	13	17	59	45	12	72	25	48	43	92
67	40	32	34	95	18	34	20	61	48	76	74	20	78	73	69	44	94	88	13
1	52	72	37	74	73	15	16	91	40	8	47	43	29	49	77	37	78	37	98
35	95	85	91	88	1	41	84	34	49	46	15	40	74	90	61	87	25	72	63
66	88	16	36	18	65	74	60	78	92	34	79	84	50	63	58	24	92	37	81
65	96	87	42	97	94	25	93	65	66	17	17	69	56	1	66	86	84	73	40
97	24	6	55	42	95	42	84	93	4	73	15	76	46	91	69	33	89	83	25
29	4	84	29	70	25	51	82	1	99	44	81	4	38	92	96	26	25	23	60
35	83	45	79	98	42	11	25	60	61	0	51	39	48	81	64	47	97	72	28
12	24	55	34	65	47	49	91	28	36	17	99	2	66	70	36	64	78	98	18
90	79	90	38	7	20	82	41	94	74	22	39	95	24	80	68	85	89	55	74

File 'TONGK.OUT'

12 1164

12 10 26

14	12	12
1	16	96
7	20	60
3	3	90
10	5	36
11	1	39
13	18	20
8	2	58
4	14	79
15	13	2
2	11	92

ÔỦ BÀI TRẦN C □ TH □ CHO D- □ I D □ NG SAU :

(Bài số 3 Đề thi Quốc gia chọn Học sinh giỏi Phổ thông năm học 1994-1995 Bảng A)

Kết quả thi đấu quốc gia của N vận động viên (đánh số từ 1 đến N) trên M môn (đánh số từ 1 đến M) đ- ọc đánh giá bằng điểm (giá trị nguyên không âm) . Với vận động viên , ta biết điểm đánh giá trên từng môn của vận động viên ấy . Các điểm này đ- ọc ghi trong File văn bản có cấu trúc :

- + Dòng đầu ghi số vận động viên và số môn .
- + Các dòng tiếp theo . mỗi dòng ghi các điểm đánh giá trên tất cả m môn của một vận động viên theo thứ tự môn thi 1,2,...,m . các dòng này đ- ọc ghi theo thứ tự vận động viên 1,2,...,N
- + Các số ghi trên một dòng cách nhau một dấu cách .

Cần chọn ra k vận động viên và k môn để lập một đội tuyển thi đấu Olympic quốc tế , trong đó mỗi vận động viên chỉ đ- ọc thi đấu 1 môn ($1 \leq k \leq M, N$) , sao cho tổng số điểm của các vận động viên trên các môn đã chọn là lớn nhất .

Yêu cầu :

Đọc bảng điểm từ 1 File văn bản (Tên File vào Từ bàn phím) , sau đó cứ mỗi lần nhận một giá trị k nguyên d- ơng từ bàn phím , ch- ơng trình đ- a lên màn hình kết quả tuyển chọn d- ối danh k cặp (i,j) với nghĩa vận động viên i đ- ọc chọn thi đấu môn j và tổng số điểm t- ơng ứng với cách đã chọn . Ch- ơng trình kết thúc khi nhận đ- ọc giá trị k=0

Các giá trị giới hạn $1 \leq M, N \leq 20$

Điểm đánh giá từ 0 đến 100 .

Thí dụ :

File dữ liệu

```
3 3
1 5 0
5 7 4
3 6 3
```

Mỗi khi nạp giá trị k ta nhận đ- ọc :

Nạp k=1 , máy trả lời (2,2) Tổng điểm = 7

Nạp k=2 , máy trả lời (2,1) (3,2) Tổng điểm = 11

Nạp k=3 , máy trả lời (1,2) (2,1) (3,3) Tổng điểm = 13

Nạp k=0 , Kết thúc

C11-B-17 (Bộ lọc Sắp xếp theo ph- ơng tiên song song)

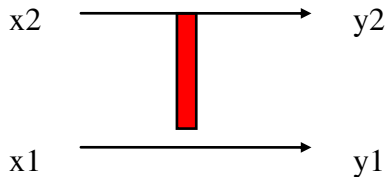
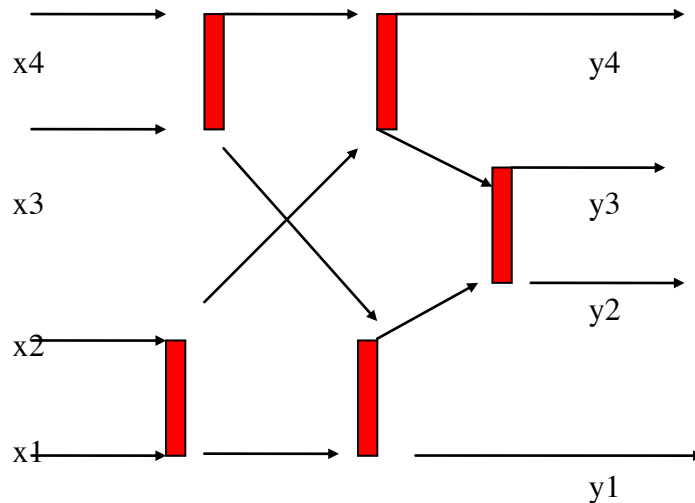
Một “Bộ lọc cỡ 2 “ để sắp xếp lại 2 phần tử là thiết bị với 2 đầu vào x_1, x_2 và hai đầu ra y_1, y_2 có dạng nh- hình vẽ 1 với mọi (x_1, x_2) qua bộ lọc cỡ 2 nhận đ- ọc $y_1 = \min(x_1, x_2)$ và $y_2 = \max(x_1, x_2)$. Với bộ lọc cỡ 2 bất kỳ đ- ờng ra chỉ số cao luôn là y_2 . Bộ lọc cỡ N ($N \leq 8$) là thiết bị đ- ọc xây dựng từ các bộ lọc cỡ 2 (coi nh- các bộ lọc cỡ 2 đã có) mà N tuyến thẳng từ lối vào tới lối ra , nó gồm N đầu vào là x_1, x_2, \dots, x_N và N đầu ra là y_1, y_2, \dots, y_N với $y_1 \leq y_2 \leq \dots \leq y_N$ là dãy sắp tăng của dãy x_1, x_2, \dots, x_N . Bộ lọc cỡ N đ- ọc đánh giá bởi 2 chỉ tiêu :

+ Số bộ lọc cỡ 2 là $S(N)$ càng ít càng tốt

+ Thời gian qua bộ lọc là $T(N)$ càng ít càng tốt (lấy thời gian qua 1 bộ lọc cỡ 2 làm đơn vị thời gian) , vậy cần bố trí có nhiều bộ lọc cỡ 2 đồng thời hoạt động càng tốt ,

Hãy lập trình chứng minh cách 1 thiết kế bộ lọc cỡ N (số cho tr- ớc) là đạt yêu cầu nêu trên .

Hình 1 : Bộ lọc cỡ 2

Hình 2 : Bộ lọc cỡ 4 ($S(4)=5, T(4)=3$)

Bảng tham khảo

N	2	3	4	5	6	7	8	≥ 9
$S(N)$	1	3	5	9	12	16	19	?
$T(N)$	1	3	3	5	5	6	6	?

Chú ý : Một bộ lọc cỡ N đ- ọc chấp nhận nếu mọi hoán vị của $1, 2, \dots, N$ qua bộ lọc đều đ- ọc lọc thành dãy tăng $1, 2, \dots, N$. Một bộ lọc cỡ N đ- ọc chấp nhận và đ- ọc gọi là tối - u nếu không thể giảm $S(N)$ và $T(N)$.

C11-B-18 (Xếp hình) Cho 3 hình với kích th- ớc nh- sau :

U		U
U		U
U	U	U

I
I

T	T	T
	T	
	T	

và một hình chữ nhật H có kích thước 6x9 ô vuông . Ta có thể một cách tùy ý các hình thuộc 3 loại trên lấp đầy hình H . Ví dụ sau đây là một cách xếp :

1- Nhập mảng A từ File văn bản có tên TT.TXT trong đó mỗi dòng của File ghi một dòng của mảng A d- ối dạng 1 xâu kí tự độ dài là 9 gồm các kí tự thuộc tập {U,I,T,C } {Không cần kiểm tra lại dữ liệu }

2- Khôi phục lại ít nhất 1 cách sắp xếp 3 loại hình nói trên lấp đầy hình H phù hợp với mảng A . Thông báo ra File văn bản có tên XEP.TXT theo qui cách viết mảng A

3- Nếu có thể , hãy tìm thêm càng nhiều càng tốt cách xếp 3 loại hình nói trên lấp đầy hình H phù hợp với mảng A .và ghi tiếp vào File XEP.TXT . Hai cách xếp liên tiếp cách nhau bởi 1 dòng trống .

U	U	U	T	I	U	U	U	T
U	T	T	T	I	U	T	T	T
U	U	U	T	I	U	U	U	T
U	U	U	T	I	U	U	U	T
U	T	T	T	I	U	T	T	T
U	U	U	T	I	U	U	U	T

Giả sử có một cách sắp xếp các hình thuộc 3 loại trên lấp đầy hình H nh- ng thông tin về cách sắp xếp đó không đầy đủ và đ- ọc cho bởi mảng A[1..6,1..9] of char , trong đó A[i,j] nhận 1 trong 4 giá trị U,I,T,C t- ơng ứng tùy theo ô đó thuộc hình chữ U , hình chữ T , hình chữ I hay bị mất thông tin .

Ví dụ

U	C	C	T	C	C	U	C	C
C	T	C	C	I	C	C	T	C
C	U	C	C	C	C	C	U	C
C	U	C	T	C	C	C	U	C
U	C	T	C	C	U	C	T	C
C	C	C	C	C	C	C	C	C

C11-B19 (Bài 3 - Đề thi chọn đội tuyển tin học quốc gia 1994)

Cho bàn cờ tổng quát $N \times N$ ô vuông , $N \leq 10$.Các ô màu trắng và màu đen đ-ợc phân bố một cách tùy ý , nh- ng phải thoả mãn hai điều kiện sau đây :

- i) Mỗi cột có ít nhất một ô màu trắng .
- ii) Có ít nhất một cột chỉ gồm các ô màu trắng

Cần xếp các con xe vào bàn cờ , sao cho :

- 1) Các con xe chỉ ở các ô màu trắng
- 2) Trên mỗi dòng và trên mỗi cột có không quá 1 con xe
- 3) Mỗi ô trắng không có xe nếu bị khống chế bởi một con xe khác trên cùng một cột

Yêu cầu : a) Đọc từ File kiểu TEXT (tên File đ-ợc cho từ bàn phím) , giá trị N và hình trạng của bàn cờ $N \times N$ gồm N xâu các kí tự 1 và 0 trong đó 1 biểu diễn ô màu trắng và 0 biểu diễn ô màu đen , mỗi xâu ứng với một hàng trên bàn cờ

b) Xếp lên bàn cờ càng nhiều con xe càng tốt , sao cho các điều kiện (1),(2),(3) nói trên thoả mãn .

c) Ghi ra File CHESS.SOL số l-ợng M các con xe đã xếp đ-ợc và hình trạng của bàn cờ sau khi xếp xe (ô có xe xếp đ-ợc đánh dấu bằng kí tự X)

Giả thiết dữ liệu vào là chuẩn xác nên không cần kiểm tra .

C11-B20 (Bài 2 - NETWORK OF SCHOOLS -Bài thi Quốc tế 1996 tại Hung Ga ri)

Một số tr-ờng học đ-ợc nối với nhau bằng một mạng máy tính . Có một sự thoả thuận giữa các tr-ờng học này : mỗi tr-ờng có một danh sách các tr-ờng học (gọi là danh sách các trường “nhận”) . và mỗi trường khi nhận được một phần mềm từ một trường khác trong mạng học từ bên ngoài , cần phải chuyển phần mềm nhận đ-ợc cho các tr-ờng trong danh sách các tr-ờng nhận của nó .Cần chú ý rằng nếu B thuộc danh sách các tr-ờng nhận của tr-ờng học A thì A nhất thiết phải xuất hiện trong danh sách các tr-ờng nhận của tr-ờng học B .

Ng-ời ta muốn gửi một phần mềm đến tất cả các tr-ờng học trong mạng . Bạn cần viết ch-ơng trình tính số ít nhất các tr-ờng học cần gửi bản sao của phần mềm này để cho phần mềm đó có thể chuyển đến tất cả các tr-ờng học trong mạng theo thoả thuận trên (Câu a) . Ta muốn chắc chắn rằng khi bản sao phần mềm đ-ợc gửi đến một tr-ờng học bất kỳ , phần mềm này sẽ đ-ợc chuyển tới tất cả các tr-ờng học trong mạng . Để đạt mục đích này , ta có thể mở rộng các danh sách các tr-ờng nhận , bằng cách thêm vào các tr-ờng mới . Tính số ít nhất các mở rộng cần thực hiện sao cho khi ta gửi một phần mềm mới đến một tr-ờng bất kỳ trong mạng , phần mềm này sẽ đ-ợc chuyển đến tất cả các tr-ờng khác (Câu b) . Ta hiểu một mở rộng là việc thêm một tr-ờng mới vào trong danh sách các tr-ờng nhận của một tr-ờng học nào đó .

Dữ liệu vào : Dòng đầu tiên của File INPUT.TXT chứa số nguyên N : số tr-ờng học trong mạng ($2 \leq N \leq 100$) . Các tr-ờng đ-ợc đánh số bởi N số nguyên d-ơng đầu tiên . Mỗi một trong N dòng tiếp theo mô tả một danh sách các tr-ờng nhận . Dòng thứ $i+1$ chứa số hiệu các tr-ờng nhận của tr-ờng i .

Mỗi danh sách kết thúc bởi số 0 . Dòng t-ơng ứng với danh sách rỗng chỉ chứa 1 số 0

Dữ liệu ra : Ch-ơng trình của bạn cần ghi hai dòng ra File OUTPUT.TXT . Dòng thứ nhất ghi một số nguyên d-ơng là lời giải của câu a) . Dòng thứ hai ghi lời giải của câu b .

Ví dụ :

INPUT.TXT

5
2 4 3 0
4 5 0
0
0
1 0

OUTPUT.TXT

1
2

PHẦN LỜI GIẢI

DÙNG ĐỆ QUI THỂ HIỆN THUẬT TOÁN VẾT CẠN (20 BÀI)

C11-B01

Uses crt;

Var i,dem : Integer;

A : Array[1..8] of Boolean;

B : Array[2..16] of Boolean;

C : Array[-7..7] of Boolean;

x : Array[1..8] of integer;

Procedure Print; { Hiện mọi nghiệm }

Var k:integer;

Begin

For k:=1 to 8 do Write(x[k]:4);

Writeln;

Inc(dem);

If dem mod 24 =0 then Readln;

End;

Procedure Try(i:integer);

{Đặt hậu vào dòng i }

Var j:integer;

Begin

For j:=1 to 8 do { Chọn cột }

If a[j] and b[i+j] and c[i-j] then

Begin

x[i]:=j;

a[j]:=False;

b[i+j]:=False;

c[i-j]:=False;

If i<8 then Try(i+1) Else

Print;

a[j]:=True;

b[i+j]:=true;

c[i-j]:=true;

End;

End;

BEGIN

dem:=0;

For i:=1 to 8 do a[i]:=True;

For i:=2 to 16 do b[i]:=True;

For i:=-7 to 7 do c[i]:=True;

Try(1);

Write(' Tong so nghiem la : ', dem);

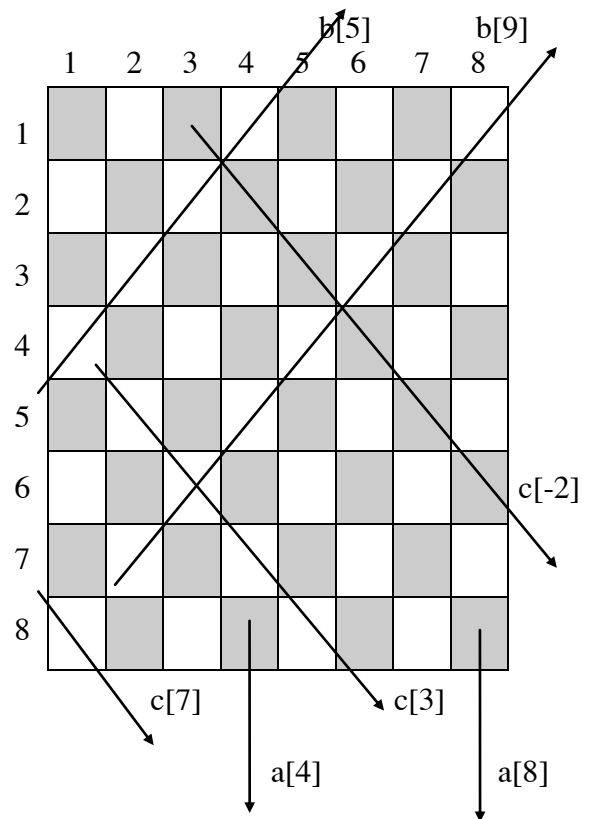
Readln;

END.

C11-B-02

Uses Crt;

Const N = 5;




```

SqrN  = N*N;
C      : Array[1..8] of Integer = (-3,3,0,0,2,-2,2,-2);
D      : Array[1..8] of Integer = (0,0,3,-3,2,-2,-2,2);
Type K  = Array[1..N,1..N] of Byte;
Var A   : K;
Sn      : Integer;
x,y     : Byte;
Procedure Khoitri;
Begin
    Writeln('Nhap toa do o xuất phát : ');
    Write('Dong y = '); Readln(y);
    Write('Cot x = '); Readln(x);
    FillChar(A,Sizeof(A),0);
    Sn := 0;
    A[x,y] := 1;
End;
Procedure Hien;
Var i,j : Byte;
Begin
    Inc(sn);
    Writeln('Nghiem thu ',sn,' : ');
    For i:=1 to N do
        Begin
            For j:=1 to N do Write(A[i,j]:3);
            Writeln;
        End;
    End;
End;
Procedure Vet(y,x : Byte);
Var k : Byte;
Function Chapnhan(x,y,k : Byte) : Boolean;
Begin
    If (x+C[k]>0) and (x+C[k]<N+1) and
       (y+D[k]>0) and (y+D[k]<N+1) and (A[y+D[k],x+C[k]]=0) then
        Chapnhan := True Else Chapnhan := False;
End;
Begin
    For k:=1 to 8 do
        Begin
            If chapnhan(x,y,k) then
                Begin
                    A[y+D[k],x+C[k]] := A[y,x] + 1;
                    If A[y+D[k],x+C[k]] < SqrN then
                        Vet(y+D[k],x+C[k]) Else Hien;
                    A[y+D[k],x+C[k]] := 0;
                End;
            End;
        End;
    End;
End;
BEGIN

```

```

    Clrscr;
    Khoitri;
    Vet(x,y);
    If sn=0 then Writeln('Khong co nghiem ')
    Else Writeln('So nghiem : ',sn);
    Readln;
END.

```

C11-B-03

```

Uses  Crt;
Const  N      = 5;
        M      = N*N;
Var    A      : Array[1..M] of Char;
        H,C    : Array[1..M] of 1..N;
        TH,TC : Array[1..N] of set of char;
        i      : Byte;
        dem    : LongInt;
Procedure Khoitri;
    Var i : Byte;
    Begin
        For i:=1 to M do
            Begin
                H[i] := (i-1) div N + 1;
                C[i] := i mod N;
                If C[i]=0 then C[i]:=N;
            End;
        For i:=1 to N do
            Begin
                TH[i] := [];
                TC[i] := [];
            End;
        dem := 0;
    End;
Procedure Hien;
    Var i : Byte;
    Begin
        Inc(dem);
        { For i:=1 to M do
            Begin
                Write(A[i]:2);
                If i mod N =0 then Writeln;
            End;
        Writeln;    }
    End;
Procedure Tim(i : Byte);
    Var j : Byte; ch : Char;
    Begin
        For ch:='A' to Char(64+N) do

```

```

Begin
  If (Not (ch in TH[H[i]]))and(Not (ch in TC[C[i]])) then
    Begin
      A[i] := ch;
      TH[H[i]] := TH[H[i]]+[ch];
      TC[C[i]] := TC[C[i]]+[ch];
      If i=M then Hien Else Tim(i+1);
      TH[H[i]] := TH[H[i]]-[ch];
      TC[C[i]] := TC[C[i]]-[ch];
    End;
  End;
End;
END.
N=4 So nghiem : 576  N=5 So nghiem : 161.280

```

C11-B-04

```

Uses  Crt;
Const Max  = 20;
      TF   = 'mecung.inp';
Var    A    : Array[1..Max*Max] of Byte;
      T     : Array[1..Max*Max] of Byte;
      D     : Array[1..Max] of Boolean;
      KQ    : Array[1..Max] of Byte;
      cs    : Integer;
      F     : Text;
      N,XP,Dich : Byte;
Procedure DocF;
  Var    i : Byte;
  Begin
    Assign(F,TF);
    Reset(F);
    Readln(F,N,Xp,Dich);
    k := 0;
    T[k] := 0;
    While Not SeekEof(F) do
      Begin
        Read(F,i);
        While Not SeekEoln(F) do
          Begin
            Inc(k) ;
            Read(F,A[k]);
          End;

```

```

        Readln(F);
        T[i] := k;
    End;
    Close(F);
End;
Procedure Hienkq; {Hiện 1 nghiệm }
    Var i : Integer;
    Begin
        For i:=1 to cs do Write(kq[i]:4);
        Readln;
        Halt;
    End;
Procedure Tim(i : Byte);
    Var j : Integer;
    Begin
        For j:=T[i-1]+1 to T[i] do
            Begin
                If Not D[A[j]] then
                    Begin
                        Inc(cs);
                        Kq[cs] := A[j];
                        D[A[j]] := True;
                        If A[j] <> Dich then Tim(A[j])
                        Else Hienkq;
                        Dec(cs);
                        D[A[j]] := False;
                    End;
            End;
        End;
    End;
BEGIN
    Clrscr;
    FillChar(D,Sizeof(D),False);
    FillChar(Kq,Sizeof(KQ),0);
    DocF;
    Cs :=1;
    Kq[cs] := Xp;
    D[Xp] := True;
    Tim(Xp);
    Hienkq;
    Readln
END.

```

C11-B-05

```

Uses   Crt;
Const  Tf      = 'Thi10b2.txt';
Type   Mang    = Array[1..254] of Byte;
       Tro     = ^Mang;
Var    i,N     : Integer;

```

```

S,SS  : Real;
d      : LongInt;
A      : Tro;
F      : Text;
T      : LongInt Absolute $0000:$046C;
Lt     : LongInt;
Procedure Nhap;
Begin
  Write('Go N=1 la thoat . Nhap N = ');
  Repeat
    Gotoxy(28,1); Clreol;
    {$I-} Readln(N); {$I+}
  Until (IoResult=0) and (N>0) and (N<255);
  If N=1 then Halt;
  Write('Nhap so ket qua da cho S = ');
  Repeat
    Gotoxy(28,2); {$I-} Readln(S); {$I+}
  Until (IoResult=0) and (S>-1.E+9) and (S<1.E+9);
End;
Procedure Hien;
Var i,j : Integer;
Begin
  Inc(d);
  For i:=1 to N-2 do Write(F,'(');
  Write(F,1);
  For i:=1 to N-1 do
    Case A^[i] of
      1: Write(F,'+',i+1,');
      2: Write(F,'-',i+1,');
      3: Write(F,'*',i+1,');
      4: Write(F,'/',i+1,');
    End ;
    Case A^[N] of
      1: Write(F,'+',i+1);
      2: Write(F,'-',i+1);
      3: Write(F,'*',i+1);
      4: Write(F,'/',i+1);
    End ;
  Writeln(F);
End;
Procedure Dondep;
Begin
  Gotoxy(1,1);
  Writeln(F,d,' nghiem : ');
  Gotoxy(1,25);
  Close(F);
  Writeln('Da xong trong thoi gian : ',((T-Lt)/18.2):10:0);
End;

```

```

Procedure Dithuan(i : Integer; Var SS : Real);
  Var j : Integer;
  Begin
    If ((T-Lt)/18.2 > 30) then
      Begin Dondep; Halt; End;
    If (Abs(SS-S) < 1.0E-4) and (i=N) then Hien ;
    If (i=N) and (SS <> S) then Exit;
    If (SS > 1.7E+37) or (SS < -1.7E+37) then
      Begin Writeln('So qua Max '); Readln; Halt; End;
    If (i <= N-1) and (A^[i] = 0) then
      For j:=1 to 4 do
        Case j of
          1: Begin
              SS := SS+i+1; A^[i]:= 1;
              Dithuan(i+1,SS);
              SS := SS-(i+1); A^[i]:= 0;
            End;
          2: Begin
              SS := SS-(i+1); A^[i]:= 2;
              Dithuan(i+1,SS);
              SS := SS+(i+1); A^[i]:= 0;
            End;
          3: Begin
              SS := SS * (i+1); A^[i]:= 3;
              Dithuan(i+1,SS);
              SS := SS/(i+1); A^[i]:= 0;
            End;
          4: Begin
              SS := SS/(i+1); A^[i]:= 4;
              Dithuan(i+1,SS);
              SS := SS *(i+1); A^[i]:= 0;
            End;
        End;
      End;
    End;
  End;
BEGIN
  Repeat
    Clrscr;
    New(A);
    Nhap;
    Lt := T;
    d := 0;
    Clrscr;
    Gotoxy(1,2);
    FillChar(A^,Sizeof(A^),0);
    If N>1 then
      Begin
        Assign(F,Tf);
        ReWrite(F);

```

```

        SS := 1;
        Dithuan(1,SS);
    End;
    Dondep;
    Readln;
    Until False ;
END.

```

C11-B-06

{Phuong phap De qui }

```

Uses  Crt;
Const TF    = 'Phanso.out';
Type  Kkq    = Array[1..1000] of LongInt;
Var    F      : Text;
        Kq     : Kkq;
        i,T,M,dem : LongInt;

```

```

Procedure Nhap;
Begin
    Repeat
        Write('Nhap tu so T ,mau so M (0<T<M<=969696) ');
        {$I-} Readln(T,M); {$I+}
        Until (IoResult=0) and (T>0) and(M>T) and(M<=969696);
    End;

```

```

Function UCLN(a,b : LongInt) : LongInt; {a,b > 0}

```

```

    Var d : LongInt;
    Begin
        d := a mod b;
        Repeat
            a := b;
            b := d;
            d := a mod b;
        Until d=0;
        UCLN := b;
    End;

```

```

Procedure Hienkq;

```

```

    Var i : LongInt;
    Begin
        Assign(F,TF);
        Append (F);
        For i:=1 to dem do Write(F,KQ[i], ' ');
        Writeln(F);
        Writeln(F,'Tong gom ',dem,' so hang ');
        Close(F);
    End;

```

```

Procedure Toigian(Var T,M : LongInt);

```

```

Var u : LongInt;
Begin
  u := UCLN(T,M);
  If u=1 then Exit;
  T := T div u;
  M := M div u;
End;
Procedure Thu(i,T,M : LongInt);
Begin
  If T=1 then
    Begin
      Inc(dem);
      Kq[dem] := M;
      Hienkq;
      Halt;
    End
  Else { T>1 }
    If (T/M<1/i) then
      Begin
        Inc(dem);
        Kq[dem] := M;
        Dec(T);
        Toigian(T,M);
        Thu(i+1,T,M);
      End
    Else { T/M>=1/i }
      Begin
        Inc(dem);
        kq[dem] := i;
        T := T*i-M;
        M := M *i;
        Toigian(T,M);
        Thu(i+1,T,M);
      End;
  End;
End;
Procedure Cau1;
Begin
  Assign(F,TF);
  ReWrite(F);
  Toigian(T,M);
  Write(F,T, ' ',M);
  Close(F);
End;
Procedure Cau2;
Begin
  Dem := 0;
  Toigian(T,M);
  Thu(2,T,M);

```



```

    End;
BEGIN
    Clrscr;
    Nhap;
    {Cau1;}
    Cau2;
    Writeln('Da xong ');
    Readln
END.

```

Lời bình :Ch-ơng trình trên dùng đệ qui kết hợp “háu ăn” nên kết quả phân tích phân số ch-a ngắn nhất . Nội dung của thuật toán nh- sau :

Mỗi lần cho số nguyên d-ơng i tăng dần , phân số T/M sau khi tối giản có 2 dạng :

+ a) Lớn hơn $1/i$

+ b) Không lớn hơn $1/i$

Nếu dạng a) thì phân tích $T/M = 1/i + (T/M - 1/i)$

Nếu dạng b) thì phân tích $T/M = 1/M + (T-1)/M$

Ch-ơng trình sau kết hợp 2 ch-ơng trình đệ qui và không đệ qui để chọn nhiệm tốt hơn (song vẫn ch-a hẳn là tối -u) vì trong bài toán này các khả năng phân tích một phân số quá nhiều , nên cũng đành chấp nhận sự ch-a tối -u hoàn toàn này vậy thôi ! . Hy vọng chờ đợi bài giải thành công của các em trong thời gian tới .

```

Uses  Crt;
Const TF      = 'Phanso.out';
Type  Kkq      = Array[1..10000] of LongInt;
Var    LT,LM,T,M,d1,d2 : LongInt;
        kq      : Kkq;
        F       : Text;
Procedure Nhap;
Begin
    Repeat
        Write('Nhap tu so T ,mau so M (0<T<M<=969696) ');
        {$I-} Readln(T,M); {$I+}
    Until (IoResult=0) and (T>0) and(M>T) and(M<=969696);
    LT := T;
    LM := M;
End;
Function UCLN(a,b : LongInt) : LongInt; {a,b > 0}
Var  d : LongInt;
Begin
    d := a mod b;
    Repeat
        a := b;
        b := d;
        d := a mod b;
    Until d=0;
    UCLN := b;
End;

```

```

End;
Procedure Hienkq;
  Var i : LongInt;
  Begin
    Writeln(F,'Cach 2 ');
    For i:=1 to d2 do
      Begin
        Write(F,Kq[i], ' ');
        If i mod 12 =0 then Writeln(F);
      End;
    Writeln(F);
    Writeln(F,d2);
  End;
Procedure Toigian(Var T,M : LongInt);
  Var u : LongInt;
  Begin
    U := UCLN(T,M);
    If U=1 then Exit;
    T := T div u;
    M := M div u;
  End;
Procedure Thu(i,T,M : LongInt);
  Begin
    If T=1 then
      Begin
        Inc(d2);
        Kq[d2] := M;
        Hienkq;
        If d1<d2 then Writeln(F,'Ket qua : Chon cach 1 ')
          Else Writeln(F,'Ket qua : Chon cach 2 ');
        Close(F);
        Halt;
      End
    Else { T>1 }
      If (T/M<1/i) then
        Begin
          Dec(T);
          Inc(d2);
          Kq[d2] := M;
          Toigian(T,M);
          Thu(i+1,T,M);
        End
      Else { T/M>=1/i }
        Begin
          Inc(d2);
          kq[d2] := i;
          T := T*i-M;
          M := M *i;

```

```

        Toigian(T,M);
        Thu(i+1,T,M);
    End;
End;
Procedure Cau2_Cach1;
Var i : LongInt;
Begin
    D1 := 0;
    Toigian(T,M);
    Writeln(F,'Cach 1 : ');
    i := M div T;
    While T>0 do
    Begin
        If (M mod i = 0 ) and (T*i>=M) then
        Begin
            T := T - M div i;
            Write(F,i,' ');
            Inc(d1);
            If d1 mod 12 =0 then Writeln(F);
            If T=0 then
            Begin
                Writeln(F);
                Writeln(F,d1);
                Exit;
            End;
        End
        Else Inc(i);
    End;
End;
Procedure Cau2_Cach2;
Begin
    d2 := 0;
    Toigian(T,M);
    Thu(2,T,M);
End;
BEGIN
    Clrscr;
    Nhap;
    d2 := 0;
    Assign(F,TF);
    ReWrite(F);
    Cau2_Cach1;
    T := Lt;
    M := Lm;
    Append(F);
    Cau2_Cach2;
    Readln
END.

```

Chương trình trên còn một hạn chế là trong File kết quả ghi cả 2 cách chọn, nếu chỉ nêu 1 cách chọn tối ưu hơn thì ban đầu ghi tạm cả 2 kết quả vào 1 File Nháp “PHANSO.BAK”. Sau đó tổ chức đọc File này và tìm kiếm chuyển kết quả tốt sang File chính thức “PHANSO.OUT”.

C11-B-07 (Bài toán cân trên 2 đĩa)

```

Uses Crt;
Const TF      = 'Can2dia.inp';
      TF2     = 'Can2dia.out';
      MN      = 20;
Var i,y,vc : Integer;
    KQ,QC   : Array[1..MN] of Integer;
    N,dem    : Integer;
    Ok       : Boolean;

    F        : Text;
    T        : LongInt Absolute $0000:$046C;
    LT,Maxvc : LongInt;
    X        : Array[0..MN] of Integer;
    D        : Array[1..1000] of Boolean;

Procedure Khoitri;
Begin
  Clrscr;
  FillChar(KQ,Sizeof(KQ),0);
  Maxvc := 0;
  X[0] := 0;
  Dem := 0;
End;

Procedure DocF;
Var i : Integer;
    F : Text;
Begin
  Assign(F,TF);
  Reset(F);
  Readln(F,N,VC);
  For i:=1 to N do
    Begin
      Read(F,QC[i]);
      Maxvc := Maxvc+QC[i];
    End;
  Close(F);
End;

Procedure Cau1;
Var stt : LongInt;
Procedure Inkq;
Var i : Integer;
    y : Longint;

```

```

Begin
    y := 0;
    For i:=1 to N do y := y+x[i]*qc[i];
    If (y>0) and (Not D[y]) then D[y] := True;
End;
Procedure Thu(i : Integer);
Var    j      : Integer;
Begin
    For j:= -1 to 1 do
        Begin
            x[i] := j;
            If i = N then Inkq Else Thu(i+1);
        End;
    End;
End;

Begin
    Lt := T;
    Assign(F,TF2);
    ReWrite(F);
    Writeln(F,'Can duoc cac vat sau : ');
    Thu(1);
    For i:=1 to Maxvc do
        If D[i] then
            Begin
                Write(F,i:4,' ');Inc(stt);
                If stt mod 10 = 0 then Writeln(F);
            End;
    Writeln(F);
    Writeln('Da xong cau 1 .Mat thoi gian : ',(T-Lt)/18.2:10:0);
End;
Procedure Cau2;
Procedure HienKQ;
Begin
    Inc(dem);
    Write(F,'Cach',dem:5,'**   Dia trai : ');
    For i:=1 to N do If KQ[i]=1 then Write(F,QC[i]:3);
    Write(F,' :9,' Dia Phai : ');
    For i:=1 to N do If KQ[i]=-1 then Write(F,QC[i]:3);
    Writeln(F);
End;
Procedure Chon(i : Integer);
Var    k,Ly    : Integer;
Begin
    For k:=-1 to 1 do
        Begin
            Ly    := y;
            y     := y+k*QC[i];
            KQ[i] := k;

```

```

        If y=vc then Hienkq
            Else If (i<N) then Chon(i+1);
        KQ[i] := 0;
        y     := Ly;
    End;
End;
Begin
    Lt := T;
    Dem := 0;
    If (vc>Maxvc) or (Not D[vc]) then
        Begin
            Writeln(F,'Khong the can duoc vat nang ',vc);
            Close(F);
            Exit;
        End;
    Writeln(F,'Cac cach can vat nang ',vc,' : ');
    Chon(1);
    Close(F);
    Writeln('Da xong cau 2 .Mat thoi gian : ',((T-Lt)/18.2):10:0);
End;
BEGIN
    Khoitri;
    DocF;
    Cau1;
    Cau2;
    Readln;
END.

```

C11-B-08 (Bài toán đổi tiền)

```

Uses  Crt;
Const Max  = 5000;
      TF   = 'DOITIEN.INP';
Type  Toanhang = Array[0..Max] of Integer;
      Kho      = Array[1..Max] of Integer;
Var    A       : Toanhang;
      Loai,slg  : Kho;
      Co        : Array[1..Max] of Boolean;
      Tien,Dem,Soloai,k : Integer;
Procedure Khoitri;
Begin
    FillChar(A,Sizeof(A),0);
    FillChar(Co,Sizeof(Co),False);
    A[0] := 1;
End;
Procedure DocF;
Var    F      : Text;
      i      : Integer;
Begin

```

```

        Assign(F,TF);
        Reset(F);
        Readln(F,Tien,soloai); Writeln(Tien,' ',Soloai);
        For i:=1 to soloai do
            Begin
                Readln(F,Loai[i],Slg[i]);
                Writeln(Loai[i]:4,' ',Slg[i]:4);
                Co[Loai[i]] := True;
            End;
        Close(F);
    End;
Function Vitri(T : Integer):Integer;{ Dong tien T la dong tien loai thu may }
    Var    i : Integer;
    Begin
        i := 1;
        While (i<=Soloai) and (T<>Loai[i]) do Inc(i);
        Vitri := i;
    End;
Function SoLuong(T,k : Integer): Integer;
    Var    phu ,i : Integer;
    Begin { Dong tien T co mat bao nhieu lan trong k so hang }
        Phu := 0;
        For i:=1 to k do
            If A[i] = T then Inc(phu);
        Soluong := Phu;
    End;
Procedure Phantich(T,k : Integer);
    Var    j,T1    : Integer;
           Ok      : Boolean;
    Procedure Hien;
        Var    j ,phu : Integer;
               TH     : Set of Byte;
        Begin
            If k>=1 then
                Begin
                    TH := [];
                    Inc(Dem);
                    Write('Cach ',dem,' : ');
                    phu := 0;
                    For j:=1 to k do
                        If Not(A[j] in TH) then
                            Begin
                                Inc(phu);
                                If phu=1 then
                                    Write(A[j],'*',SoLuong(A[j],k));
                                Else Write('+',A[j],'*',SoLuong(A[j],k));
                                TH := TH + [A[j]];
                            End;
                End;
            End;
        End;
    Write(A[j],'*',SoLuong(A[j],k))
    Else Write('+',A[j],'*',SoLuong(A[j],k));
    TH := TH + [A[j]];
End;

```

```

                                Writeln;
                                End;
                                End;
Begin
    If T=0 then Hien
    Else
        Begin
            T1 := A[k];
            For j:= T1 to T do
                If Co[j] then
                    If (Soluong(j,k)<Slg[Vitri(j)]) then
                        Begin
                            Inc(k);
                            A[k] := j;
                            T := T-j;
                            Phantich(T,k);
                            Dec(k);
                            T := T+j;
                        End;
                    End;
            End;
        End;
End;
BEGIN
    Clrscr;
    Khoitri;
    DocF;
    k :=0;
    Phantich(Tien,k);
    If Dem=0 then Writeln('Khong co cach phan tich ');
    Writeln('Da xong ');
    Readln;
END.

```

C11-B-08 (Cách 2 : Đơn giản và hiệu suất hơn . Lời giải TDH 2/1999)

```

uses crt;
const max = 5000;
      fi   = 'doitien.inp';
      fo   = 'doitien.out';
type k1   = array[1..max] of integer;
var g,s,kq : k1;
    n,m : integer;
    sn : longint;
    f : text;
procedure docf;
var f : text;
    i : integer;
begin

```



```

    for i:=1 to max do kq[i] := 0;
    assign(f,fi);
    reset(f);
    readln(f,n,m);
    writeln(n,' ',m);
    for i:=1 to m do
    begin
        readln(f,g[i],s[i]);
        writeln(g[i],' ',s[i]);
    end;
    close(f);
end;
procedure hien;
var i,dem : integer;
begin
    inc(sn);
    write(f,'nghiem ',sn,' : ',n,' = ');
    dem := 0;
    for i:=1 to m do
    if kq[i]>0 then
    begin
        inc(dem);
        if dem=1 then write(f,g[i],'*',kq[i])
        else write(f,'+',g[i],'*',kq[i])
    end;
    writeln(f);
end;
Procedure doi(T,i : integer); { Doi so tien con la T ra cac dong tien tu g[i] tro
len }
var j : integer;
begin
    for j:=0 to s[i]-kq[i] do
    begin
        inc(kq[i],j);
        T := T-g[i]*j;
        if T=0 then hien else
            if (T>0) and (i<m) then doi(T,i+1);
        dec(kq[i],j);
        T := T+g[i]*j;
    end;
end;

```

```

BEGIN
    clrscr;
    docf;
    assign(f,fo);
    rewrite(f);
    doi(n,1);
    writeln(f,sn);
    close(f);
    writeln('da xong ');
    readln;
END.

```

C11-B-09 (Bài toán khôi phục lại tình trạng cũ)

```

Uses    Crt;
Const   Max    = 100;
        Fi     = 'Khoiphuc.Inp';
        Fo     = 'Khoiphuc.Out';
        Fn     = 'Khoiphuc.Nhp';
        D      : Array [1..8] Of -1..1 = (-1,-1,-1,0,1,1,1,0);
        C      : Array [1..8] Of -1..1 = (-1,0,1,1,1,0,-1,-1);

Var
    A      : Array [0..Max,0..Max] Of Integer;
    B      : Array [0..Max,0..Max] Of Integer;
    M,N    : Integer;
    F      : Text;
    Ok     : Boolean;

Procedure Taofile;
    Var i,j,u,v,k, Dem : Integer;
    Begin
        Write('Nhap N,M : '); { Tạo File đáp số }
        Readln(N,M);
        Assign(F,Fn);
        Rewrite(F);
        Randomize;
        For i:=1 to N do
            Begin
                For j:=1 to M do
                    Begin
                        A[i,j]:=Random(2);
                        Write(F,A[i,j]:2);
                    End;
                Writeln(F);
            End;
        Close(F);
        Assign(F,Fi); { Từ File đáp số , tạo File dữ liệu vào là File KHOIPHUC.INP }
    End;

```

```

Rewrite(F);
Writeln(F,N,' ',M);
For i:=1 to N do
Begin
    For j:=1 to M do
    Begin
        Dem:=0;
        For k:=1 to 8 do
        Begin
            u:=i+D[k];
            v:=j+C[k];
            If (u>=1) and (v>=1) and (u<=N)and (v<=M)
                and (A[u,v]=1) then Inc(Dem);
        End;
        Write(F,Dem,' ');
    End;
    Writeln(F);
End;
Close(F);
FillChar(A,Sizeof(A),0); { Xoá mảng A }
End;
Procedure Docfile; { Lấy dữ liệu từ File KHOIPHUC.INP vào Mảng A }
Var i,j : Integer;
Begin
    Assign(F,Fi);
    Reset(F);
    Readln(F,N,M);
    For i:=1 to N do
    Begin
        For j:=1 to M do
        Begin
            Read(F,A[i,j]);
            Write(A[i,j]:3);
        End;
        Writeln;
        Readln(F);
    End;
    Close(F);
End;
Function Kt(i,x,y : Integer): Boolean; { Kiểm tra có giảm ô (x,y) i đơn vị đ-ọc không }
Var k : Integer;
Begin
    Kt:=True;
    For k:=1 to 8 do
        If ( A[x+D[k],y+C[k]] - i < 0 ) and (B[x+D[k],y+C[k]]<>-1) then
        Begin
            Kt:=False;
            Exit;
        End;
    End;
End;

```

```

        End;
    End;
Function Dem(x,y : Integer):Byte; {Đếm xem xung quanh ô (x,y) đã khôi phục đ- ọc bao
nhiêu}
    Var i,t :Integer;
    Begin
        t:=0;
        For i:=1 to 8 do
            If ( B[x+D[i],y+C[i]] =1 ) then Inc( t );
            Dem:=t;
        End;
Function Duoc: Boolean; {Kiểm tra bảng B tạo ra có chấp nhận đ- ọc không }
    Begin
        Duoc := A[N,M-1]-Dem(N,M-1))=(A[N-1,M]-Dem(N-1,M)) ;
    End;
Procedure Init;
    Var i,j : Integer;
    Begin
        For i:=0 to N+1 do
            For j:=0 to M+1 do    B[i,j]:=-1;
        For i:=0 to N+1 do A[i,0]:=0;
        For i:=0 to M+1 do A[0,i]:=0;
    End;
Procedure Inkq;      { Ghi kết quả vào File KHOIPHUC.OUT }
    Var    i,j      : Integer;
          F        : Text;
    Begin
        Ok:=True;      { Theo dõi bài toán có nghiệm }
        Assign(F,Fo);
        Rewrite(F);
        For i:=1 to N do
            Begin
                For j:=1 to M do Write(F,B[i,j]:2);
                Writeln(F);
            End;
        Close(F);
    End;
Procedure Vet(x,y : Integer);
    Var k,phu : Integer;
    Begin
        If (x=1) or (y=1) then
            Begin
                For k:=0 to 1 do
                    If Kt(k,x,y) then
                        Begin
                            B[x,y]:=k;
                            If y<M then Vet(x,y+1)
                        Else

```

```

                                If x<N then Vet(x+1,1)
                                    Else
                                        If Duoc then Inkq;
                                B[x,y]:=-1;
                                End;
                            End;
                        Else
                            Begin
                                B[x,y]:=A[x-1,y-1]-Dem(x-1,y-1);
                                If (B[x,y]=0) or (B[x,y]=1) then
                                    If y<M then Vet(x,y+1)
                                        Else
                                            If x<N then Vet(x+1,1)
                                                Else
                                                    If Duoc then Inkq;
                                    B[x,y]:=-1;
                                End;
                            End;
                        End;
                    BEGIN
                        Clrscr;
                        Ok:=False;
                        { Taofile; }
                        Docfile;
                        Init;
                        Vet(1,1);
                        If Not ok then Write('Vo nghiem ');
                        Readln;
                    END.

```

C11-B-10 (Bài toán du lịch 2)

```

Uses  Crt;
Const  MN    = 101;
        TF1   = 'DULICH2.INP';
        TF2   = 'DULICH2.OUT';
Var    F      : Text;
        C,H   : Array[1..MN,1..MN] of Integer;
        N     : Byte;
        KQ,LKQ : Array[1..MN] of Byte;
        D     : Array[1..MN] of Boolean;
        Lcs,cs,xp : Byte;
        Conghiem : Boolean;
        Tong,LTong,nhonhat,KC,LKC : LongInt;
Procedure Batdau;
Begin
    Conghiem := False;
    FillChar(C,Sizeof(C),0);
    FillChar(D,Sizeof(D),False);

```

```

    FillChar(KQ,Sizeof(KQ),0);
    FillChar(LKQ,Sizeof(LKQ),0);
End;
Procedure TaoF;
    Var      F      : Text;
            i,j,r,k,ph : Byte;
    Begin
        Write('Nhap so thanh pho      : ');Readln(N);
        Write('Nhap thanh pho xuat phat : ');Readln(xp);
        Assign(F,TF1);
        ReWrite(F);
        Writeln(F,N,' ',Xp);
        Randomize;
        For i:=1 to N do
            Begin
                Write(F,i:4);
                For j:=i+1 to N do
                    Begin
                        r := Random(2);
                        If r=1 then
                            Begin
                                k := Random(8)+1;
                                ph := Random(8)+1;
                                Write(F,j:4,k:2,ph:2);
                            End;
                        End;
                    End;
                Writeln(F);
            End;
        Close(F);
    End;
Procedure DocF;
    Var      i,j      : Byte;
            F      : Text;
    Begin
        Nhonhat := MaxInt div 2 ;
        Assign(F,TF1);
        Reset(F);
        Readln(F,N,XP);
        While Not SeekEof(F) do
            Begin
                Read(F,i);
                While Not Eoln(F) do
                    Begin
                        Read(F,j);
                        Read(F,C[i,j],H[i,j]);
                        C[j,i] := C[i,j];
                        H[j,i] := H[i,j];
                        If nhonhat>C[i,j] then nhonhat:= C[i,j];
                    End;
                Readln(F);
            End;
    End;

```

```

        End;
    End;
    Close(F);
    For i:=1 to N do
        Begin
            C[i,N+1] := C[i,xp];
            H[i,N+1] := H[i,xp];
            C[N+1,i] := C[i,xp];
            H[N+1,i] := H[i,xp];
        End;
    Tong := 0;
    LTong := MaxInt div 2;
    KC := 0;
    cs := 1;
    KQ[cs] := xp;
    D[xp] := True;
End;
Procedure Hien;
Var i,j : Byte;
Begin
    For i:=1 to N+1 do
        Begin
            For j:=1 to N+1 do
                If C[i,j]>0 then Write(C[i,j]:2)
                Else Write('*':2);
            Writeln;
        End;
    Writeln;
    For i:=1 to N+1 do
        Begin
            For j:=1 to N+1 do
                If C[i,j]>0 then Write(H[i,j]:2)
                Else Write('*':2);
            Writeln;
        End;
    End;
End;
Procedure Tim (i: Byte; Tong, KC : LongInt);
Var j : Byte;
Begin
    For j:=1 to N do
        If (Not D[j]) and (i<>j) then
            If (C[i,j]>0) and (LTong-Tong>=C[i,j]+(N-cs-1)*nhonhat) then
                Begin
                    Inc(cs);
                    KQ[cs] := j;
                    D[j] := True;
                    Tong := Tong + C[i,j];
                    KC := KC + H[i,j];
                End;
        End;
    End;
End;

```

```

    If (cs=N) then
    Begin
        If C[j,xp]>0 then
        Begin
            Tong := Tong + C[j,xp];
            KC := KC + H[j,xp];
            If (Tong<Ltong)
                or((Tong=Ltong) and (KC<LKC)) then
            Begin
                If Not conghiem then conghiem := True;
                Ltong := Tong;
                LKQ := KQ;
                LKC:= KC;
            End;
        End Else
        Begin
            Tong := Tong - C[j,xp];
            KC := KC - H[j,xp];
        End;
    End
    Else Tim(j,Tong,KC) ;
    Dec(cs);
    D[j] := False;
    Tong := Tong - C[i,j];
    KC := KC - H[i,j];
End;
End;
Procedure HienKQ;
Var i : Byte;
Begin
    For i:=1 to N do Write(LKQ[i]:4);
    Writeln(Xp:4);
    Writeln('Tong chi phi la : ',LTong);
    Writeln('Tong duong di : ',LKC);
End;
BEGIN
    Clrscr;
    { TaoF; }
    Batdau;
    DocF;
    Tim(xp,Tong,KC);
    Hien;{ Chi gọi khi N<=10}
    Writeln;
    If conghiem then HienKq Else Writeln('Vo nghiem ');
    Readln;
END.

```



```

Uses  Crt;
Const Max  = 10000;
Type  Giatri = Array[0..Max] of Integer;
      Tem    = Array[1..10] of Byte;
Var    M,N   : Integer;
      Lt,T   : Tem;
      GT     : Giatri;
      S,Ls   : Integer;
Procedure Nhap;
Begin
    Write('Nhap so loai tem la      N = ');Readln(N);
    Write('So tem dan toi da tren 1 vat pham M = ');Readln(m);
End;
Function MaxGt(x : Integer) :Integer; { Dãy giá c- ớc liên tục, do các tem từ 1 đến x sinh ra }
Var    i,h    : Integer;
      Procedure TimGt(i,j: Integer;Var h : Integer);{ Tìm các giá trị sau giá trị h , chúng
đ- ọc                                           sinh ra do có thể dán thêm
không quá j tem i }
      Var p : Byte; Lh : Integer;
      Begin
          For p:=0 to j do
          Begin
              Lh := h;
              Inc(h,T[i]*p);
              If (h < Max) and (GT[h]=0) then GT[h]:=1;
              If (i < x) then Timgt(i+1,j-p,h);
              h := Lh;
          End;
      End;
      Begin
          Fillchar(GT,Sizeof(GT),0);
          h:=0;
          Timgt(1,m,h);
          i:=h+1;
          While GT[i]<>0 Do Inc(i);
          MaxGt:=i-1;
      End;

Procedure Vet(k : Byte); { Bài toán xét tới tem thứ k }
Var    i,L : Integer;
      Begin
          L := MaxGt(k-1); { Day gia tri do cac tem 1->k-1 tao ra dai 1->L }
          For i:=T[k-1]+1 to L+1 do { i : du kien Gia tri cua tem moi }
          Begin
              T[k]:=i;
              If k<N then Vet(k+1)

```

```

Else
Begin
    S:=MaxGt(k);
    If S>Ls then
        Begin
            Ls := S;
            Lt := T;
        End;
    End;
End;
End;
End;
Procedure Lam;
Var    i : Byte;
Begin
    Ls:=0;
    T[1]:=1;
    Vet(2);
    Writeln('Day gia cuoc tu 1 --> ',Ls);
    Write('Bo tem can phat hanh la : ');
    For i:=1 to N do Write(Lt[i]:3);
End;
BEGIN
    Clrscr;
    Nhap;
    Lam;
END.

```

C11-B-12 (Bài toán ô tô buýt và các tuyến đ-ờng)

```

Uses    Crt;
Const   Max    = 60;
        Input   = 'Otobuyt.txt';
Type    Mang    = Array [0..59] of Byte;
Var      A,Batdau,Congsai : Mang;
        N,Sotuyen    : Byte;
Procedure Nhap;
Var      F        : Text;
        i,j        : Word;
Begin
    Fillchar(a,sizeof(a),0);
    Assign(F,input);
    Reset(F);
    Readln(F,N);
    For i:=1 to N do
        Begin Read(f,j);inc(a[j]);End;
    Close(F);
    Sotuyen:=31;
End;

```

```

Function KiemTra(xp,t : Byte) : Boolean;
Begin
    KiemTra:=false;
    Repeat
        If a[xp]=0 then exit;
        Inc(xp,t);
    Until (xp>59) ;
    KiemTra:=true;
End;
Function DauTien: byte;
Var
    i : byte;
Begin
    For i:=0 to 59 do
        If a[i]<>0 then
            Begin
                Dautien:=i;
                Exit;
            End;
    Dautien:= Max;
End;
Procedure Giam(xp,t : Byte);
Begin
    While xp<=59 do
        Begin
            Dec(a[xp]);
            Inc(xp,t);
        End;
End;
Procedure Tang(xp,t : Byte);
Begin
    While xp<=59 do
        Begin
            Inc(a[xp]);
            Inc(xp,t);
        End;
End;
Procedure Hien;
Var i : Byte;
Begin
    Writeln('So tuyen xe la : ',sotuyen);
    For i:=1 to sotuyen do writeln(Batdau[i], ' ',Congsai[i]);
End;
Procedure Vet( i : Byte);
Var j,k : Byte;
Begin
    k := Dautien;

```

```

        If k = Max then
        Begin
            Hien;
            Readln;
            Halt; {Đ- ọc nghiệm đầu tiên là thoát ngay, vì nghiệm này tốt nhất
rồi }
        End
    Else
    For j:=1 to 59-k do {Thuật ‘Hấu ăn’ : chọn công sai từ nhỏ đến lớn}
        Begin {tốt nhất vì phải lần l- ợt xét các tuyến theo thứ tự thời gian của
điểm xp}
            If kiểmtra(k,j) then
            Begin
                Giam(k,j);
                Batdau[i]      := k;
                Congsai[i]      := j;
                Sotuyen         := i;
                Vet(i+1);
                Tang(k,j);
            End;
        End;
    End;
BEGIN
    ClrScr;
    Nhap;
    Vet(1);
END.

```

Sau đây là một cách viết chuẩn mực , không ‘bay b- óm ‘ và ‘liều lĩnh ‘ nh- cách viết trên . Hãy test 2 lối viết này bằng các bộ Test hữu hiệu,mong các em sẽ có thêm một số kinh nghiệm nào đó khi lập trình ‘ thi đấu ! ‘

```

Uses  Crt;
Const Max  = 59;
      Fi    = 'oto.inp';
      Fo    = 'oto.out';
Type  Mang  = Array[0..max] of Byte;
      Ta    = Array[0..31] of Record Tg,Cs  :Byte; End;
Var   LT    : LongInt;
      T     : Longint Absolute $0:$046C;
      A     : Mang;
      Kq,Lkq: Ta;
      N,St,MinSt,dem,i : Byte;
Procedure Nhap;
    Var   i,j      : Byte;
          F        : Text;
    Begin
        Assign(F,Fi);{$i-} Reset(F) {$i+};

```

```

    If (Ioresult<>0) then
    Begin
        Write('Error file data ',Fi,' .Enter to quit');
        Readln;Halt;
    End;
    Readln(F,N);
    Fillchar(A,Sizeof(A),0);
    For i:=1 to N do
    Begin
        Read(F,j);
        Inc(A[j]);
    End;
    Close(F);
End;
Function Tim : Byte;
    Var i : Byte;
    Begin
        For i:=0 to Max do
            If A[i]>0 then
                Begin
                    Tim := i;
                    Exit;
                End;
        Tim := Max+1;
    End;
Function Kt(tg1,cs1,k1:Byte):Boolean;
Begin
    Kt := False;
    While tg1<=max do
        Begin
            If A[tg1]=0 then Exit;
            tg1 := tg1+ cs1;
        End;
        [With kq[k1] do
            If (Tg=tg1) and (Cs>cs1) then Exit;]
        KT:=True;
    End;
Procedure DoITT(tg,cs,chieu : Integer);
    Begin
        While tg<=max do
            Begin
                Dec(A[tg],chieu);
                tg := tg+cs;
            End;
        End;
Procedure Vet(k:Byte);
    Const tam = 45;
    Var cs1,tg1: Byte;

```

```

Procedure Toiuu;
  Begin
    Inc(dem);
    St := k-1;
    If St<MinSt then
      Begin
        MinSt := St;
        Lkq := Kq;
      End;
  End;
Procedure Ghitam;
  Var F : Text;
  Begin
    If dem>0 then
      Begin
        Assign(F,Fo);
        ReWrite(F);
        Writeln(' Tong So Tuyen tuong doi it nhat = ',MinSt);
        For i:=1 to MinSt do Writeln(F,Lkq[i].Tg,Lkq[i].Cs:3);
        Close(F);
        Readln;
        Halt;
      End
    Else
      Begin
        Writeln('Ch/tr không chạy được dữ liệu này trong ',tam,' giây ');
        Readln;
        Halt;
      End;
  End;
Begin
  If (T-Lt)/18.2>Tam then Ghitam
  Else
    Begin
      tg1 := Tim;
      If tg1 = Max+1 then Toiuu
      Else
        For cs1:=1 to Max-tg1 do
          If KT(tg1,cs1,k) then
            With kq[k] do
              Begin
                DoiTT(tg1,cs1,1);
                Tg := tg1;
                Cs := cs1;
                If k<St then Vet(k+1);
                DoiTT(tg1,cs1,-1);
              End;
        End;
    End;
End;

```

```

End;
Procedure Ghinghiem;
  Var F : Text;
  Begin
    Assign(F,Fo);
    ReWrite(F);
    If dem>0 then
      Begin
        Writeln(F,' Tong So Tuyen it nhat = ',MinSt);
        For i:=1 to MinSt do
          Writeln(F,Lkq[i].Tg:7,Lkq[i].Cs:3);
      End
    Else Writeln('Vo nghiem ');
    Close(F);
  End;
Procedure Khoitri;
  Begin
    LT := T; { Theo doi thoi gian bat dau chay chuong trinh }
    St := 31;MinSt := 31;
    Dem := 0;
    FillChar(Kq,Sizeof(kq),0);
    Lkq := kq;
  End;
BEGIN
  Clrscr;
  Nhap;
  Khoitri;
  Vet(1);
  Ghinghiem;
  Writeln('Da xong ');
  Readln;
END.
17
0 3 5 13 13 15 21 26 27 29 37 39 39 45 51 52 53

```

File ‘Otobuyt.inp’

```

17
0 3 5 13 13 15 21 26 27 29 37 39 39 45 51 52 53

```

File ‘Otobuyt.out’

```

0 13
3 12
5 8

```

C11-B-13 (Bài toán tô màu)

```

Uses      Crt;
Const     Max      = 14;
          Fi        = 'c:\tp97\soan\dequi\Tomau.txt';
Var       A        : Array[1..Max,1..Max] of 0..1;

```

```

Mau,LMau    : Array[1..Max] of Byte;
N,i,Minmau,MaxMau    : Integer;
Procedure NhapFile;
  Var i,j    : Integer;
      F      : Text;
  Begin
    FillChar(A,Sizeof(A),0);
    Assign(F,Fi);
    Reset(F);
    Readln(F,N);
    While not Eof(F) do
      Begin
        Read(F,i);Readln(F,j);
        A[i,j] := 1;
        A[j,i] := 1;
      End;
    End;
End;
Procedure Hien;
  Var i,j : Integer;
  Begin
    Writeln;
    For i:=1 to N do
      Begin
        For j:=1 to N do Write(A[i,j]:4);
        Writeln;
      End;
    End;
  End;
Function Kt(x,m : Integer): Boolean;{ Mau m gan cho dinh x }
  Begin
    Kt := False;
    For i:=1 to N do
      If (A[x,i]=1) and (m=Mau[i]) then Exit;
    Kt := True;
  End;
Procedure Tomau(x : Integer); { To mau cho dinh x }
  Var
    m,luu : Integer;
  Begin
    If x=N+1 then
      Begin { Đ- ọc 1 nghiệm , đổi lại cận trên   MaxMau }
        LMau    := Mau;
        MaxMau := MinMau;
        Exit
      End;
    m := 1;
    While m<Maxmau do
      Begin
        If KT(x,m) then

```



```

m;
Begin
    Mau[x] := m;
    Luu      := Minmau;
    If Minmau < m then Minmau :=
        Tomau(x+1);
        Minmau := Luu;
        Mau[x]  := 0;
    End;
    Inc(m);
End;

End;
Procedure Khoitri;
Begin
    FillChar(Mau, sizeof(Mau), 0);
    Maxmau    := N;
    Minmau    := 0;
    Mau[1] := 1;
End;
Procedure Thongbao;
Var i : Integer;
Begin
    For i:=1 to N do Writeln( ' Diem ',i:2,'      to mau : ',LMau[i]);
End;
BEGIN
    Clrscr;
    NhapFile;
    Hien;
    Khoitri;
    Tomau(2);
    Thongbao;
END.

```

C11-B-14 (Bài toán giao thông)

```

Uses  Crt;
Const Max  = 100;
      Fi   = 'Gthong.txt';
      Fo   = 'Gthong.out';
Type  M1   = Array[1..Max,1..Max] of Byte;
      M2   = Array[1..Max*Max+1] of Byte;
      M3   = Array[0..Max] of Byte;
Var    N    : Byte;
      A     : ^M1;
      B     : M2;
      T     : M3;
      MinM,MaxM : Integer;
      M,Lm    : M3;
Procedure DocF;

```

```

Var      F      : Text;
         i,j,so  : Byte;
Begin
  Assign(F,Fi);
  {$I-} Reset(F); {$I+}
  If Ioresult <> 0 then
    Begin
      Writeln('Loi File ');
      Readln;
      Halt;
    End;
  Readln(F,N);
  New(A);
  For i:=1 to N do
    For j:=1 to N do A^[i,j] := 0;
  While Not SeekEof(F) do
    Begin
      Read(F,i);
      While Not Seekeoln(F) do
        Begin
          Read(F,j);
          A^[i,j] := 1;
        End;
      Readln(F);
    End;
  Close(F);
End;
Procedure Chuyen_dl;
Var      i,j      : Byte;
         so       : Integer;
Begin
  T[0] := 0;
  so := 0;
  For i:=1 to N do
    Begin
      For j:=1 to N do
        If A^[i,j] = 1 then
          Begin
            Inc(so);
            B[so] := j;
          End;
        T[i] := so;
      End;
    End;
End;
Function KT(x,mau : Byte) : Boolean;
Var      p : Integer;
Begin
  Kt := False;

```

```

    For p:= T[x-1]+1 to T[x] do
        If M[B[p]]=mau then Exit;
    Kt := true;
End;
Procedure Inkq;
    Var      F      : Text;
            i      : Byte;
    Begin
        Assign(F,Fo);
        Rewrite(F);
        Writeln(F,'So mau can dung : ',MaxM);
        For i:=1 to N do Writeln(F,'Tuyen ',i,' to mau ',Lm[i]);
        Close(F);
    End;
Procedure GhiToiuu;
    Begin
        Lm := M;
        MaxM := MinM;
    End;
Procedure Tomau(i : Byte);
    Var j,Luu : Byte;
    Begin
        If i=N+1 then  Ghitoiuu
        Else
            Begin
                j := 1;
                While j<MaxM do
                    Begin
                        If Kt(i,j) then
                            Begin
                                M[i] := j;
                                Luu := MinM;
                                If MinM<j then MinM := j;
                                Tomau(i+1);
                                MinM := Luu;
                                M[i] := 0;
                            End;
                        Inc(j);
                    End;
            End;
    End;
End;
Procedure Khoitri;
    Begin
        MinM := 0;
        MaxM := N;
        FillChar(M,Sizeof(M),0);
    End;
BEGIN

```

```

Clrscr;
DocF;
Chuyen_dl;
Khoitri;
M[1] := 1;
Tomau(2);
Inkq;
END.

```

C11-B-15 (Bài toán ghép cặp)

```

Uses  Crt;
Const  N      = 8;
       Fi      = 'c:\tp97\soan\dequi\chonviec.inp';
       Fo      = 'chonviec.out';
Type   Mang    = Array[1..N] of 1..N;
       Qhe     = Array[1..N,1..N] of 1..N;
Var     Gheptho,GhepCV      : Mang;
       ChonTho,ChonCV,NgvTho,HqCV : Qhe;
       Thodxet : Array[1..N] of Boolean;
       F1,F2    : Text;
       Yeucau,Congviec,Tho,TongNv,Tonghiequua : Integer;
Procedure Hien;
  Var Congviec : Integer;
  Begin
    TongNv := 0;
    Tonghiequua := 0;
    For Congviec := 1 to N do
      Begin
        TongNv := TongNv+NgvTho[Gheptho[Congviec],Congviec];
        Tonghiequua := Tonghiequua+HqCV[Congviec,Gheptho[Congviec]];
      End;
    Writeln(F2,'Tong nguyen vong cua tho : ',TongNv);
    Writeln(F2,'Tong hieu qua Congviec      : ',Tonghiequua);
    Writeln(F2,'Phuong an hieu qua toi uu    (Congviec,Tho) : ');
    For Congviec:=1 to n do

Writeln(F2,'(',Congviec,',',Gheptho[Congviec],')= ',HqCV[Congviec,Gheptho[Congviec]]);
  End;
Procedure Ghep(Congviec : Byte); { xet tung Congviec }
  Var   yeucau      : Byte;
       Tho         : Byte;
  Function Benvung : Boolean;
  Var   CVx,Thox,i,Lim : Byte;
       Ok           : Boolean;
  Begin
    Ok := True;
    Lim := NgvTho[Tho,Congviec];
    i := 1;

```

```

While (i<Lim) and Ok do
Begin
    CVx := ChonCV[Tho,i];
    Inc(i);
    If CVx<Congviec then
        Ok := HqCV[Congviec,Tho]>HqCV[Congviec,Gheptho[CVx]]
    End;
    i := 1;
    While (i<Yeucau) and Ok do
        Begin
            ThoX := ChonTho[Congviec,i];
            Inc(i);
            If Thodxet[ThoX] then
                OK := NgvTho[ThoX,Congviec]>NgvTho[ThoX,GhepCV[ThoX]];
            End;
            Benvung := Ok;
        End;
    End;
    Begin
        For yeucau := 1 to N do
            Begin
                Tho := ChonTho[Congviec,yeucau];
                If Not Thodxet[Tho] then
                    If benvung then
                        Begin
                            Gheptho[Congviec] := Tho;
                            GhepCV[Tho] := Congviec;
                            Thodxet[Tho] := True;
                            If Congviec<N then Ghep(Congviec+1)
                            Else Hien;
                            Thodxet[Tho] := False;
                        End;
                    End;
                End;
            End;
        End;
    BEGIN
        Clrscr;
        Assign(F1,Fi);
        Reset(F1);
        Assign(F2,Fo);
        Rewrite(F2);
        While Not SeekEof(F1) do
            Begin
                For Congviec := 1 to n do
                    Begin
                        For yeucau := 1 to N do
                            Begin
                                Read(F1,ChonTho[Congviec,yeucau]);
                                HqCV[Congviec,ChonTho[Congviec,yeucau]] := yeucau;
                            End;
                        End;
                    End;
                End;
            End;
        End;
    End;

```

```

        Readln(f1);
    End;
    For Tho := 1 to N do
    Begin
        For yeucau:=1 to N do
        Begin
            Read(F1,ChonCV[Tho,yeucau]);
            NgvTho[Tho,ChonCV[Tho,yeucau]] := yeucau;
        End;
        Readln(f1);
    End;
End;
Close(F1);
FillChar(Thodxet,Sizeof(Thodxet),false);
Ghep(1);
Close(F2);
Writeln(#13#10'Da ghi xong vao file ',Fo);
Readln;
END.

```

C11-B-16

```

Uses  Crt;
Const Max  = 100;
      Fi    = 'Tongk.txt';
      Fo    = 'Tongk.out';
Type  M1    = Array[1..Max*Max+1] of Integer;
      M2    = Array[1..Max*Max+1] of Byte;
      M3    = Array[1..Max] of Byte;
      M4    = Array[1..Max] of Boolean;
Var   B,LB  : M1;
      D,C   : M2;
      M,N,k : Byte;
      Dx,DxC : M4;
      Tong,LTong,csMax: LongInt;
      KqD,KqC,LkqD,LkqC : M3;
Procedure DocF;
    Var   i,j   : Byte;
          F     : Text;
    Begin
        Assign(F,Fi);
        {$I-} Reset(F); {$I+}
        If IoResult<>0 then
            Begin
                Writeln('Loi File ');
                Readln;
                Halt;
            End;
        Readln(F,M,N,k);
    End;

```

```

    For i:=1 to M do
        Begin
            For j:=1 to N do
                Begin
                    Read(F,B[(i-1)*N+j]);
                    D[(i-1)*N+j] := i;
                    C[(i-1)*N+j] := j;
                End;
            Readln(F);
            Writeln;
        End;
    Close(F);
    LB := B;
    CsMax := M*N;
End;
Procedure Sapxep_dl; { Sap giam dan }
Procedure Quick(dau, cuoi : LongInt);
    Var    i,j,L,phu : LongInt;
    Begin
        i := dau;
        j := cuoi;
        L := (i+j) div 2;
        Repeat
            While B[i]>B[L] do Inc(i);
            While B[j]<B[L] do Dec(j);
            If i<=j then
                Begin
                    phu := B[i];
                    B[i] := B[j];
                    B[j] := phu;
                    phu := D[i];
                    D[i] := D[j];
                    D[j] := phu;
                    phu := C[i];
                    C[i] := C[j];
                    C[j] := phu;
                    Inc(i);
                    Dec(j);
                End;
            Until i>j;
            If dau<j then Quick(dau,j);
            If i<cuoi then Quick(i,cuoi);
        End;
    Begin
        Quick(1,M*N);
    End;
Procedure Khoitri;

```

```

Begin
  FillChar(B,Sizeof(B),0);
  FillChar(C,Sizeof(C),0);
  FillChar(DxD,Sizeof(DxD),False);
  FillChar(DxC,Sizeof(DxC),False);
  FillChar(KqD,Sizeof(KqD),0);
  FillChar(KqC,Sizeof(KqC),0);
  Tong := 0;
  Ltong := 0;
End;
Procedure GhiToiuu;
Begin
  LkqD := kqD;
  LkqC := kqC;
  Ltong:= Tong;
End;
Procedure Chon(i,j : Byte);{ xet toi o thu i trong Kq, tu o j trong B }
Var      d1,c1 : Byte;
         delta,j1,p,cL,Luu : LongInt;
Begin
  cL := k-i;
  Delta := Tong-LTong;
  If cL<0 then
    Begin
      If Delta>=0 then GhiToiuu;
    End
  Else
    Begin
      j1 := j-1;
      Repeat
        Inc(j1);
        d1 := D[j1];
        c1 := C[j1];
      Until (j1> Csmx) or ((Not DxD[d1])and (Not DxC[c1]));
      If j1<= csMax then
        If B[j1]+B[j1+1]*cL+Delta>0 then
          For p := j1 to csMax-1 do
            Begin
              d1 := D[p];
              c1 := C[p];
              If (B[p]+B[p+1]*cL+Delta>0) and
                (Not DxD[d1]) and (Not DxC[c1]) then
                Begin
                  DxD[d1] := True;
                  DxC[c1] := True;
                  Luu := Tong;
                  Tong := Tong+B[p];
                  KqD[i] := d1;

```



```

KqC[i] := c1;
Chon(i+1,p+1);
DxD[d1] := False;
DxC[c1] := False;
Tong := Luu;
KqD[i] := 0;
KqC[i] := 0;
End;
End;
End;
End;
End;
Procedure Inkq;
Var i : Byte;
F : Text;
Begin
Assign(F,Fo);
ReWrite(F);
Writeln(F,'k= ',k,' Tong = ',LTong);
For i:=1 to k do
Writeln(F,LkqD[i]:2,' ',LkqC[i]:2,' = ',LB[(LkqD[i]-1)*N+LkqC[i]]);
Close(F);
End;
BEGIN
Clrscr;
Khoitri;
DocF;
Sapxep_dl;
Chon(1,1);
Inkq;
END.

```

Sau đây là lời giải của Lê Sỹ Quang 12 Chuyên Tin 1995 (Bài đạt giải nhì toàn quốc 1995)

(Bài số 3 Đề thi Quốc gia chọn Học sinh giỏi Phổ thông năm học 1994-1995 Bảng A)

Kết quả thi đấu quốc gia của N vận động viên (đánh số từ 1 đến N) trên M môn (đánh số từ 1 đến M) đ- ợc đánh giá bằng điểm (giá trị nguyên không âm) . Với vận động viên , ta biết điểm đánh giá trên từng môn của vận động viên ấy . Các điểm này đ- ợc ghi trong File văn bản có cấu trúc :

+ Dòng đầu ghi số vận động viên và số môn .
+ Các dòng tiếp theo . mỗi dòng ghi các điểm đánh giá trên tất cả m môn của một vận động viên theo thứ tự môn thi 1,2,...,m . các dòng này đ- ợc ghi theo thứ tự vận động viên 1,2,...,N

+ Các số ghi trên một dòng cách nhau một dấu cách .

Cần chọn ra k vận động viên và k môn để lập một đội tuyển thi đấu Olympic quốc tế , trong đó mỗi vận động viên chỉ đ- ợc thi đấu 1 môn ($1 \leq k \leq M, N$) , sao cho tổng số điểm của các vận động viên trên các môn đã chọn là lớn nhất .

Yêu cầu :

Đọc bảng điểm từ 1 File văn bản (Tên File vào Từ bàn phím), sau đó cứ mỗi lần nhận một giá trị k nguyên dương từ bàn phím, chương trình đưa lên màn hình kết quả tuyển chọn đối danh k cặp (i,j) với nghĩa vận động viên i được chọn thi đấu môn j và tổng số điểm tương ứng với cách đã chọn. Chương trình kết thúc khi nhận được giá trị $k=0$

Các giá trị giới hạn $1 \leq M, N \leq 20$

Điểm đánh giá từ 0 đến 100.

Thí dụ :

File dữ liệu

```
3 3
1 5 0
5 7 4
3 6 3
```

Mỗi khi nạp giá trị k ta nhận được :

Nạp $k=1$, máy trả lời (2,2) Tổng điểm = 7

Nạp $k=2$, máy trả lời (2,1) (3,2) Tổng điểm = 11

Nạp $k=3$, máy trả lời (1,2) (2,1) (3,3) Tổng điểm = 13

Nạp $k=0$, Kết thúc

```
{ $A+,B-,D+,E+,F-,I+,L+,N-,O-,R-,S+,V- }
```

```
{ $M 16384,0,655360 }
```

```
Uses Crt;
```

```
Const Max = 20;
```

```
Type Ta = Array[1..max,1..max] of Integer;
```

```
Tb = Array[1..max] of Byte;
```

```
Tl = Array[1..max] of Integer;
```

```
Var N,M,k : Byte;
```

```
a : Ta;
```

```
b,lb : Tb;
```

```
G,Lg : Integer;
```

```
Ok : Set of Byte;
```

```
Procedure Input;
```

```
Var Tf : String;
```

```
f : Text;
```

```
Ok : Boolean;
```

```
i,j : Byte;
```

```
Begin
```

```
Repeat
```

```
Write(#10#13,'Cho biet ten file du lieu : ');
```

```
Readln(tf);
```

```
{ $i- } Assign(f,tf); Reset(f); { $i+ }
```

```
Ok:=Ioresult=0;
```

```
If Not Ok then
```

```
Begin Writeln('File loi hoac khong co file ten la : ',tf); End;
```

```
Until Ok or (tf="");
```

```
If tf="" then Halt;
```

```
Readln(f,n,m);
```

```
For i:=1 to n do
```

```

Begin
  For j:=1 to m do Read(f,a[i,j]);
  Readln(f);
End;
Close(f);
End;
Procedure NhapK;
Begin
  Repeat
    Write(#10#13,'Cho biet so mon can chon K:=');
    {$i-} Readln(k); {$i+}
  Until (Ioresult=0) and (k>=0) and (k<=m) and (k<=n);
End;
Procedure Hien;
Var i,j : Byte;
Begin
  For i:=1 to n do
    Begin
      For j:=1 to m do Write(a[i,j]:4); Writeln;
    End;
  End;
Procedure HienNghiem;
Var i : Byte;
Begin
  For i:=1 to n do
    If (Lb[i]>0) then Write('(','i,',',Lb[i],')');
    Writeln(#10#13,'Tong so diem = ',lg);
  End;
Procedure VETCAN(i,somon:Byte);
Var j : Byte;
Begin
  If (somon>k) then
    Begin
      If (lg<g) then
        Begin
          Lb:=b;
          Lg:=g;
        End;
      Exit;
    End;
  If (i>n) then Exit;
  For j:=1 to m do
    If Not (j in ok) then
      Begin
        g:=g+a[i,j];
        b[i]:=j;
        Ok:=Ok+[j];
        Vetcan(i+1,somon+1);
      End;

```

```

        g:=g-a[i,j];
        b[i]:=0;
        Ok:=Ok-[j];
    End;
    Vetcan(i+1,somon);
End;
Procedure Vet;
    Var i : Byte;
    Begin
        For i:=1 to m do B[i]:=0;
        Lg:=-maxint div 2;
        G:=0;
        Ok:=[ ];
        Vetcan(1,1);
        Hienngkiem;
    End;
BEGIN
    Clrscr;
    Repeat
        Input;
        Hien;
        Repeat
            NhapK;
            If (k>0) Then VET;
        Until (k=0);
        Write(#10#13,'ESC de thoat hoac phim bat ki de thu ');
        Write('lai voi file khac');
    Until (readkey=#27);
END.

```

C11-B-17

(Bài toán xây dựng bộ lọc)

```

Uses  Crt;
Const Max  = 10;
Type  Mang  = Array[1..Max] of Integer;
Var    i,n,dem,shv : Integer;
        M,M1 : Mang;
Procedure Hien;
    Var    k      : Byte;
    Begin
        Inc(shv);
        For k:=1 to n do Write(M[k]:3);
        Writeln;
    End;
Procedure Trao(Var a,b : Integer);
    Var c : Integer;
    Begin
        c := a;

```

```

        a := b;
        b := c;
    End;
Procedure L2(Var a,b : Integer);
    Var c : Integer;
    Begin
        If a > b then Trao(a,b);
    End;
Procedure L3(Var a,b,c : Integer);
    Begin
        L2(a,b);
        L2(b,c);
        L2(a,b);
    End;
Procedue L4(Var a,b,c,d : Integer);
    Var coc : Integer;
    Begin
        L2(a,b);
        L2(c,d);
        L2(a,c);
        L2(b,d);
        L2(b,c);
    End;
Procedure L5( var a,b,c,d,e : Integer);
    Var    coc : Integer;
    Begin
        L2(d,e);
        L2(b,c);
        L2(b,d);
        L2(c,e);
        L2(a,b);
        L2(b,e);
        L3(b,c,d);
    End;
Function OK(X,Y : Mang) : Boolean;
    Var    i : byte;
    Begin
        For i:=1 to N do
            If X[i]<>Y[i] then Begin OK := False; Exit; End;
        Ok := True;
    End;

Procedure Taohoanvi(n : Byte );
    Procedure Doicho (Var M : Mang; k : Integer);
        Var    i,j    : Byte;
               c      : Integer;
        Begin
            If k=1 then

```

```

                                Begin
                                    Writeln;
                                    Hien;
                                    L5(M[1],M[2],M[3],M[4],M[5]);
                                    Hien;
                                    If not Ok(M,M1) then Inc(dem);
                                End
                            Else
                                For i:= k downto 1 do
                                    Begin
                                        c := M[k];
                                        M[k] := M[i];
                                        M[i] := c;
                                        Doicho(M,k-1)
                                    End;
                                End;
                            End;
                        Doicho(M,n);
                    End;
                BEGIN
                    Clrscr;
                    dem := 0;
                    N:= 5;
                    For i:=1 to n do M[i] := i;
                    M1:=M;
                    Writeln;
                    Taohoanvi(n);
                    Writeln('So hoan vi cua ',n,' = ',shv div 2);
                    Writeln('So mac loi cua bo loc da xay dung la : ',dem );
                    If dem=0 then Writeln('OK ! ');
                    Readln;
                END.

```

C11-B-18 (Xếp hình U,I,T)

```

Program XapXep;
Uses Crt;
Const Input = 'xep_uit.txt';
Type  Mang1 = Array [1..6,1..9] of Char;
      Mang2 = Array [1..4,1..4] of Char;
Var   A,B   : Mang1;
      Dem   : Integer;
      Hinh  : Array [1..21] of Mang2;
      Cod,Coc : Array [1..21] of Byte;
Procedure Nhap;
Var   F   : Text;
      i,j : Byte;
Begin
    Assign(F,Input);

```

```

        Reset(F);
        For i:=1 to 6 do
        Begin
            For j:=1 to 9 do read(F,B[i,j]);
            Readln(F);
        End;
        Close(F);
        FillChar(A,Sizeof(A),' ');
    End;
Procedure Quay(k : Byte;Var h2: Mang2);
    Var i,j : Byte;
    Begin
        For i:=1 to Cod[k] do
            For j:=1 to Coc[k] do
                h2[j,Cod[k]+1-i] := hinh[k,i,j];
            Cod[k+1] := Coc[k];
            Coc[k+1] := Cod[k];
        End;
Procedure Taomau;
    Var i : Byte;
    Begin
        For i:=1 to 21 do
            FillChar(hinh[i],Sizeof(hinh[i]),' ');

            Hinh[1,1,1]:='U';Hinh[1,1,2]:=' ';Hinh[1,1,3]:='U';
            Hinh[1,2,1]:='U';Hinh[1,2,2]:=' ';Hinh[1,2,3]:='U';
            Hinh[1,3,1]:='U';Hinh[1,3,2]:='U';Hinh[1,3,3]:='U';
            Cod[1] := 3;
            Coc[1] := 3;
            Quay(1,Hinh[2]);
            Quay(2,Hinh[3]);
            Quay(3,Hinh[4]);

            Hinh[5,1,1]:='T';Hinh[5,1,2]:='T';Hinh[5,1,3]:='T';
            Hinh[5,2,1]:=' ';Hinh[5,2,2]:='T';Hinh[5,2,3]:=' ';
            Hinh[5,3,1]:=' ';Hinh[5,3,2]:='T';Hinh[5,3,3]:=' ';
            Cod[5] := 3;
            Coc[5] := 3;
            Quay(5,Hinh[6]);
            Quay(6,Hinh[7]);
            Quay(7,Hinh[8]);

            Hinh[8,1,1]:='T';Hinh[8,1,2]:='T';
            Hinh[9,1,1]:='T';Hinh[9,2,1]:='T';

            Cod[8] :=1; Coc[8]:=2;
            Cod[9] :=2; Coc[9]:=1;

```

```

Hinh[10,1,1]:='T';Hinh[10,1,2]:='T';Hinh[10,1,3]:='T';
    Hinh[10,2,1]:='U';Hinh[10,2,2]:='T';Hinh[10,2,3]:='U';
    Hinh[10,3,1]:='U';Hinh[10,3,2]:='T';Hinh[10,3,3]:='U';
Hinh[10,4,1]:='U';Hinh[10,4,2]:='U';Hinh[10,4,3]:='U';
Cod[10] := 4;
Coc[10] := 3;
Quay(10,Hinh[11]);
Quay(11,Hinh[12]);
Quay(12,Hinh[13]);

Hinh[14,1,1]:='T';Hinh[14,1,2]:='T';Hinh[14,1,3]:='T';
    Hinh[14,2,1]:='T';Hinh[14,2,2]:='T';Hinh[14,2,3]:='T';
    Hinh[14,3,1]:='T';Hinh[14,3,2]:='T';Hinh[14,3,3]:='T';
Cod[14] := 3;
Coc[14] := 3;
Quay(14,Hinh[15]);
Quay(15,Hinh[16]);
Quay(16,Hinh[17]);

Hinh[18,1,1]:='U';Hinh[18,1,2]:='T';Hinh[18,1,3]:='U';
    Hinh[18,2,1]:='U';Hinh[18,2,2]:='T';Hinh[18,2,3]:='U';
    Hinh[18,3,1]:='U';Hinh[18,3,2]:='U';Hinh[18,3,3]:='U';
Cod[18] := 3;
Coc[18] := 3;
Quay(18,Hinh[19]);
Quay(19,Hinh[20]);
Quay(20,Hinh[21]);
End;
Function Chapnhan(x,y,sh: Byte)      : Boolean;
Var d,c : Byte;
Begin
    If A[x,y]<>' ' then
        Begin
            Chapnhan := False;
            Exit;
        End;
    If Not ((x+Cod[sh]<8)
and (y+Coc[sh]<11)) then
        Begin
            Chapnhan := False;
            Exit;
        End;
    For d:=1 to Cod[sh] do
        For c:=1 to Coc[sh] do
            If Hinh[sh,d,c]<>' ' then
                Begin
                    If (A[d+x-1,c+y-1]<>' ') or ((B[d+x-1,c+y-1]<>'C') and
(B[d+x-1,c+y-1]<>Hinh[sh,d,c])) then

```



```

        Begin
            Chapnhan := False;
            Exit;
        End;
    End;
    Chapnhan := True
End;
Procedure Lap(x,y,sh : Byte);
    Var d,c : Byte;
    Begin
        For d:=1 to Cod[sh] do
            For c:=1 to Coc[sh] do
                Begin
                    If (Hinh[sh,d,c]<>' ') then
                        Begin
                            A[d+x-1,c+y-1] := Hinh[sh,d,c];
                        End;
                    End;
                End;
            End;
        End;
    End;
Procedure Thao (x,y,sh : Byte);
    Var d,c : Byte;
    Begin
        For d:=1 to Cod[sh] do
            For c:=1 to Coc[sh] do
                Begin
                    If (Hinh[sh,d,c]<>' ') then
                        Begin
                            A[d+x-1,c+y-1] := ' ';
                        End;
                    End;
                End;
            End;
        End;
    End;
Procedure HienKq;
    Var i,j : Byte;
    Begin
        Inc(dem);
        Writeln(dem);
        For i:=1 to 6 do
            Begin
                For j:=1 to 9 do Write(A[i,j]:2);
                Writeln;
            End;
        Writeln
    End;
Function Ketthuc : Boolean;
    Var i,j : Byte;
    Begin
        Ketthuc := False;
        For i:=1 to 6 do

```

```

        For j:=1 to 9 do
            If A[i,j]=' ' then Exit;
        Ketthuc := True
    End;
Procedure Tim(Var x,y : Byte);
Begin
    While A[x,y]<>' ' do
        Begin
            If y<9 then Begin Inc(y);End
            Else
                If x<6 then
                    Begin Inc(x);y := 1; End
                End
        End;
End;
Procedure Vet(x,y : Byte);
    Var Lx,Ly,i,j : Byte;
    Begin
        Begin
            Tim(X,Y);
            For i:=1 to 21 do
                Begin
                    If Chapnhan(x,y,i) then
                        Begin
                            Lap(x,y,i);
                            Lx :=1;Ly:=1;
                            If Ketthuc then HienKq Else Vet(Lx,Ly);
                            Thao(x,y,i);
                        End;
                    End;
                End
            End;
        End;
    End;
Begin
    ClrScr;
    Nhap;
    Taomau; dem := 0;
    Vet(1,1);
    Writeln('Da xong ',dem,' nhkiem ');
    Readln
End.

```

TEST

```

UUUUUUUUU
UUUUUUUUU
UUUUUUUUU
CUUUUUUUU
CCCCCCCCC

```

CCCCCCCCC

U	U	U	I	I	I	I	I	I
U	U	U	U	I	I	I	I	I
U	U	U	U	I	U	U	U	I
I	U	U	U	I	U	U	U	U
I	I	I	I	I	U	U	U	U
I	I	I	I	I	I	U	U	U

C11-B-19 (Bài 3 - Đề thi toàn quốc 1994)

{ Bài 3 - Đề thi toàn quốc 1994 }

Uses Crt;

Const Max = 16;

Fi = 'tq94_b3.txt';

Type M1 = Array[1..max,1..max] of Byte;

M2 = Array[1..max] of Boolean;

M3 = Array[1..max*max] of Record

x,y : Byte;

End;

Var A,B : M1;

Dxh,Dxc : M2;

N,d,Tong,LT : Byte;

Tr,KQ,LKQ : M3;

Procedure Input;

Var f : Text; S : String;

i,j : Byte;

Begin

Assign(f,fi); {\$i-} Reset(f); {\$i+}

If (ioresult<>0) then

Begin

Write('Error file data : ',fi, ' . Enter de thoat ');

Readln; Halt;

End;

Readln(f,n);

For i:=1 to n do

Begin

Readln(f,S);

For j:=1 to N do A[i,j] := Ord(S[j])-48;

End;

Close(f);

End;

Procedure Hien(A : M1);

Var i,j : Byte;

Begin

For i:=1 to n do

Begin

For j:=1 to n do

```

        Begin
            If A[i,j]=2 then Textcolor(10);
            Write(A[i,j]:2); Textcolor(15);
        End;
    Writeln;
End;
Function Kiemtra:Boolean;
Var i : Byte;
Begin
    Kiemtra:=False;
    If (Tong<=LT) then Exit;
    For i:=1 to d do
        If (B[Tr[i].x,Tr[i].y] = 1) and
            (Not Dxh[Tr[i].x] and Dxc[Tr[i].y]) then Exit;
    Kiemtra:=True;
End;
Procedure Vet(i,j:Byte);
Begin
    If (i = N+1) then
        Begin
            If Kiemtra then
                Begin
                    LT := Tong;
                    LKQ := KQ;
                End;
            Exit;
        End;
    If (A[i,j]=1) then
        Begin
            If Dxh[i] and Dxc[j] then
                Begin
                    Dxh[i]:=False;
                    Dxc[j]:=False;
                    Inc(Tong);
                    KQ[Tong].x:=i;
                    KQ[Tong].y:=j;
                    B[i,j] := 1;
                    If (j=N) Then Vet(i+1,1)
                    Else Vet(i,j+1);
                    Dxh[i]:=True;
                    Dxc[j]:=True;
                    B[i,j]:=0;
                    Dec(Tong);
                End;
            {If (j=N) Then Vet(i+1,1)
            Else Vet(i,j+1);
            Exit;}
        End;
    End;

```

```

    End;
    If (j=N) Then Vet(i+1,1)
      Else Vet(i,j+1);
  End;
Procedure Khoitao;
  Var i,j : Byte;
  Begin
    For i:=1 to N do
      Begin
        Dxh[i]:=True;
        Dxc[i]:=True;
      End;
    d:=0;
    For i:=1 to N do
      For j:=1 to N do
        Begin
          If A[i,j]=1 then
            Begin
              Inc(d);
              Tr[d].x:=i;
              Tr[d].y:=j;
            End;
          End;
        End;
      Tong:=0;
      LT:=0;
      Vet(1,1);
      For i:=1 to N do
        For j:=1 to N do B[i,j]:=1;
      For i:=1 to d do B[Tr[i].x,Tr[i].y] := 1;
      For i:=1 to LT do B[LKQ[i].x,LKQ[i].y] := 2;
      Writeln('Cach xep duoc nhieu xe nhat : ',LT);
      Hien(B);
    End;
  BEGIN
    Clrscr;
    Input;
    Hien(A);
    Khoitao;
    Write(#10#13,'Enter to quit ');
    Readln;
  END.

```

C11 B20 (Bài NETWORK OF SCHOOLS) - Đề thi quốc tế 1996

```

Uses  Crt;
Const  Max = 110;
       { Inp = 'c:\qt96\data\net\input9.txt'; }

```



```

        Tv[j] := h;
    End
End;
End;
Procedure Lam;
    Var F      : Text;
        s      : Byte;
        colap  : Boolean;
        i,scum1, scum2, scl , khac,p,T,LT      : Integer;
    Procedure Loai(i:Byte;Var s:Byte;gd:Byte);{ Lan tu truong i }
        Var k,j:Integer;
        Begin
            For k:=Tr[i-1]+1 to Tr[i] do
                Begin
                    { Xoa cum loai 1 }
                    j := Ra[k];
                    If Not D[j] then
                        Begin
                            D[j]:= True;
                            If R[j]=0 then Inc(s);
                            Loai(j,s,kk);
                        End;
                    End;
                End;
            If gd=1 then
                For k:=Tv[i-1]+1 to Tv[i] do { Xoa cum loai 2 }
                    Begin
                        j := Vao[k];
                        If Not D[j] then
                            Begin
                                D[j]:= True;
                                If R[j]=0 then Inc(s);
                                Loai(j,s,kk);
                            End;
                        End;
                    End;
                End;
            End;
        End;
    Begin
        Fillchar(D,sizeof(D),False);
        Assign(f,Out);
        Rewrite(f);
        scum1:=0; scum2:=0; T := 0; scl:=0; khac:= 0;
        For i:=1 to N do
            If Not D[i] and (V[i]=0) and (R[i]<>0) then
                Begin
                    D[i]:= True;
                    s:=0;
                    Loai(i,s,0);
                    If s>0 then T := T+s;
                    Inc(scum1);
                End;
        End;
    End;

```

```

For i:=1 to N do
  If Not D[i] and ((R[i]=0) and (V[i]<>0)) then
    Begin
      D[i]:= True;
      s:=0;
      Loai(i,s,1);
      If s>0 then T:=T+s;
      Inc(scum2);
    End;
  { Xoa nhung diem con lai : co lap hoac luan quan }
For i:=1 to N do
  If Not D[i] then
    Begin
      colap := False;
      If (V[i]=0) and (R[i]=0) then
        Begin
          Inc(scl); { k:so diem co lap hoac luan quan }
          colap := true;
        End;
      D[i]:= True;
      If Not colap then
        Begin
          Inc(khac);
          s:=0;
          Loai(i,s,1);
        End;
      End;
      LT := scum1+scum2+scl+khac;
      Writeln(f,LT);
      LT := T+scum1+scum2+scl+khac;
      If scum1+scum2+khac = 1 then LT := T;
      Writeln(LT);
      Close(F);
    End;
BEGIN
  Clrscr;
  DocF;
  Lam;
  Readln;
  Writeln('Da xong ');
END.

```

ĐẠI HỌC THAM KHẢO

TRIMINÔ

Bài 2 - Đề thi chọn đội tuyển Quốc gia năm 1997 (dự kỳ thi quốc tế tại Nam Phi)

Cho 1- ới ô vuông kích thước 8×8 và 21 thanh Triminô , mỗi thanh là một hình chữ nhật gồm 3 ô vuông , trên mỗi ô của thanh Triminô có một chữ số trong phạm vi từ 1 đến 8 .

Yêu cầu tìm cách xếp 21 quân Triminô này lên 1- ới , sao cho :

- Chỉ còn đúng 1 ô của 1- ới không bị phủ .
- Số có 8 chữ số tạo thành bằng cách đọc các giá trị số trên các ô của đường chéo bắt đầu từ góc trên trái và kết thúc ở góc phải d- ới là lớn nhất (Quy - ớc : ô không bị phủ đ- ọc coi là có chứa số 0).

Dữ liệu vào : Cho trên File văn bản 'TRIMINO.INP' gồm 21 dòng , mỗi dòng 3 chữ số có trên một quân Triminô , số thứ 2 là số ở giữa của Triminô.

Dữ liệu ra : Kết quả ghi lên File văn bản 'TRIMINO.OUT' theo cấu trúc :

- Dòng đầu ghi số tìm đ- ọc
- 8 dòng tiếp theo , mỗi dòng ứng với 1 hàng của 1- ới tính từ trên xuống , ghi 8 giá trị số trên các ô của hàng theo thứ tự từ trái qua phải .

'TRIMINO.INP'

1 1 7
1 3 6
1 2 3
1 1 4
1 7 1
1 8 3
1 3 6
1 6 3
1 3 4
1 7 3
1 2 7
1 7 8
1 8 7
1 2 2
1 1 5
1 7 6
1 6 5
1 6 5
1 6 8
1 6 3
1 7 4

'TRIMINO.OUT'

8 7 1 1 3 6 1 1
3 8 1 1 8 3 2 1
1 7 8 1 1 0 3 4
3 1 1 8 1 1 8 7
6 7 2 6 7 1 2 2
1 3 7 1 1 7 2 1
1 1 6 5 7 1 7 1
5 1 6 3 6 1 3 6

```

Uses  Crt;
Const  Fi    = 'Trimino.inp';
       Fo    = 'Trimino.out';
Type   Banco      = Array[1..8,1..8] of Byte;
       Mathanh    = Array[1..8,1..8] of Byte;
       Daxet      = Array[1..21] of Boolean;
       Thanh      = Array[1..21,1..4] of Byte;
Var     B,LB : Banco;
       M,LM : Mathanh;
       D      : Daxet;
       T      : Thanh;
       F      : Text;
       Ldcheo : LongInt;
       q,x,y,x1,x2,y1,y2,h1 : Byte;
Procedure TaoF;
  Var  i : Byte;
       F : Text;
  Begin
    Assign(F,Fi);
    ReWrite(F);
    For i:=1 to 21 do
      Writeln(F,Random(8)+1,' ',Random(8)+1,' ',Random(8)+1,'
');
    Close(F);
  End;
Procedure DocF;
  Var  F : Text;
       i,j : Byte;
  Begin
    Assign(F,Fi);
    {$I-} Reset(f);{$I+}
    If IoResult<>0 then
      Begin
        Writeln('Khong thay ',Fi);
        Readln;
        Halt;
      End;
    For i:=1 to 21 do
      Begin
        For j:=1 to 3 do Read(F,T[i,j]);
        Readln(F);
        T[i,4] := i;
      End;
    Close(F);
  End;
Procedure Timhuong(q: Byte;Var h1 : Byte);
  Var i,j,d1 : Byte;
  Begin

```

```

    x1 := 1; y1 := 1; x2 := 1; y2 := 1;
    If q=22 then Exit;
    d1 := 0;
    For i:=1 to 8 do
      For j:=1 to 8 do
        Begin
          If (M[i,j] = q) then
            If (d1=0) then
              Begin
                x1 := i; y1 := j;
                Inc(d1);
              End
            Else
              Begin
                x2 := i; y2 := j;
                Inc(d1);
                If d1=3 then
                  Begin
                    If y2>y1 then h1 := 1 Else h1 := 2;
                    Exit;
                  End;
                End;
              End;
            End;
          End;
        End;
      End;
    End;
  End;
Function Timvitri(i : Byte) : Byte; { Tim vi tri quan A[i,i] la 1,2,3 }
  Begin
    If M[i,i]=22 then Begin Timvitri := 0; Exit; End;
    x1 := 1; y1 := 1; x2 := 1; y2 := 1;
    Timhuong(M[i,i],h1); { Tim huong cua quan 8 }
    If (i=x1) and (i=y1) then Timvitri := 1 Else
    If (i=x2) and (i=y2) then Timvitri := 3 Else
    Timvitri := 2;
  End;
Function QMax(vt : Byte): Byte; { Tim thanh co phan tu max o vitri=vt }
  Var t1,i : Byte;
  Max : Byte;
  Begin
    Max := 0;
    If vt = 0 then Exit;
    For i:=1 to 21 do
      If (Not D[i]) then
        If vt in [1..3] then
          If (T[i,vt]> Max) then
            Begin
              T1 := T[i,4];
              Max := T[i,vt];
            End;
          End;
        End;
      End;
    End;
  QMax := T1;

```

```

End;
Procedure Doi(i : Byte);
  Var q1,q2,q3,vt1,k: Byte;  Nguoc : Boolean;
  Begin
    q1 := M[i,i];
    vt1 := Timvitri(i); { Duoc gia tri x1,x2,y1,y2,h1 }
    If vt1=2 then q2 := QMax(2)
    Else
      Begin
        q2 := QMax(1);
        q3 := Qmax(3);
        If q2<q3 then
          Begin
            q2 := q3;
            nguoc := True;
          End
        Else nguoc := False;
      End;
    End;
    If Not (q2 in [1..21] ) then Exit;
    D[q2] := True;
    If Not nguoc then
      Begin
        Case h1 of
          1: For k:=1 to 3 do B[x1,y1+k-1]:=T[q2,k];
          2: For k:=1 to 3 do B[x1+k-1,y1]:=T[q2,k];
        End;
      End
    Else
      {If nguoc then}
      Begin
        Case h1 of
          1: For k:=1 to 3 do B[x1,y1+k-1]:=T[q2,4-k];
          2: For k:=1 to 3 do B[x1+k-1,y1]:=T[q2,4-k];
        End;
      End;
    End;
  End;
Procedure Tim(Var x,y : Byte);
  Begin
    While (M[x,y]>0) and (x in [1..8]) and (y in [1..8]) do
      If y<8 then Inc(y)
      Else If x<8 then
        Begin Inc(x);y:=1;End;
      End;
  End;
Function Chapnhan(x,y,hg : Byte): Boolean;
  Var i : Byte;
  Begin
    Chapnhan := False;
    If ((hg=1) and (y>6)) or ((hg=2) and (x>6)) then Exit;

```

```

        Case hg of
            1 : For i:=1 to 3 do If M[x,y+i-1]>0 then Exit;
            2 : For i:=1 to 3 do If M[x+i-1,y]>0 then Exit;
        End;
        Chapnhan := True;
    End;
Procedure Dat(x,y,hg : Byte);
    Var i : Byte;
    Begin
        Case hg of
            1 : For i:=1 to 3 do M[x,y+i-1] := T[q,4];
            2 : For i:=1 to 3 do M[x+i-1,y] := T[q,4];
        End;
    End;
Function Duongcheo(B : Banco): LongInt;
    Var dc: LongInt; i : Byte;
    Begin
        dc := 0;
        For i:=1 to 8 do
            If (B[i,i]= 0) then dc := dc*10
            Else dc := dc*10+B[i,i];
        Duongcheo := dc;
    End;
Procedure Xoa(x,y,hg : Byte);
    Var i : Byte;
    Begin
        Case hg of
            1 : For i:=1 to 3 do M[x,y+i-1] := 0;
            2 : For i:=1 to 3 do M[x+i-1,y] := 0;
        End;
    End;
Procedure GhiLB;
    Var i,j : Byte;
    Begin
        For i:=1 to 8 do
            Begin
                For j:=1 to 8 do Write(F,LB[i,j]:3);
                Writeln(F);
            End;
        End;
    End;
Procedure GhiLM;
    Var i,j : Byte;
    Begin
        For i:=1 to 8 do
            Begin
                For j:=1 to 8 do Write(F,LM[i,j]:3);
                Writeln(F);
            End;
        End;
    End;

```

```

        Writeln(F);
    End;
Procedure Ghitoiuu;
    Var i : Byte; p : LongInt;
    Begin
        FillChar(D,Sizeof(D),False);
        FillChar(B,Sizeof(B),0);
        For i:=1 to 8 do Doi(i);
        p := duongcheo(B);
        If p>Ldcheo then
            Begin
                Ldcheo := p;
                LB      := B;
                LM      := M;
            End;
    End;
End;
Procedure Vet(x,y : Byte);
    Var hg,i,j : Byte;
    Begin
        Tim(x,y);
        For hg := 1 to 2 do
            If chapnhan(x,y,hg) then
                Begin
                    Inc(q);
                    Dat(x,y,hg);
                    If q=21 then
                        Ghitoiuu Else Vet(x,y);
                    Dec(q);
                    Xoa(x,y,hg);
                End;
        End;
    End;
End;
Procedure Datnot;
    Var i,j,k,dem : Byte;
    Begin
        FillChar(D,Sizeof(D),False);
        For i:=1 to 8 do D[LM[i,i]]:= True;
        For k:=1 to 21 do
            If Not D[k] then
                Begin
                    dem := 0;
                    For i:=1 to 8 do
                        For j:=1 to 8 do
                            If LM[i,j]=k then
                                Begin
                                    Inc(dem);
                                    LB[i,j]:= T[k,dem];
                                End;
                End;
        End;
    End;
End;

```

```

End;
BEGIN
  Clrscr; {      TaoF; }
  DocF;    Assign(F,Fo);      ReWrite(F);
  Ldcheo := 0;
  Writeln('Please wait ... ');
  For x:=1 to 8 do
    For y:=1 to 8 do
      Begin
        FillChar(M,Sizeof(M),0);
        FillChar(B,Sizeof(B),0);
        q := 0;
        M[x,y] := 22;
        Vet(1,1);
      End;
    Datnot;
    GhiLM;
    GhiLB;
    Close(F);
    Writeln('Da xong ');
    Readln;
  END.

```

Bài trên làm theo sơ đồ sau :

1 - Cho ô trống tùy ý trên bàn cờ , coi các Triminô nh- nhau (nghĩa là không để ý tới các số trên chúng) , đặt 21 quân Triminô lên bàn cờ , sẽ được kết quả đầu tiên là : chỉ khi ô trống ở vào các vị trí (3,3) ; (3,6) ; (6,3) ; (6,6) thì mới đặt đ- ợc . Tất cả có 1424 cách đặt theo kiểu này (tạm gọi mỗi cách là 1 cấu hình của bàn cờ).

2 - Với mỗi cách đặt trên , bây giờ xếp các Triminô lần l- ợt vào các vị trí trên đ- ờng chéo từ góc trên_ trái cho đến góc d- ối_ phải , sao cho tại mỗi vị trí là tốt nhất :

+ Xem ô (i,i) đang xét là ô ở vị trí thứ mấy trong thanh Triminô Ti chứa ô (i,i) của cấu hình đang xét .(gọi vị trí này là vt)

+ Duyệt các Triminô ch- a dùng trong 21 Triminô , tìm thanh nào có số lớn nhất ở vị trí vt . Nếu vt=1 hoặc 3 thì phải tìm số lớn nhất ở cả 2 vị trí 1 và 3 .Gọi thanh tìm đ- ợc là thanh Tx

+ Trên bàn cờ thay t- ơng ứng thanh Ti bằng thanh Tx , xoá thanh Tx vì đã sử dụng

3 - Tính đ- ờng chéo , nếu thấy tốt hơn thì l- u lại bàn cờ và cấu hình t- ơng ứng

4 - Đặt nốt các thanh Triminô ch- a dùng vào bàn cờ theo l- u cấu hình (chỉ cần 1 cách đặt nốt)

PHẦN 3

CÂY - CÂY KHUNG NGỌN NHỎ

I/ Định nghĩa :

Cây là đồ thị hữu hạn, vô hướng, liên thông, không có chu trình, có ít nhất 2 đỉnh.

II / Tính chất :

1 - Định lý 1 :

Nếu H là cây có N đỉnh thì H có các tính chất sau đây :

- Thêm vào H một cạnh nối 2 đỉnh bất kỳ không kề nhau, H sẽ xuất hiện chu trình.
- Bớt đi 1 cạnh trong H thì H không liên thông
- Giữa 2 đỉnh bất kỳ của H luôn tồn tại 1 đường đi duy nhất (vậy H là đồ thị đơn)
- H có $N-1$ cạnh

2 - Định lý 2 :

Nếu đồ thị G liên thông có N đỉnh và $N-1$ cạnh thì G là cây.

Vậy cây là đồ thị liên thông có chu số bằng 0 (suy từ công thức Ôle)

3 - Ghi chú :

Từ 1 đồ thị có thể hình thành nhiều cây khác nhau (gọi là các cây khung của đồ thị). Trong số các cây khung của đồ thị, có 1 cây được tạo ra một cách đơn giản nhất sau : nối 1 đỉnh với $n-1$ đỉnh còn lại !

Số cây khung của 1 đồ thị đầy đủ là N^{n-2} (N số đỉnh)

Số cây khung của một đồ thị có hữu hạn đỉnh là một số hữu hạn, nên luôn tìm được ít nhất 1 cây khung có tổng độ dài nhỏ nhất (nguyên lý biên). Ta gọi cây khung này là cây khung ngắn nhất.

Bài toán tìm cây khung ngắn nhất là một bài toán gặp trong thực tế :

Thí dụ : Xây dựng mạng dây điện thoại nối N thành phố sao cho 2 thành phố bất kỳ liên lạc được với nhau và tổng đường dây điện ngắn nhất. Đó là bài toán tìm cây khung ngắn nhất. Ngược lại : Xây dựng mạng dây điện thoại nối N thành phố sao cho 2 thành phố bất kỳ liên lạc được với nhau và tổng độ tin cậy trên các đường dây điện là lớn nhất. Đó là bài toán tìm cây khung dài nhất.

III / Thuật toán Prim tìm cây khung nhỏ nhất :

B-ớc 1 : Khởi trị - Lấy 1 đỉnh i tùy ý đưa vào tập đỉnh của cây. Khi đó tập đỉnh của cây là $\mathcal{D} = \{i\}$. Tập cạnh của cây là $C = \emptyset$ (Tập rỗng)

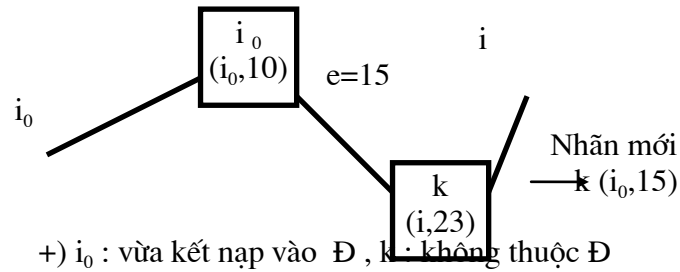
B-ớc 2 : Gán nhãn - Với mỗi đỉnh k không thuộc \mathcal{D} , ta gán cho nó nhãn $k(i, d)$ trong đó i là tên đỉnh thuộc \mathcal{D} , kề với k , gần k nhất, còn d là khoảng cách giữa i và k . Nếu trong \mathcal{D} không tìm được đỉnh i kề với k thì gán cho k nhãn $k(0, \infty)$.

B-ớc 3 : Kết nạp - Chọn đỉnh k không thuộc tập \mathcal{D} , có nhãn d nhỏ nhất, kết nạp k vào \mathcal{D} . Vậy $\mathcal{D} = \mathcal{D} + \{k\}$. Nhãn của k là $k(i, d)$ thì kết nạp cạnh (i, k) vào tập cạnh C . Vậy $C = C + \{ \text{cạnh}(i, k) \}$. Gọi đỉnh k vừa kết nạp là i_0 .

Nếu số đỉnh của \mathcal{D} bằng N thì kết thúc, còn không chuyển sang b-ớc 4

B-ớc 4 : Sửa nhãn - Với mọi đỉnh k ch- a thuộc \mathcal{D} có nhãn là $k(i, d)$ mà k kề với i_0 - là đỉnh vừa đ- ợc kết nạp vào tập đỉnh ở b- ớc 3 - ta sửa lại nhãn của k theo nguyên tắc sau : Gọi độ dài cung (i_0, k) là e

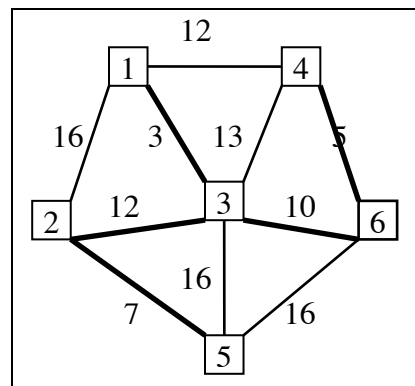
Nếu $d > e$ thì đỉnh k có nhãn mới là $k(i_0, e)$



Thí dụ :

File dữ liệu vào : PRIM.INT

```
6
0 16 3 12 0 0
16 0 12 0 7 0
3 12 0 13 16 10
12 0 13 0 0 5
0 7 16 0 0 16
0 0 10 5 16 0
```



File dữ liệu ra : PRIM.OUT

$(1, 3)= 3$ $(3, 6)= 10$ $(6, 4)= 5$ $(3, 2)= 12$ $(2, 5)= 7$

Tổng giá trị cây khung ngắn nhất là 37

```
Uses  Crt;
Const  Fi    = 'prim.txt';
        Fo    = 'prim.out';
        Max   = 200;
Var    A      : Array[1..Max,1..Max] of Byte;
        D      : Array[1..Max] of Boolean;
```

```

        C      : Array[0..Max] of record x1,x2 : Byte; end;
        Nh     : Array[1..Max] of record truoc,giatri : Byte; end;
        N,dd,socanh : Byte;
        { canh : Integer; }
    {-----}
    Procedure DocF;
    Var    f      : Text;
          i,j     : Byte;
    Begin
        Assign(f,fi);
        Reset(f);
        Readln(f,n);
        For i:=1 to n do
            Begin
                For j:=1 to n do read(f,a[i,j]);
                Readln(f);
            End;
        Close(f);
    End;
    {-----}
    Procedure Napdinh1;
    Begin
        Fillchar(d,sizeof(d),False);
        d[1] := True;
        dd := 1;
    End;
    {-----}
    Function Min(xj : Byte): Byte;
    Var    xi,p,i : Byte;
    Begin
        xi := 0; p := 255;
        For i:=1 to N do
            If d[i] then
                If (p>a[i,xj]) and (a[i,xj]>0) then
                    Begin
                        xi := i;    p := a[i,xj];
                    End;
        Min := xi;
    End;
    {-----}
    Procedure Gannhan;
    Var    xi,xj : Byte;
    Begin
        For xj:=1 to N do
            If not d[xj] then
                Begin
                    xi := Min(xj);
                    If (xi>0) and (A[xi,xj]>0) then

```

```

    Begin
        nh[xj].truoc := xi;
        nh[xj].giatri:= A[xi,xj];
    End
Else
    If xi=0 then
        Begin
            nh[xj].truoc := 0;
            nh[xj].giatri:= 255;
        End;
    End;
End;
{ ----- }
Procedure Ketnapthem;
Var    p,j,xj : Byte;
Begin
    p := 255;
    For j:=1 to n do
        If not d[j] then
            Begin
                If (nh[j].giatri<p) then
                    Begin
                        xj := j;
                        p := nh[j].giatri;
                    End;
            End;
        d[xj] := True;
        Inc(socanh);
        c[socanh].x1 := nh[xj].truoc;
        c[socanh].x2 := xj;
        dd := xj;
    End;
    { ----- }
Procedure Suanhan;
Var    xj : Byte;
Begin
    For xj:=1 to N do
        If (not D[xj]) and (A[xj,dd]>0) then
            Begin
                If Nh[xj].giatri>A[xj,dd] then
                    Begin
                        Nh[xj].truoc := dd;
                        Nh[xj].giatri:= A[xj,dd];
                    End;
            End;
    End;
End;
{ ----- }
Procedure Hiencanh;

```

```

Var   i,p : Byte;f : Text;
Begin
  Assign(f,fo);
  Rewrite(f);p:=0;
  For i:=1 to n-1 do
    Begin
      p := A[c[i].x1,c[i].x2]+p;
      Write(f,('c[i].x1:2,',c[i].x2:2,')=',A[c[i].x1,c[i].x2]:3,' ':3);
    End;
    Writeln(f);
    Writeln(f,'Tong gia tri cay khung ngan nhat la ',p);
    Close(f);
  End;
  {-----}
Procedure TT_Prim;
Var   Ok : Boolean;
Begin
  SoCanh := 0;
  Fillchar(nh,sizeof(nh),0);
  Napdinh1;
  Gannhan;
  Ok := False;
  Repeat
    Ketnapthem;
    If Socanh=N-1 then Ok:= True
    Else Suanhan;
  Until Ok;
  Hiencanh;
End;
{-----}
BEGIN
  Clrscr;
  DocF;
  TT_Prim
END.

```

PHẦN 4

TÌM ĐƯỜNG ĐI NGẮN NHẤT

THUẬT TOÁN DIJKSTRA VÀ FORD-BELLMAN

Một bài toán thường gặp trên đồ thị là tìm đường đi ngắn nhất từ đỉnh thứ nhất (ký hiệu là x_p) tới đỉnh thứ hai (ký hiệu là d). Khi vét cạn duyệt mọi đường đi từ x_p tới d , nếu không chú ý các cạnh (trên hoặc dưới) thích hợp để tránh các đường đi không tới đích, có thể duyệt không hết đồ thị khi đồ thị nhiều cung. Sau đây là 2 thuật toán giúp tránh tình trạng đó trong nhiều đồ thị.

I / Thuật toán Dijkstra (gán nhãn) :

T- t- ờng của thuật toán là trong quá trình xây dựng đ- ờng đi từ xp tới đ ,luôn kết hợp với việc chọn lựa đ- ờng đi để nó tốt dần lên bằng cách thay đổi liên tục nhãn tại các đỉnh .Mỗi đỉnh i sẽ có nhãn gồm 2 đặc tr- ng : Đặc tr- ng 1 ghi nhận đỉnh kề đi tới i , đặc tr- ng 2 ghi nhận độ dài đ- ờng đi ngắn nhất từ đỉnh xp tới đỉnh i này . Do đó khi tới đỉnh cuối cùng ta có ngay đ- ờng đi ngắn nhất . Các b- ớc của thuật toán nh- sau :

B- ớc 1 - Khởi trị :

+ Nhãn đỉnh xuất phát là $x_p(0,0)$: đỉnh đi tới đỉnh xp là đỉnh 0 ,đ- ờng đi đã qua là 0 .Các đỉnh i còn lại có nhãn là $i(0, \infty)$: có nghĩa đỉnh tới i là đỉnh 0 , đ- ờng đi đã qua tới i là vô cùng lớn .

+ Khởi trị mảng đánh dấu : Các đỉnh đều ch- a tới .

B- ớc 2 - Sửa nhãn :

Vòng lặp :

Begin

+ Chọn một đỉnh i trong các đỉnh ch- a tới và có nhãn độ dài nhỏ nhất . Đánh dấu đã tới đỉnh i.

+ Sửa lại nhãn các đỉnh k ch- a tới theo công thức quy hoạch động

$$\text{Nhãn}[k] = \text{Min} \{ \text{Nhãn}[k], \text{Nhãn}[i] + A[i,k] \}$$

End;

Cho đến khi tới đỉnh đích .

B- ớc 3 - Lần ng- ọc ,hiên đ- ờng đi ngắn nhất :

+ Bắt đầu : đỉnh := đ ; cs := 1 ; KQ[cs] := đỉnh ;

+ Vòng lặp

Begin

đỉnh := Nhãn thứ nhất của đỉnh ;

Inc(cs);

KQ[cs] := đỉnh;

End;

Cho đến khi đỉnh = xp;

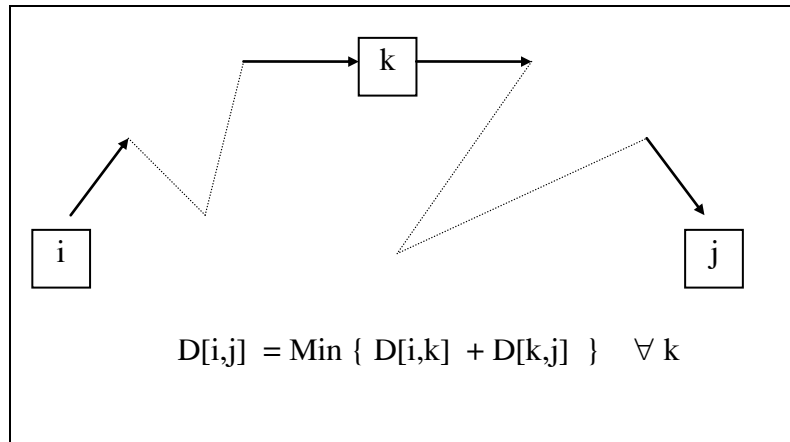
+ Duyệt ng- ọc mảng KQ để hiện hành trình

+ Hiện độ dài đ- ờng đi .

II / Thuật toán Ford - BellMan :

Bảng 3 vòng For đơn giản , thuật toán đã thể hiện tinh thần quy hoạch động một cách “ đẹp để bắt ngờ “ :

Với 2 đỉnh i và j ($1 \leq i, j \leq N$) , đ- ờng đi ngắn nhất từ i tới j là $D[i,j]$ rõ ràng là đại l- ợng nhỏ nhất trong các tổng : $D[i,k] + D[k,j]$ trong đó k là mọi đỉnh trung gian (con đ- ờng đi từ i tới j sẽ đi qua k) .



```

Procedure DgdiFB;
Var i,j,k : Integer;
Begin
  For k:=1 to N do
    For i:=1 to N do
      For j := 1 to N do
        if A[i,k]^dd + A[i,k]^dd < A[i,j]^dd then
          Begin
            A[i,j]^dd := A[i,k]^dd + A[i,k]^dd ;
            A[i,j]^đỉnh := k;
          End;
        End;
      End;
    End;
  End;

```

III / Bài tập mẫu :

Bài 1 : Cho đồ thị vô hướng liên thông từ File “DGDI.INP” tổ chức như sau :

+ Dòng thứ nhất ghi 3 số : N,xp,đ (số đỉnh , tên đỉnh xuất phát , đỉnh đích)

+ Các dòng tiếp theo : mỗi dòng 3 số : i,j , A[i,j] (A[i,j] là khoảng cách i tới j)

Nếu i=0 thì kết thúc dữ liệu về đồ thị này

Bằng thuật toán Di jsktra tìm đ- ờng đi ngắn nhất từ xp tới đ

Bài 2 : Nội dung nh- trên nh- ng tìm đ- ờng đi ngắn nhất bằng thuật toán For-Bellman

Lời giải :

Bài 1 : Bằng thuật toán Di jsktra tìm đ- ờng đi ngắn nhất

Uses Crt;

Const Max = 100;

Fi = 'duongdi.inp';

Type Ta = Array[1..Max,1..Max] of Integer;

Re = Record

t : Byte;

h : Word;

End;

```

    Nhan  = Array[0..Max] of Re;
    Dau   = Array[1..Max] of Boolean;
Var    N,xp,d : Byte;
    A      : ^Ta;
    F      : Text;
Procedure DocF;
    Var i,j : Byte;
    Begin
        Assign(F,Fi);
        Reset(F);
        Readln(F,N,xp,d);
        New(A);
        For i:=1 to N do
            For j:=1 to n do A^[i,j] := MaxInt;
        While not Seekeof(F) do
            Begin
                Read(F,i,j);
                If i=0 then
                    Begin Close(F);Exit;End;
                Readln(F,A^[i,j]);
            End;
        For i:=1 to N do A^[i,i] := 0;
        Close(F);
    End;
Procedure Lam;
    Var NH    : Nhan;
        dd    : Dau;
        i,j    : Byte;
    Procedure Khoitao;
        Var i    : Byte;
        Begin
            For i:=1 to N do
                Begin
                    NH[i].h := MaxInt;
                    DD[i]  := False;
                End;
            NH[xp].h := 0;
            NH[xp].t := 0;

        End;
    Function Min : Byte;
    Var i,k    : Byte;
    Begin
        i := 0;
        For k:=1 to N do
            If (Not DD[k]) and (NH[k].h<NH[i].h) then i := k;
        Min := i;
    End;

```

```

Procedure Sua(i : Byte); { i : dinh cuoi cua hanh trinh hien tai }
Var j      : Byte;
Begin
  DD[i] := True;
  For j:=1 to N do
    If (Not DD[j]) and (NH[j].h>NH[i].h+A^[i,j]) then
      Begin
        NH[j].h := NH[i].h+A^[i,j];
        NH[j].t := i;
      End;
  End;
End;
Procedure Lannguoc;
Var S      : String;
    i,j    : Byte;
Begin
  i := d;
  S := "";
  While i>0 do
    Begin
      S := chr(i)+S;
      i := NH[i].t;
    End;
  For i:=1 to Length(S) do Write(Ord(S[i]),' ');
End;
Begin
  Clrscr;
  Khoitao;
  While Not DD[d] do
    Begin
      i := Min;
      If i=0 then
        Begin
          Writeln('vo nghiem ');
          Exit;
        End;
      Sua(i);
    End;
  Lannguoc;
End;
BEGIN
  Clrscr;
  DocF;
  Lam;
  Dispose(A);
  Writeln('Da xong ');
  Readln;
END.

```

Input

8 1 8
1 2 3
2 1 3
1 3 5
3 1 5
1 4 2
4 1 2
2 3 1
3 2 1
2 5 7
5 2 7
3 4 4
4 3 4
3 5 5
5 3 5
4 6 3
6 4 3
5 8 3
8 5 3
6 7 4
7 6 4
6 8 6
8 6 6
7 8 5
8 7 5
6 3 1
6 5 2
7 4 6
0

OUT

Nếu $x_p=1, d=8$ thì có đ-ờng đi 1 4 6 5 8

Nếu $x_p=8, d=1$ thì có đ-ờng đi 8 6 3 2 1

Bài 2 : Bằng thuật toán For-Bellman tìm đ-ờng đi ngắn nhất từ x_p tới đ

Uses Crt;

Const Max = 100;

Fi = 'Duongdi.inp';

Type Ta = Array[1..Max, 1..Max] of Record h : Word; tg : Byte; End;

Dau = Array[1..Max] of Boolean;

Var N, xp, t : Integer;

A : ^Ta;

F : Text;

Procedure DocF;

Var i, j : Byte;

Begin

Assign(F, Fi);

Reset(F);

New(A);

Readln(F, N, xp, t);

For i:=1 to N do

For j:=1 to N do

Begin

A^[i, j].h := MaxInt;

A^[i, j].tg := 0;

End;

For i:=1 to N do A^[i, i].h := 0;

While Not SeekEof(F) do

Begin

Read(F, i, j);

If i=0 then

Begin

Close(F);

Exit;

End;

Readln(F, A^[i, j].h);

End;

Close(F);

End;

Procedure FB;

Var i, j, k : Integer;

Begin

For k:=1 to N do

For i:=1 to N do

For j:=1 to N do

If (A^[i, k].h + A^[k, j].h < A^[i, j].h) then

Begin

A^[i, j].h := A^[i, k].h + A^[k, j].h;

```

        A^[i,j].tg := k;
    End;
End;
Procedure Lannguoc;
Var S : String;
    i,x1,y1 : Byte;
Begin
    If A^[xp,t].h = MaxInt then
    Begin
        Writeln('Vo nghiem ');
        Exit;
    End;
    S := Char(xp)+char(t);
    i := 1;
    While i<Length(S) do
    Begin
        x1 := Ord(S[i]);
        y1 := Ord(S[i+1]);
        If A^[x1,y1].tg=0 then Inc(i)
        Else Insert(Char(A^[x1,y1].tg),S,i+1);
    End;
    For i:=1 to Length(S) do Write(Ord(S[i]):4);
    Writeln;
    Writeln('Do dai : ',A^[xp,t].h);
End;
BEGIN
    Clrscr;
    DocF;
    FB;
    Lannguoc;
    Dispose(A);
END.

```

PHẦN 3 CÂY - CÂY KHUNG NGỌN NHỎ

I/ Định nghĩa :

Cây là đồ thị hữu hạn , vô hướng , liên thông , không có chu trình , có ít nhất 2 đỉnh .

II / Tính chất :

1 - Định lý 1 :

Nếu H là cây có N đỉnh thì H có các tính chất sau đây :

- a) Thêm vào H một cạnh nối 2 đỉnh bất kỳ không kề nhau , H sẽ xuất hiện chu trình .
- b) Bớt đi 1 cạnh trong H thì H không liên thông
- c) Giữa 2 đỉnh bất kỳ của H luôn tồn tại 1 đường đi duy nhất (vậy H là đồ thị đơn)

d) H có N-1 cạnh

2 - Định lý 2 :

Nếu đồ thị G liên thông có N đỉnh và N-1 cạnh thì G là cây .
 Vậy cây là đồ thị liên thông có chu số bằng 0 (suy từ công thức Ôle)

3 - Ghi chú :

Từ 1 đồ thị có thể hình thành nhiều cây khác nhau (gọi là các cây khung của đồ thị) . Trong số các cây khung của đồ thị , có 1 cây đ-ợc tạo ra một cách đơn giản nh- sau : nối 1 đỉnh với n-1 đỉnh còn lại !

Số cây khung của 1 đồ thị đầy đủ là N^{n-2} (N số đỉnh)
 Số cây khung của một đồ thị có hữu hạn đỉnh là một số hữu hạn ,nên luôn tìm đ-ợc ít nhất 1 cây khung có tổng độ dài nhỏ nhất (nguyên lý biên) . Ta gọi cây khung này là cây khung ngắn nhất .

Bài toán tìm cây khung ngắn nhất là một bài toán gặp trong thực tế :

Thí dụ : Xây dựng mạng dây điện thoại nối N thành phố sao cho 2 thành phố bất kỳ liên lạc đ-ợc với nhau và tổng đ-ờng dây điện ngắn nhất .Đó là bài toán tìm cây khung ngắn nhất . Ng-ợc lại : Xây dựng mạng dây điện thoại nối N thành phố sao cho 2 thành phố bất kỳ liên lạc đ-ợc với nhau và tổng độ tin cậy trên các đ-ờng dây điện là lớn nhất .Đó là bài toán tìm cây khung dài nhất .

III / Thuật toán Prim tìm cây khung nhỏ nhất :

B-ớc 1 : Khởi trị - Lấy 1 đỉnh i tùy ý đ- a vào tập đỉnh của cây . Khi đó tập đỉnh của cây là $\mathcal{D} = \{i\}$. Tập cạnh của cây là $C = \emptyset$ (Tập rỗng)

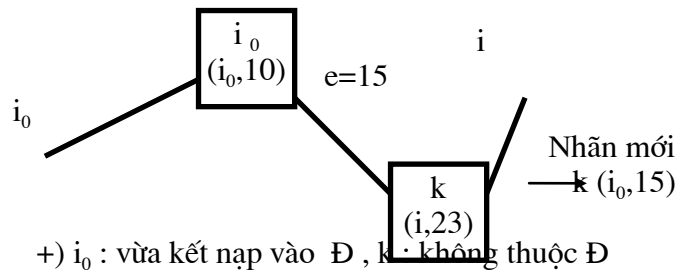
B-ớc 2 : Gán nhãn - Với mỗi đỉnh k không thuộc \mathcal{D} , ta gán cho nó nhãn $k(i, d)$ trong đó i là tên đỉnh thuộc \mathcal{D} ,kề với k , gần k nhất , còn d là khoảng cách giữa i và k . Nếu trong \mathcal{D} không tìm đ-ợc đỉnh i kề với k thì gán cho k nhãn $k(0, \infty)$.

B-ớc 3 : Kết nạp - Chọn đỉnh k không thuộc tập \mathcal{D} , có nhãn d nhỏ nhất , kết nạp k vào \mathcal{D} .Vậy $\mathcal{D} = \mathcal{D} + \{k\}$. Nhãn của k là $k(i, d)$ thì kết nạp cạnh (i, k) vào tập cạnh C .
 Vậy $C = C + \{ \text{cạnh}(i, k) \}$. Gọi đỉnh k vừa kết nạp là i_0 .

Nếu số đỉnh của \mathcal{D} bằng N thì kết thúc , còn không chuyển sang b-ớc 4

B-ớc 4 : Sửa nhãn - Với mọi đỉnh k ch- a thuộc \mathcal{D} có nhãn là $k(i, d)$ mà k kề với i_0 - là đỉnh vừa đ-ợc kết nạp vào tập đỉnh ở b-ớc 3 - ta sửa lại nhãn của k theo nguyên tắc sau :
 Gọi độ dài cung (i_0, k) là e

Nếu $d > e$ thì đỉnh k có nhãn mới là $k(i_0, e)$

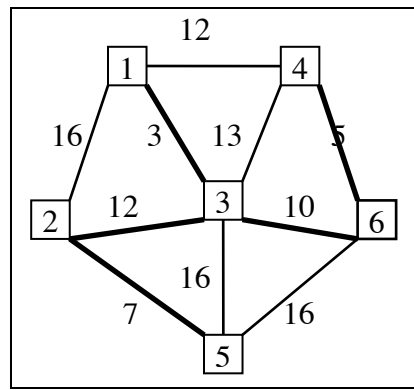


Thí dụ :

File dữ liệu vào : PRIM.INT

6

```
0 16 3 12 0 0
16 0 12 0 7 0
3 12 0 13 16 10
12 0 13 0 0 5
0 7 16 0 0 16
0 0 10 5 16 0
```



File dữ liệu ra : PRIM.OUT

(1, 3)= 3 (3, 6)= 10 (6, 4)= 5 (3, 2)= 12 (2, 5)= 7

Tổng giá trị cây khung nhỏ nhất là 37

```
Uses Crt;
Const Fi = 'prim.txt';
      Fo = 'prim.out';
      Max = 200;
Var A : Array[1..Max,1..Max] of Byte;
    D : Array[1..Max] of Boolean;
    C : Array[0..Max] of record x1,x2 : Byte; end;
    Nh : Array[1..Max] of record truoc,giatri : Byte; end;
    N,dd,socanh : Byte;
    { canh : Integer; }
{-----}
Procedure DocF;
```

```

Var    f      : Text;
       i,j    : Byte;
Begin
  Assign(f,fi);
  Reset(f);
  Readln(f,n);
  For i:=1 to n do
    Begin
      For j:=1 to n do read(f,a[i,j]);
      Readln(f);
    End;
  Close(f);
End;
{-----}
Procedure Napdinh1;
Begin
  Fillchar(d,sizeof(d),False);
  d[1] := True;
  dd := 1;
End;
{-----}
Function Min(xj : Byte): Byte;
Var    xi,p,i : Byte;
Begin
  xi := 0; p := 255;
  For i:=1 to N do
    If d[i] then
      If (p>a[i,xj]) and (a[i,xj]>0) then
        Begin
          xi := i;  p := a[i,xj];
        End;
  Min := xi;
End;
{-----}
Procedure Gannhan;
Var    xi,xj : Byte;
Begin
  For xj:=1 to N do
    If not d[xj] then
      Begin
        xi := Min(xj);
        If (xi>0) and (A[xi,xj]>0) then
          Begin
            nh[xj].truoc := xi;
            nh[xj].giatri:= A[xi,xj];
          End
        Else
          If xi=0 then

```

```

        Begin
            nh[xj].truoc := 0;
            nh[xj].giatri:= 255;
        End;
    End;
End;
{-----}
Procedure Ketnapthem;
Var    p,j,xj : Byte;
Begin
    p := 255;
    For j:=1 to n do
        If not d[j] then
            Begin
                If (nh[j].giatri<p) then
                    Begin
                        xj := j;
                        p := nh[j].giatri;
                    End;
            End;
        d[xj] := True;
        Inc(socanh);
        c[socanh].x1 := nh[xj].truoc;
        c[socanh].x2 := xj;
        dd := xj;
    End;
{-----}
Procedure Suanhan;
Var    xj : Byte;
Begin
    For xj:=1 to N do
        If (not D[xj]) and (A[xj,dd]>0) then
            Begin
                If Nh[xj].giatri>A[xj,dd] then
                    Begin
                        Nh[xj].truoc := dd;
                        Nh[xj].giatri:= A[xj,dd];
                    End;
            End;
    End;
End;
{-----}
Procedure Hiencanh;
Var    i,p : Byte;f : Text;
Begin
    Assign(f,fo);
    Rewrite(f);p:=0;
    For i:=1 to n-1 do
        Begin

```

```

    p := A[c[i].x1,c[i].x2]+p;
    Write(f,('c[i].x1:2,',c[i].x2:2,')=',A[c[i].x1,c[i].x2]:3,' ':3);
End;
Writeln(f);
Writeln(f,'Tong gia tri cay khung ngan nhat la ',p);
Close(f);
End;
{-----}
Procedure TT_Prim;
Var   Ok : Boolean;
Begin
    SoCanh := 0;
    Fillchar(nh,sizeof(nh),0);
    Napdinh1;
    Gannhan;
    Ok := False;
    Repeat
        Ketnapthem;
        If Socanh=N-1 then Ok:= True
        Else Suanhan;
    Until Ok;
    Hiencanh;
End;
{-----}
BEGIN
    Clrscr;
    DocF;
    TT_Prim
END.

```

PHẦN 4

TÌM ĐƯỜNG NGẮN NHẤT

THUẬT TOÁN DIJKSTRA VÀ FORD-BELLMAN

Một bài toán thường gặp trên đồ thị là tìm đường đi ngắn nhất từ đỉnh thứ nhất (ký hiệu là x_p) tới đỉnh thứ hai (ký hiệu là d). Khi vét cạn duyệt mọi đường đi từ x_p tới d , nếu không chú ý các cạnh (trên hoặc dưới) thích hợp để tránh các đường đi không tới đích, có thể duyệt không hết đồ thị nhiều cung. Sau đây là 2 thuật toán giúp tránh tình trạng đó trong nhiều đồ thị.

1/ Thuật toán Dijkstra (gán nhãn) :

T- t- ờng của thuật toán là trong quá trình xây dựng đường đi từ x_p tới d , luôn kết hợp với việc chọn lựa đường đi để nó tốt dần lên bằng cách thay đổi liên tục nhãn tại các đỉnh. Mỗi đỉnh i sẽ có nhãn gồm 2 đặc trưng : Đặc trưng 1 ghi nhận đỉnh kề đi tới i , đặc

tr- ng 2 ghi nhận độ dài đ- ờng đi ngắn nhất từ đỉnh x_p tới đỉnh i này . Do đó khi tới đỉnh cuối cùng ta có ngay đ- ờng đi ngắn nhất . Các b- ớc của thuật toán nh- sau :

B- ớc 1 - Khởi trị :

+ Nhãn đỉnh xuất phát là $x_p(0,0)$: đỉnh đi tới đỉnh x_p là đỉnh 0 ,đ- ờng đi đã qua là 0 .Các đỉnh i còn lại có nhãn là $i(0, \infty)$: có nghĩa đỉnh tới i là đỉnh 0 , đ- ờng đi đã qua tới i là vô cùng lớn .

+ Khởi trị mảng đánh dấu : Các đỉnh đều ch- a tới .

B- ớc 2 - Sửa nhãn :

Vòng lặp :

Begin

+ Chọn một đỉnh i trong các đỉnh ch- a tới và có nhãn độ dài nhỏ nhất . Đánh dấu đã tới đỉnh i .

+ Sửa lại nhãn các đỉnh k ch- a tới theo công thức quy hoạch động

$$\text{Nhãn}[k] = \text{Min} \{ \text{Nhãn}[k], \text{Nhãn}[i] + A[i,k] \}$$

End;

Cho đến khi tới đỉnh đích .

B- ớc 3 - Lần ng- ọc ,hiện đ- ờng đi ngắn nhất :

+ Bắt đầu : đỉnh := đ ; cs := 1 ; KQ[cs] := đỉnh ;

+ Vòng lặp

Begin

đỉnh := Nhãn thứ nhất của đỉnh ;

Inc(cs);

KQ[cs] := đỉnh;

End;

Cho đến khi đỉnh = x_p ;

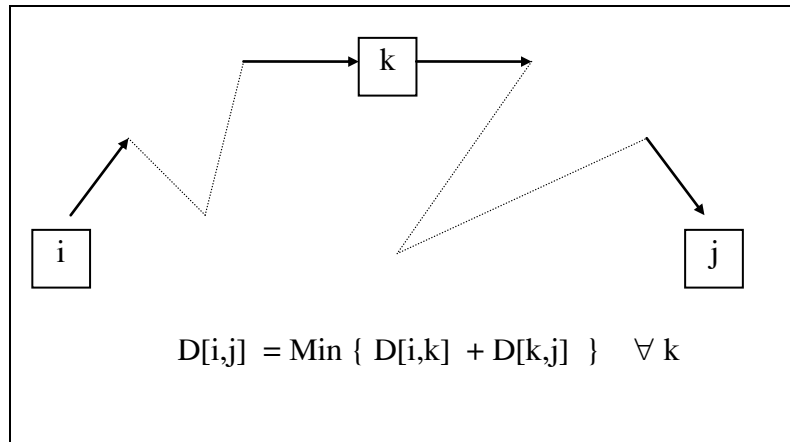
+ Duyệt ng- ọc mảng KQ để hiện hành trình

+ Hiện độ dài đ- ờng đi .

II / Thuật toán Ford - BellMan :

Bảng 3 vòng For đơn giản , thuật toán đã thể hiện tinh thần quy hoạch động một cách “ đẹp để bắt ngờ “ :

Với 2 đỉnh i và j ($1 \leq i, j \leq N$) , đ- ờng đi ngắn nhất từ i tới j là $D[i,j]$ rõ ràng là đại l- ợng nhỏ nhất trong các tổng : $D[i,k] + D[k,j]$ trong đó k là mọi đỉnh trung gian (con đ- ờng đi từ i tới j sẽ đi qua k).



```

Procedure DgdiFB;
Var i,j,k : Integer;
Begin
  For k:=1 to N do
    For i:=1 to N do
      For j := 1 to N do
        if A[i,k]^dd + A[i,k]^dd < A[i,j]^dd then
          Begin
            A[i,j]^dd := A[i,k]^dd + A[i,k]^dd ;
            A[i,j]^đỉnh := k;
          End;
        End;
      End;
    End;
  End;

```

III / Bài tập mẫu :

Bài 1 : Cho đồ thị vô hướng liên thông từ File “DGDI.INP” tổ chức như sau :

+ Dòng thứ nhất ghi 3 số : N,xp,đ (số đỉnh , tên đỉnh xuất phát , đỉnh đích)

+ Các dòng tiếp theo : mỗi dòng 3 số : i,j , A[i,j] (A[i,j] là khoảng cách i tới j)

Nếu i=0 thì kết thúc dữ liệu về đồ thị này

Bằng thuật toán Di jsktra tìm đ- ờng đi ngắn nhất từ xp tới đ

Bài 2 : Nội dung nh- trên nh- ng tìm đ- ờng đi ngắn nhất bằng thuật toán For-Bellman

Lời giải :

Bài 1 : Bằng thuật toán Di jsktra tìm đ- ờng đi ngắn nhất

Uses Crt;

Const Max = 100;

Fi = 'duongdi.inp';

Type Ta = Array[1..Max,1..Max] of Integer;

Re = Record

t : Byte;

h : Word;

End;

```

    Nhan  = Array[0..Max] of Re;
    Dau   = Array[1..Max] of Boolean;
Var    N,xp,d : Byte;
        A      : ^Ta;
        F      : Text;
Procedure DocF;
    Var i,j : Byte;
    Begin
        Assign(F,Fi);
        Reset(F);
        Readln(F,N,xp,d);
        New(A);
        For i:=1 to N do
            For j:=1 to n do A^[i,j] := MaxInt;
        While not Seekeof(F) do
            Begin
                Read(F,i,j);
                If i=0 then
                    Begin Close(F);Exit;End;
                Readln(F,A^[i,j]);
            End;
        For i:=1 to N do A^[i,i] := 0;
        Close(F);
    End;
Procedure Lam;
    Var NH    : Nhan;
        dd    : Dau;
        i,j    : Byte;
    Procedure Khoitao;
        Var i    : Byte;
    Begin
        For i:=1 to N do
            Begin
                NH[i].h := MaxInt;
                DD[i]  := False;
            End;
        NH[xp].h := 0;
        NH[xp].t := 0;

    End;
    Function Min : Byte;
    Var i,k    : Byte;
    Begin
        i := 0;
        For k:=1 to N do
            If (Not DD[k]) and (NH[k].h<NH[i].h) then i := k;
        Min := i;
    End;

```

```

Procedure Sua(i : Byte); { i : dinh cuoi cua hanh trinh hien tai }
Var j      : Byte;
Begin
  DD[i] := True;
  For j:=1 to N do
    If (Not DD[j]) and (NH[j].h>NH[i].h+A^[i,j]) then
      Begin
        NH[j].h := NH[i].h+A^[i,j];
        NH[j].t := i;
      End;
  End;
End;
Procedure Lannguoc;
Var S      : String;
    i,j    : Byte;
Begin
  i := d;
  S := "";
  While i>0 do
    Begin
      S := chr(i)+S;
      i := NH[i].t;
    End;
  For i:=1 to Length(S) do Write(Ord(S[i]), ' ');
End;
Begin
  Clrscr;
  Khoitao;
  While Not DD[d] do
    Begin
      i := Min;
      If i=0 then
        Begin
          Writeln('vo nghiem ');
          Exit;
        End;
      Sua(i);
    End;
  Lannguoc;
End;
BEGIN
  Clrscr;
  DocF;
  Lam;
  Dispose(A);
  Writeln('Da xong ');
  Readln;
END.
Input

```

8 1 8
1 2 3
2 1 3
1 3 5
3 1 5
1 4 2
4 1 2
2 3 1
3 2 1
2 5 7
5 2 7
3 4 4
4 3 4
3 5 5
5 3 5
4 6 3
6 4 3
5 8 3
8 5 3
6 7 4
7 6 4
6 8 6
8 6 6
7 8 5
8 7 5
6 3 1
6 5 2
7 4 6
0

OUT

Nếu $x_p=1, d=8$ thì có đường đi 1 4 6 5 8

Nếu $x_p=8, d=1$ thì có đường đi 8 6 3 2 1

Bài 2 : Bảng thuật toán For-Bellman tìm đường đi ngắn nhất từ x_p tới d

```

Uses Crt;
Const Max = 100;
      Fi = 'Duongdi.inp';
Type Ta = Array[1..Max,1..Max] of Record h : Word;tg : Byte; End;
      Dau = Array[1..Max] of Boolean;
Var N,xp,t : Integer;
    A : ^Ta;
    F : Text;
Procedure DocF;
  Var i,j : Byte;
  Begin
    Assign(F,Fi);
    Reset(F);
    New(A);
    Readln(F,N,xp,t);
    For i:=1 to N do
      For j:=1 to N do
        Begin
          A^[i,j].h := MaxInt;
          A^[i,j].tg := 0;
        End;
    For i:=1 to N do A^[i,i].h := 0;
    While Not SeekEof(F) do
      Begin
        Read(F,i,j);
        If i=0 then
          Begin
            Close(F);
            Exit;
          End;
        Readln(F,A^[i,j].h);
      End;
    Close(F);
  End;
Procedure FB;
  Var i,j,k : Integer;
  Begin
    For k:=1 to N do
      For i:=1 to N do
        For j:=1 to N do
          If (A^[i,k].h+A^[k,j].h<A^[i,j].h) then
            Begin
              A^[i,j].h := A^[i,k].h+A^[k,j].h;
              A^[i,j].tg := k;
            End;
      End;
  End;
Procedure Lannguoc;
  Var S : String;
      i,x1,y1 : Byte;

```

```

Begin
  If A^[xp,t].h = MaxInt then
    Begin
      Writeln('Vo nghiem ');
      Exit;
    End;
  S := Char(xp)+char(t);
  i := 1;
  While i<Length(S) do
    Begin
      x1 := Ord(S[i]);
      y1 := Ord(S[i+1]);
      If A^[x1,y1].tg=0 then Inc(i)
      Else Insert(Char(A^[x1,y1].tg),S,i+1);
    End;
  For i:=1 to Length(S) do Write(Ord(S[i]):4);
  Writeln;
  Writeln('Do dai : ',A^[xp,t].h);
End;
BEGIN
  Clrscr;
  DocF;
  FB;
  Lannguoc;
  Dispose(A);
END.

```

Đồ thị

PHẦN 1 : KHÁI NIỆM CHUNG

I/ Định nghĩa đồ thị :

Đồ thị gồm tập hợp X và một ánh xạ F từ X vào X (ánh xạ này có thể đa trị). Kí hiệu đồ thị là $G(X,F)$.

Thí dụ : Trong mặt phẳng , hình ảnh hình học của đồ thị có thể nh- :

- + Tập X : tập điểm (gọi là tập đỉnh của đồ thị)
- + Ánh xạ F biểu hiện nh- tập cung U (có h-ớng hoặc vô h-ớng)

Cung nối đỉnh x_i với đỉnh x_k kí hiệu là u_{ik} .

Đỉnh x_i gọi là đỉnh gốc , đỉnh x_k gọi là đỉnh ngọn của cung u_{ik} . Cung nối 1 đỉnh với chính đỉnh ấy gọi là cung khuyên .

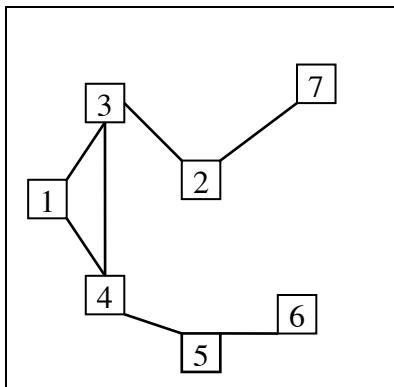
Đỉnh treo là đỉnh chỉ có 1 cung nối với nó , cung này cũng gọi là cung treo

Đỉnh cô lập là đỉnh không có cung nào nối với nó .

Tập hợp các cung của một đồ thị kí hiệu là U , thì đồ thị kí hiệu là $G(X,U)$

Ma trận kề của đồ thị (có N đỉnh) là ma trận $A(N,N)$ đ-ợc tạo nh- sau :

Nếu có s cung nối đỉnh i với đỉnh k thì $A[i,k] = s$ (thông th-ờng $s=1$) . Nếu không có cung nào nối thì $A[i,k]=0$



0	0	1	1	0	0	0
0	0	1	0	0	0	1
1	1	0	1	0	0	0
1	0	1	0	1	0	0
0	0	0	1	0	0	0
0	0	0	0	1	0	0
0	1	0	0	0	0	0

Trong ma trận $A(7,7)$ qui định $A[i,i]=0$ ($i=1..7$)

II/ Phân loại đồ thị :

Cách phân loại theo số cung S nối 2 đỉnh : nếu $S = 0..1$ thì

có đơn đồ thị , nếu $S>1$ có đa đồ thị

Cách phân loại theo cung có h-ớng và vô h-ớng :

- + Trong đồ thị có h-ớng qui định chiều đi trên cung từ đỉnh gốc đến đỉnh ngọn.
- + Trong đồ thị vô h-ớng không phân biệt chiều đi trên cung (nghĩa là không định h-ớng trên cung). Khi đó trong ma trận kề ta có $A[i,k] = A[k,i]$ (số cung từ i tới k cũng là số cung từ k tới i). Đồ thị vô h-ớng còn gọi là đồ thị đối xứng . Cung trong đồ thị đối xứng đ-ợc gọi là cạnh của đồ thị

III/ Một số định nghĩa khác :

a) Trong đồ thị có h-ớng :

+ Tổng số cung đi vào một đỉnh gọi là bán bậc vào của đỉnh . Tổng số cung đi ra từ một đỉnh gọi là bán bậc ra của đỉnh .

+ Một dãy cung liên tiếp (có thể không cùng chiều) gọi là một dây chuyền.

- + Một dây chuyền mà ngọn của cung này là gốc của cung tiếp theo (trừ cung cuối cùng) được gọi là một mạch (còn gọi là đường đi có hướng)
- + Một mạch khép kín (ngọn cung cuối cùng trùng với gốc cung đầu tiên) gọi là mạch đóng (còn gọi là chu trình có hướng)
- + Chu trình sơ cấp là chu trình đi qua các đỉnh của nó không quá 1 lần (trừ đỉnh đầu và đỉnh cuối)
- + Độ dài của mạch là tổng khoảng cách các cung của nó (trong một số trường hợp ng-ời ta coi mỗi cung dài bằng 1 thì độ dài của mạch là số lượng cung trên mạch)
- + Hai đỉnh được gọi là liên thông nếu tồn tại ít nhất 1 dây chuyền nối chúng. Hai đỉnh được gọi là liên thông mạnh nếu tồn tại ít nhất 1 mạch nối chúng. Một vùng liên thông của đồ thị là tập hợp một số đỉnh của đồ thị mà 2 đỉnh bất kỳ trong chúng liên thông nhau. Một vùng liên thông mạnh của đồ thị là tập hợp một số đỉnh của đồ thị mà 2 đỉnh bất kỳ trong chúng liên thông mạnh với nhau.
- Một đồ thị được gọi là đồ thị liên thông nếu nó chỉ gồm 1 vùng liên thông duy nhất, một đồ thị được gọi là đồ thị liên thông mạnh nếu nó chỉ gồm 1 vùng liên thông mạnh duy nhất.
- Ta cũng có các định nghĩa tương tự cho đồ thị vô hướng :

b) Trong đồ thị vô hướng :

- + Tổng số cạnh nối tới một đỉnh gọi là bậc của đỉnh.
- + Một dãy cạnh và đỉnh liên tiếp gọi là một đường đi
- + Một đường đi khép kín gọi là một chu trình
- + Chu trình sơ cấp là chu trình đi qua các đỉnh của nó không quá 1 lần (trừ đỉnh đầu và đỉnh cuối)
- + Độ dài của đường đi là tổng khoảng cách các cạnh của nó (trong một số trường hợp ng-ời ta coi mỗi cạnh dài bằng 1 thì độ dài của đường đi là số lượng cạnh trên đường đi)
- + Hai đỉnh được gọi là liên thông nếu tồn tại ít nhất 1 đường đi nối chúng. Một vùng liên thông của đồ thị là tập hợp một số đỉnh của đồ thị mà 2 đỉnh bất kỳ trong chúng liên thông nhau.
- Một đồ thị được gọi là đồ thị liên thông nếu nó chỉ gồm 1 vùng liên thông duy nhất.
- + Cầu của đồ thị là cạnh có tính chất : nếu xóa nó khỏi đồ thị thì số vùng liên thông của đồ thị tăng thêm 1 vùng
- c) Đường đi và chu trình đặc biệt :

- + Đường đi qua tất cả các đỉnh, mỗi đỉnh qua đúng 1 lần, gọi là đường đi Haminton. Chu trình đi qua tất cả các đỉnh, mỗi đỉnh qua đúng 1 lần, gọi là chu trình Haminton.
- + Đường đi qua tất cả các cạnh, mỗi cạnh qua đúng 1 lần, gọi là đường đi Euler. Chu trình đi qua tất cả các cạnh, mỗi cạnh qua đúng 1 lần, gọi là chu trình Euler.

IV / Một vài tính chất khác trong đồ thị vô hướng:

- 1) Nếu đồ thị vô hướng, liên thông và không có chu trình thì khi xóa 1 cạnh sẽ mất tính liên thông.
- 2) Ngược lại : một đồ thị vô hướng, liên thông khi xóa 1 cạnh mà mất tính chất liên thông thì đồ thị đó không có chu trình
- 3) Điều kiện cần và đủ để đồ thị có chu trình Euler là bậc của mọi đỉnh đều chẵn
- 4) Điều kiện cần và đủ để đồ thị có đường đi Euler: số đỉnh bậc lẻ không lớn hơn 2
- 5) Hệ thức Euler :

$$CT = SC - SD + SVLT$$

CT : số chu trình Sc : số cạnh
Sđ : số đỉnh Svlt : số vùng liên thông.

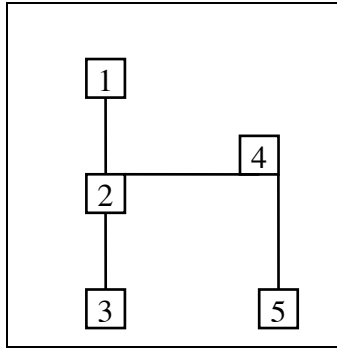
Thí dụ :

Đồ thị

4 cạnh, 5

Do đó số

CT = 4 -



bên có :

đỉnh, 1 vùng liên thông

chu trình là :

$5 + 1 = 0$ (Không có chu trình)

V / Số ổn định trong và số ổn định ngoài :

1) Số ổn định trong :

+ Tập con A các đỉnh thuộc đồ thị $G(X,E)$ là tập ổn định trong nếu mỗi cặp đỉnh thuộc A đều không kề nhau

+ Tập ổn định trong lớn nhất : Là tập ổn định trong và nếu thêm một đỉnh tùy ý thì không còn là tập ổn định trong .

+ Số phần tử của tập ổn định trong lớn nhất gọi là số ổn định trong . Ký hiệu là $\alpha(G)$

2) Số ổn định ngoài :

+ Tập đỉnh B thuộc đồ thị $G(X,E)$ gọi là tập ổn định ngoài nếu với mọi đỉnh y của đồ thị không thuộc B thì đều tìm thấy một đỉnh x thuộc B mà x và y có cạnh nối .

+ Tập ổn định ngoài nhỏ nhất là tập ổn định ngoài có số phần tử ít nhất .

+ Số phần tử của tập ổn định ngoài nhỏ nhất đ-ợc gọi là số ổn định ngoài . Ký hiệu là $\beta(G)$

3) Một số tính chất :

+ Mọi tập con của tập ổn định trong cũng là tập ổn định trong .

+ Mọi tập đỉnh của đồ thị chứa tập ổn định ngoài cũng là tập ổn định ngoài .

4) Nhân đồ thị :

+ Nhân đồ thị là tập đỉnh của đồ thị có tính chất : vừa là tập ổn định trong vừa là tập ổn định ngoài

VI / Sắc số của đồ thị :

+ Sắc số của đồ thị là số màu ít nhất có thể tô các đỉnh đồ thị sao cho 2 đỉnh kề nhau tùy ý khác màu .

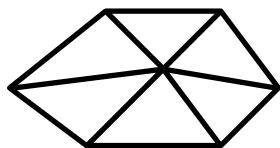
+ Một số định lý về sắc số :

ĐL1 : Đồ thị đầy đủ n đỉnh có sắc số bằng n

ĐL2 : Một chu trình có độ dài chẵn luôn có sắc số = 2

ĐL3 : Một chu trình có độ dài lẻ luôn có sắc số = 3

ĐL4 : Đồ thị hình hoa thị gồm 1 chu trình và 1 đỉnh A nối với các đỉnh của chu trình (hình vẽ) có sắc số = 3 nếu chu trình chẵn , có sắc số = 4 nếu chu trình lẻ



+ Thuật toán tìm sắc số :

Thuật toán 1 : Bằng cách áp dụng các định lý trên , ta tìm đ-ợc khẳng định về số màu tô ít nhất là p . Vậy sắc số $\geq p$. Sau đó chỉ ra đ-ợc 1 cách tô chỉ bằng p màu . Từ đó kết luận đ-ợc sắc số = p .

Thuật toán 2 : (Tìm đ-ợc gần đúng)

+ Các đỉnh ch- a đánh dấu

- + Tính bậc các đỉnh
- + Sắp các đỉnh theo thứ tự bậc giảm dần
- + Tô đỉnh có bậc cao nhất và những đỉnh không kề với đỉnh này và ch- a bị đánh dấu bằng cùng màu 1
- + Đánh dấu các đỉnh đã đ- ọc tô màu.
- + Lại chọn đỉnh có bậc cao nhất , tô đỉnh có bậc cao nhất và những đỉnh không kề với đỉnh này và ch- a bị đánh dấu bằng cùng màu mới (giả sử đã dùng các màu từ 1 đến i thì bây giờ tô màu i+1)
- + Quá trình nh- thể cho đến khi các đỉnh đều đã đ- ọc đánh dấu

BÀI TẬP

1) Cho ma trận kề $A(N,N)$ của đồ thị N đỉnh . Tìm số vùng liên thông của đồ thị .

Yêu cầu : File input : 'SVLT.txt'

- + Dòng đầu : N
 - + N dòng tiếp theo : Ma trận $A(N,N)$
- Dữ liệu ra trên File 'SVLT.out'
- + Dòng đầu : số S là số vùng liên thông
 - + S dòng tiếp theo : Mỗi dòng ghi các đỉnh thuộc cùng 1 vùng liên thông

2) Cho hình chữ nhật $H(M,N)$ m dòng , n cột gồm $M \times N$ ô vuông , mỗi ô vuông chứa số 0 hoặc 1. Tìm và tính diện tích các vùng liên thông chứa toàn số 0 trong 2 tr- ờng hợp :

- + Các ô số 0 nếu chung cạnh thì có đ- ờng đi tới nhau
- + Các ô số 0 nếu có điểm chung thì có đ- ờng đi tới nhau

Yêu cầu :

File input 'HCN.txt'

Dòng đầu : 2 số M,N

M dòng tiếp theo : ma trận thể hiện hình chữ nhật $H(M,N)$

File output 'HCN.out'

Mỗi tr- ờng hợp thể hiện một ma trận hình chữ nhật $D(M,N)$ sao cho các ô của D cùng thuộc 1 vùng liên thông thì có cùng 1 mã số vùng . Những ô số 1 trong H thay bằng ô t- ờng ứng trong D là kí tự '*'

Dòng cuối cùng là diện tích của các vùng .

3) Đề thi Quốc tế 1994 (tại Thụy Điển) : Bài 2 (5-7-1994)

Hình 2 biểu diễn bản đồ lâu đài . Hãy viết ch- ơng trình tính :

- 1 - Lâu đài có bao nhiêu phòng ?
- 2 - Phòng lớn nhất là bao nhiêu ?
- 3 - Bức t- ờng nào cần loại bỏ để phòng càng rộng càng tốt ?

Lâu đài chia thành $M \times N$ ($M \leq 50, N \leq 50$) modul vuông . Mỗi modul vuông có thể có từ 0 đến 4 bức t- ờng

INPUT DATA

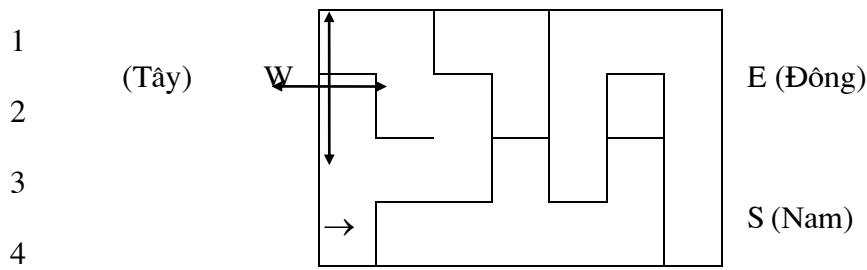
Bản đồ đ- ọc l- u trữ trong file Input.txt ở dạng các số cho các modul .

File bắt đầu từ số l- ợng các modul theo h- ớng Bắc-Nam và số l- ợng các modul theo h- ớng Đông-Tây.

Trong các dòng tiếp theo , mỗi modul đ- ọc mô tả bởi 1 số ($0 \leq p \leq 15$). Số đó là tổng của : 1 (= t- ờng phía Tây), 2 (= t- ờng phía Bắc), 4 (= t- ờng phía Đông) , 8 (= t- ờng phía Nam) .

1 2 3 4 5 6 7

N (Bắc)



Mũi tên chỉ bức tường cần loại bỏ theo kết quả ở ví dụ

Các bức tường ở bên trong đọc xác định hai lần ; bức tường phía Nam trong modul (1,1) đồng thời là bức tường phía Bắc trong modul (2,1)

* Lâu đài luôn có ít nhất 2 phòng

INPUT.TXT của ví dụ :

```
4
7
11 6 11 6 3 10 6
7 9 6 13 5 15 5
1 10 12 7 13 7 5
13 11 10 8 10 12 13
```

Output data

Trong file ra OUTPUT.TXT viết trên 3 dòng : dòng thứ nhất viết số lượng phòng ,dòng tiếp đến là diện tích của phòng lớn nhất (tính theo số modul) và bức tường cần loại bỏ (trước tiên là hàng sau đó là cột của modul có tường đó) và dòng cuối cùng là hướng của bức tường .Trong ví dụ “4 1 E “ là một trong số các khả năng có thể ,bạn chỉ cần chỉ ra một)

```
5
9
4 1 E
```

4) Một vùng lãnh thổ có dạng một lưới ô vuông A gồm $N \times N$ ô ($4 \leq N \leq 12$) với mục đích phủ sóng truyền hình toàn vùng lãnh thổ ,ngay-ời ta lập một dự án xây dựng một hệ thống gồm k trạm tiếp sóng ở k ô của lưới .Một trạm tiếp sóng đặt ở một ô nào đó của lưới không những bảo đảm phủ sóng ô này mà còn cho tất cả các ô có chung đỉnh với nó .Dữ liệu về dự án đọc cho trong 1 File dạng Text là PHUSONG.TXT trong đó dòng đầu tiên ghi số N ,trong k dòng tiếp theo , mỗi dòng ghi 2 số nguyên d-ong (x_i, y_i) là tọa độ trên lưới của một trạm tiếp sóng của dự án (hai số cách nhau bởi dấu cách).Dữ liệu ra ghi trong File PHUSONG.OUT :

a) N dòng đầu là ma trận A(N,N) (các trạm tiếp sóng ghi số 1,ô khác ghi số 0)

b) Dòng tiếp theo là số 0 hoặc số 1 : Số 1 là dự án phủ sóng toàn lãnh thổ,số 0 là dự án không phủ đ-ợc toàn lãnh thổ

Trong trường hợp dự án không phủ toàn lãnh thổ , dòng tiếp theo là số S : số các ô ch-a đ-ợc phủ sóng , sau đó S dòng tiếp theo lần lượt mỗi dòng ghi tọa độ của một ô ch-a đ-ợc phủ sóng .

c) Trong trường hợp phủ sóng toàn lãnh thổ,hãy tìm cách loại bớt 1 số trạm tiếp sóng mà vẫn phủ sóng toàn lãnh thổ ,nếu không loại bỏ đ-ợc thì ghi số 0 ,nếu loại bỏ đ-ợc thì ghi số trạm loại bỏ nhiều nhất ,sau đó nêu rõ tọa độ các trạm bị loại bỏ (mỗi trạm 1 dòng)

Trong File PHUSONG.OUT , để ngăn cách kết quả từng câu , trước kết quả câu a) là dòng chữ “ CAU A” ; trước kết quả câu b) là dòng chữ “ CAU B” ; trước kết quả câu c) là dòng chữ “ CAU C”

5) Bài kiểm tra :

Cho đồ thị G vô hướng gồm N đỉnh , biểu diễn bởi ma trận A : $A[i,j]=A[j,i]=0$ hoặc 1(0 là không có đ-ờng nối i với j , 1 là ngược lại).Đồ thị gọi là liên thông đơn nếu với mọi i,j bất kỳ có đúng 1 đ-ờng đi nối i với j .

- a) Kiểm tra A có liên thông đơn không. Nếu không thì loại bớt một số cạnh để liên thông đơn.
b) Giả sử G liên thông đơn, hãy tìm các cạnh độc đạo (là cạnh mà mọi đường đi dài nhất đều qua nó)

6) Cho đồ thị $G(X, E)$. Lập chương trình tìm số ổn định trong, số ổn định ngoài, tìm tập nhân ít phần tử nhất.

7) Cho N điểm, hãy dùng số màu ít nhất tô màu các điểm sao cho 2 điểm kề nhau thì khác màu nhau.

8) Đề thi Tin học Toàn quốc 3-1998 : Dàn đèn màu

Cho một lưới tọa độ nguyên, hoành độ từ 0 đến M, tung độ từ 0 đến N ($M, N \leq 200$). Trên k nút cho trước, mỗi nút cần đặt một đèn màu sao cho 2 đèn ở 2 nút có cùng hoành độ hoặc có cùng tung độ phải có màu khác nhau. Hãy tìm cách bố trí đèn sao cho số màu phải dùng là ít nhất. Các màu đã sử dụng phải được đánh số bởi các số nguyên dương liên tục bắt đầu từ số 1

Dữ liệu vào : File BL1.INP

* Dòng đầu ghi 3 số M, N, K

* Dòng thứ i trong số k dòng tiếp theo ghi hoành độ và tung độ của nút thứ i trong dãy k nút cần đặt đèn ($i = 1, 2, \dots, k$)

Kết quả : Ghi vào File BL1.OUT

* Dòng đầu ghi số lượng màu cần sử dụng p

* Dòng thứ i trong số k dòng tiếp theo ghi màu của đèn ở nút thứ i ($i = 1, 2, \dots, k$)

Ví dụ

BL1.INP

```
4 5 13
1 1
1 2
1 5
3 1
4 1
3 2
2 3
3 3
4 3
2 4
4 4
2 5
4 5
```

BL1.OUT

```
4
1
2
3
2
3
1
1
3
2
3
1
2
4
```

PHẦN BÀI CHÈA

Bài 1 (Tìm số vùng liên thông)

```

Uses  Crt;
Const Max  = 100;
      Fi    = 'Lthong.txt';
      Fo    = 'Lthong.out';
Type  MA    = Array[1..Max,1..Max] of 0..1;
      MD    = Array[1..Max] of Byte;
      MQ    = Array[1..Max*Max] of Byte;
Var    A     : MA;
      D     : MD;
      Q     : MQ;
      N,dau,cuoi,sv : Byte;
Procedure DocF;
  Var    F      : Text;
        i,j    : Byte;
  Begin
    Assign(F,Fi);
    Reset(F);
    Readln(F,N);
    For i:=1 to N do
      Begin
        For j:=1 to N do Read(F,A[i,j]);
        Readln(F);
      End;
    Close(F);
  End;
Function Tim : Byte;
  Var    i      : Byte;
  Begin
    Tim := 0;
    For i:=1 to N do
      If D[i]=0 then
        Begin
          Tim := i;
          Exit;
        End;
    End;
  End;

Procedure TaoQ_rong;
  Begin
    FillChar(Q,sizeof(Q),0);
    Dau := 0;
    Cuoi := 0;
  End;
Procedure Loang(i : Byte);
  Var    j,k    : Byte;
  Begin
    Inc(cuoi);
    Q[cuoi] := i;
    D[i]    := sv;
    While (dau+1<=cuoi) do
      Begin

```

```

    Inc(dau);
    j := Q[dau];
    For k:=1 to N do
    If (D[k]=0) and (A[j,k]=1) then
    Begin
        Inc(cuoi);
        Q[cuoi] := k;
        D[k] := sv;
    End;
    End;
End;
Procedure Timstplt;
Var      i      : Byte;
        Ok      : Boolean;
Begin
    sv := 0;
    FillChar(D,sizeof(D),0);
    Repeat
        TaoQ_rong;
        Ok := True;
        i := Tim;
        If i>0 then
        Begin
            Inc(sv);
            Loang(i);
            Ok := False;
        End;
    Until Ok;
    Writeln('So thanh phan lien thong : ',sv);
End;

Procedure GhiF;
Var      F      : Text;
        i,j      : Byte;
Begin
    Assign(F,Fo);
    Rewrite(F);
    Writeln(F,'So thanh phan lien thong la : ',sv);
    For i:=1 to sv do
    Begin
        Write(F,'Vung ',i,' : ');
        For j:=1 to N do
            If D[j]=i then Write(F,j:4);
        Writeln(F);
    End;
    Close(F);
End;
BEGIN
    Clrscr;
    DocF;
    Timstplt;
    GhiF;
END.

```

SVLT.TXT

```

11
0 1 0 0 0 0 0 0 0 0 0
1 0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 0 0 0 0 0 0
0 0 0 0 1 0 0 0 0 0 0
0 0 1 1 0 1 0 0 0 0 0
0 0 0 0 1 0 1 1 0 0 0
0 0 0 0 0 1 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 1
0 0 0 0 0 0 0 0 0 1 0

```

SVLT.OUT

So thanh phan lien thong la : 4

Vung 1 : 1 2

Vung 2 : 3 4 5 6 7 8

Vung 3 : 9

Vung 4 : 10 11

Bài 2 (Tìm số vùng liên thông của các ô số 0 trong hình chữ nhật theo 2 cách : chung cạnh, chung đỉnh)

```

Uses Crt;
Const Max = 100;
      Fi   = 'SVLT2.txt';
      Fo   = 'SVLT2.out';
      aDc  : Array[1..4] of -1..1 = ( 0 , 1 , 0 , -1); { so gia cot }
      aDd  : Array[1..4] of -1..1 = (-1, 0 , 1 , 0); { so gia dong }
      bDc  : Array[1..8] of -1..1 = ( 0, 1, 1, 1, 0, -1, -1, -1); { so gia cot }
      bDd  : Array[1..8] of -1..1 = (-1, -1, 0, 1, 1, 1, 0, -1); { so gia dong }
Type KA = Array[1..Max, 1..Max] of 0..1;
      KD = Array[1..Max, 1..Max] of Byte;
      KQ = Array[1..Max*Max] of Record d,c : Byte; End;
      KDT = Array[1..Max*Max] of Integer;
Var A : KA;
     D : KD;
     Q : KQ;
     DT : KDT;
     N, M, i, j, dau, cuoi, sv, cau : Byte;

```

Procedure DocF;

Var i, j : Byte; F : Text;

Begin

Assign(F, Fi);

Reset(F);

Readln(F, M, N);

For i:=1 to M do

Begin

For j:=1 to N do Read(F, A[i, j]);


```

    Readln(F);
End;
Close(F);
End;

```

```

Function Tim(Var i,j : Byte): Boolean;
Var      x,y : Byte;
Begin
    Tim := False;
    For x:=1 to M do
    For y:=1 to N do
    If (D[x,y]=0) and (A[x,y]=0) then
    Begin
        i := x;
        j := y;
        Tim := True;
        Exit;
    End;
End;

```

```

Procedure Q_rong;
Begin
    Fillchar(Q,Sizeof(D),0);
    Dau := 0;
    Cuoi := 0;
End;

```

```

Procedure Loang1(i,j : Byte);
Var      k,dong,cot,u,v : byte;
Begin
    Inc(cuoi);
    Q[cuoi].d := i;
    Q[cuoi].c := j;
    D[i,j] := sv;
    While dau+1<=cuoi do
    Begin
        Inc(dau);
        dong := Q[dau].d;
        cot := Q[dau].c;
        For k:=1 to 4 do
        Begin
            u := dong + aDd[k];
            v := cot + aDc[k];
            If (u>0) and (u<=M) and (v>0) and (v<=N) then
            If (A[u,v]=0) and (D[u,v]=0) then
            Begin
                Inc(cuoi);
                Q[cuoi].d := u;
                Q[cuoi].c := v;
                D[u,v] := sv;
                Inc(DT[sv]);
            End;
        End;
    End;
End;

```

```

Procedure Loang2(i,j : Byte);
  Var      k,dong,cot,u,v : byte;
  Begin
    Inc(cuoi);
    Q[cuoi].d := i;
    Q[cuoi].c := j;
    D[i,j] := sv;
    While dau+1<=cuoi do
      Begin
        Inc(dau);
        dong := Q[dau].d;
        cot := Q[dau].c;
        For k:=1 to 8 do
          Begin
            u := dong + bDd[k];
            v := cot + bDc[k];
            If (u>0) and (u<=M) and (v>0) and (v<=N) then
              If (A[u,v]=0) and (D[u,v]=0) then
                Begin
                  Inc(cuoi);
                  Q[cuoi].d := u;
                  Q[cuoi].c := v;
                  D[u,v] := sv;
                  Inc(DT[sv]);
                End;
              End;
            End;
          End;
        End;
      End;
    End;
  End;

```

```

Procedure Timsvlt(cau : Byte);
  Var      Ok : Boolean;
  Begin
    Sv := 0;
    For i:=1 to M*N do DT[i] := 1;
    Fillchar(D,sizeof(D),0);
    Repeat
      Ok := True;
      Q_rong;
      If Tim(i,j) then
        Begin
          Inc(sv);
          If cau=1 then
            Loang1(i,j) Else Loang2(i,j);
          Ok := False;
        End;
      Until Ok;
    End;
  End;

```

```

Procedure HienBandoV;
  Var      i,j : Byte; F : Text;
  Begin
    Assign(F,Fo);
    Rewrite(F);
    For i:=1 to M do
      Begin

```

```

    For j:=1 to N do
        If D[i,j]=0 then Write(F,'*':4)
        Else Write(F,D[i,j]:4);
    Writeln(F);
End;
Writeln(F,'Dien tích tung vung : ');
For i:=1 to sv do Write(F,DT[i]:4);
Close(F);
End;
Procedure Menu;
Var      ch : Char;
Begin
    Writeln('Go ESC thoat ! ');
    Writeln('Chon cau A hay B (A/B) ');
    Repeat
        Ch := Upcase(Readkey);
        If ch=#27 then Exit;
        If ch='A' then cau:=1 Else cau:=2;
        Timsvlt(cau);
        HienBandoV;
    Until ch in ['A'..'B',#27]

End;
BEGIN
    Clrscr;
    DocF;
    Menu;
    Writeln('Da xong . Moi go Enter va Xem du lieu ra trong File ',Fo);
    Readln;
END.

```

Dữ liệu vào trong File SVLT2.TXT

```

8 10
0 1 0 0 0 0 0 0 1 0
1 1 0 0 0 0 0 0 1 0
0 0 0 1 1 0 0 0 1 0
1 1 1 0 1 1 0 0 1 0
0 0 1 1 0 0 0 0 1 0
0 0 0 1 1 1 1 1 1 0
1 1 0 1 0 0 0 1 0 1
0 0 0 1 0 0 1 0 1 0

```

Kết quả câu a) trong SVLT2.OUT

```

1 * 2 2 2 2 2 2 * 3
* * 2 2 2 2 2 2 * 3
2 2 2 * * 2 2 2 * 3
* * * 4 * * 2 2 * 3
5 5 * * 2 2 2 2 * 3
5 5 5 * * * * * 3
* * 5 * 6 6 6 * 7 *
5 5 5 * 6 6 * 8 * 9
Dien tích tung vung :

```

1 24 6 1 9 5 1 1 1

Kết quả câu b) trong SVLT2.OUT

```

1 * 2 2 2 2 2 2 * 3
* * 2 2 2 2 2 2 * 3
2 2 2 * * 2 2 2 * 3
* * * 2 * * 2 2 * 3
4 4 * * 2 2 2 2 * 3
4 4 4 * * * * * 3
* * 4 * 3 3 3 * 3 *
4 4 4 * 3 3 * 3 * 3

```

Diện tích từng vùng :

1 25 14 9

Bài 3 :

```

Uses  Crt;
Const  MM  = 50;
       MN  = 50;
       Fi  = 'Input.txt';
       Fo  = 'Output.txt';
Type   KA  = Array[1..MM,1..MN] of Byte;
       KDT = Array[1..MM*MN] of Integer;
       KDD = Array[0..MM+1,0..MN+1] of Integer;
       Kpt = Record x,y : Byte; End;
       KQ  = Array[1..MM*MN] of KPT;
Var     A   : KA;
       DT   : KDT;
       Q    : KQ;
       D    : KDD;
       ch   : Char;
       M,N,x,y : Byte;
       dau,cuoi,sp,dtm : Integer;
Procedure DocF;
  Var    F      : Text;
         i,j    : Byte;
  Begin
    Assign(F,Fi);
    Reset(f);
    Readln(F,M);
    Readln(F,N);
    For i:=1 to M do
      Begin
        For j:=1 to N do Read(F,A[i,j]);
        Readln(F);
      End;
    Close(F);
  End;
Procedure Q_rong;
  Begin
    Fillchar(Q,sizeof(Q),0);
    dau := 0;
    cuoi := 0;

```

```

End;
Procedure Lay(var x,y : Byte);
Begin
    Inc(dau);
    x := Q[dau].x;
    y := Q[dau].y;
End;
Procedure Nap(x,y : Byte);
Begin
    Inc(cuoi);
    Q[cuoi].x := x;
    Q[cuoi].y := y;
    D[x,y] := sp;
    Inc(DT[sp]);
End;
Procedure Loang(x,y : Byte); { o(x,y) dau tien cua 1 phong moi }
Var    i,j    : Byte;
Begin
    Nap(x,y);
    While (dau+1<=cuoi) do
        Begin
            Lay(x,y);
            If (A[x,y] and 1 = 0) and (D[x,y-1]=0) then Nap(x,y-1);
            If (A[x,y] and 2 = 0) and (D[x-1,y]=0) then Nap(x-1,y);
            If (A[x,y] and 4 = 0) and (D[x,y+1]=0) then Nap(x,y+1);
            If (A[x,y] and 8 = 0) and (D[x+1,y]=0) then Nap(x+1,y);
        End;
    End;
End;

Function Tim(Var x,y : Byte) : Boolean;
Var    i,j    : Byte;
Begin
    Tim := False;
    For i:=1 to M do
        For j:=1 to N do
            If D[i,j]=0 then
                Begin
                    x:=i;
                    y:=j;
                    Tim:=true;
                    Exit;
                End;
        End;
    End;
End;
Procedure Timsophong;
Var    i,j    : Byte;
        Ok    : Boolean;
Begin
    For i:=0 to M+1 do
        For j:=0 to N+1 do D[i,j] := -1;
    For i:=1 to M do
        For j:=1 to N do D[i,j] := 0;
    sp := 0;
    Repeat
        Ok := True;
        If Tim(x,y) then

```

```

Begin
    Q_rong;
    Inc(sp);
    Loang(x,y);
    Ok := False;
End;
Until Ok;
End;
Procedure Dientich_Max;
Var    i      : Integer;
Begin
    DtM := DT[1];
    For i:=2 to sp do
        If DT[i]>dtm then dtm := DT[i];
    End;
Procedure PhaPhong(Var x,y : Byte; Var ch : Char);
Var    i,j    : Byte;
        phu   : Integer;
Begin
    phu := 0;
    For i:=1 to M-1 do
        For j:=1 to N-1 do
            Begin
                If (D[i,j]<>D[i+1,j]) and (DT[D[i,j]]+DT[D[i+1,j]]>phu) then
                    Begin
                        x := i;
                        y := j;
                        ch := 'S';
                        phu := DT[D[i,j]]+DT[D[i+1,j]];
                    End;
                If (D[i,j]<>D[i,j+1]) and (DT[D[i,j]]+DT[D[i,j+1]]>phu) then
                    Begin
                        x := i;
                        y := j;
                        ch := 'E';
                        phu := DT[D[i,j]]+DT[D[i,j+1]];
                    End;
            End;
        End;
    End;
End;
Procedure Lam_GhiF;
Var    F : Text;
Begin
    Assign(F,Fo);
    Rewrite(F);
    Timsophong;
    Writeln(F,sp);
    Dientich_Max;
    Writeln(F,dtm);
    Phaphong(x,y,ch);
    Writeln(F,x,y:3,ch:3);
    Close(F);
End;
BEGIN
    Clrscr;
    DocF;

```

Lam_GhiF;

END.

INPUT.TXT

5

10

3 10 10 2 10 10 2 10 10 6

1 6 3 4 3 6 1 6 3 4

1 4 1 4 1 4 1 4 1 4

1 12 9 4 9 12 1 12 9 4

9 10 10 8 10 10 8 10 10 12

OUTPUT.TXT

2

44

1 5 S

1 6 S

2 4 E

2 5 N

2 5 W

2 6 E

2 6 N

2 7 W

3 4 E

3 5 W

3 6 E

3 7 W

4 4 E

4 5 S

4 5 W

4 6 E

4 6 S

4 7 W

5 5 N

5 6 N

Bài 4 : (Phủ sóng)

Uses Crt;

Const MN = 12;

Fi = 'Phusong.txt';

Fo = 'Phusong.out';

Di : Array[1..8] of -1..1 = (-1,-1, 0, 1, 1, 1, 0,-1);

Dj : Array[1..8] of -1..1 = (0, 1, 1, 1, 0,-1,-1,-1);

Type Ka = Array[1..Mn,1..Mn] of 0..1;

Kpt = Record x,y : Byte; End;

KTram = Array[1..Mn*Mn] of Kpt;

Kddau = Array[1..Mn,1..Mn] of Byte;

Kketqua = Array[0..Mn*Mn] of Byte;

Var A,B : Ka;

T,CP : Ktram;

D : Kddau;

KQ,LKq : Kketqua;

N,st,Luu_bo : Byte;

F2 : Text;

Phutatca : Boolean;

```

Dabo      : Array[1..Mn*Mn] of Boolean;
Procedure DocF;
  Var F : Text;
  Begin
    Assign(F,Fi);
    Reset(F);
    Readln(F,N);
    st := 0;
    While not eof(F) do
      Begin
        Inc(st);
        Readln(F,T[st].x,T[st].y);
      End;
    Close(F);
  End;
Procedure Hien(X : KA);
  Var i,j : Byte;
  Begin
    For i:=1 to N do
      Begin
        For j:=1 to N do Write(F2,A[i,j]:2);
        Writeln(F2);
      End;
    End;
Procedure MoF_out;
  Begin
    Assign(F2,Fo);
    ReWrite(F2);
  End;
Procedure CauA;
  Var i : Byte;
  Begin
    Writeln(F2,'CAU A');
    Fillchar(A,sizeof(A),0);
    For i:=1 to st do A[T[i].x,T[i].y] := 1;
    Hien(A);
  End;
Procedure CauB;
  Var i,j,k : Byte;
  Begin
    PHUTATCA := False;
    Writeln(F2,'CAU B');
    B := A;
    For i:=1 to N do
      For j:=1 to N do
        If B[i,j]=1 then
          For k:=1 to 8 do
            If (i+Di[k]>0) and (j+Dj[k]>0)
              and (i+Di[k]<=N) and (j+Dj[k]<=N) then
              Inc(A[i+Di[k],j+Dj[k]]);
    k := 0;
    For i:=1 to N do
      For j:=1 to N do
        If A[i,j]=0 then
          Begin

```



```

        Inc(k);
        CP[k].x := i;
        CP[k].y := j;
    End;
If k=0 then
Begin
    Writeln(F2,1);
    PHUTATCA := True;
End
Else
Begin
    Writeln(F2,0); { Nhung o chua duoc phu song }
    For i:=1 to k do Writeln(F2,CP[i].x:3,CP[i].y:3);
End;
End;
Procedure Giam(i : Byte);
Var k : Byte;
Begin
    Dec(A[T[i].x,T[i].y]);
    For k:=1 to 8 do Dec(A[T[i].x+Di[k],T[i].y+Dj[k]]);
End;
Procedure Tang(i : Byte);
Var k : Byte;
Begin
    Inc(A[T[i].x,T[i].y]);
    For k:=1 to 8 do Inc(A[T[i].x+Di[k],T[i].y+Dj[k]]);
End;
Function Boduoc(i : Byte) : Boolean;
Var k,u,v : Byte;
Begin
    Boduoc := True;
    If A[T[i].x,T[i].y]<=1 then
    Begin
        Boduoc := False;
        Exit;
    End;
    For k:=1 to 8 do
    Begin
        u := T[i].x+Di[k];
        v := T[i].y+Dj[k];
        If (A[u,v]<=1) and (u>0) and (u<=N) and (v>0) and (v<=N)
        then
        Begin
            Boduoc := False;
            Exit;
        End;
    End;
End;
Procedure Try(k : Byte);{ So tram loai bo la k }
Var i : Byte;
    Ok : Boolean;
Begin
    Ok := False;
    For i := 1 to ST do
        If Boduoc(i) and (Not Dabo[i]) then

```

```

Begin
  Giam(i);
  KQ[k]:= i;
  Dabo[i] := True;
  Ok := True;
  Try(k+1);
  Tang(i);
  Dabo[i] := False;
End;
If Not Ok then
If k-1>luu_bo then
Begin
  LKQ  := KQ;
  Luu_bo := k-1;
End;
End;
Procedure CauC;
Var i : Byte;

Begin
  Fillchar(Dabo,Sizeof(Dabo),False);
  Writeln(F2,'CAU C');
  If Not phutatca then
  Begin
    Writeln(F2,0,' khong phu duoc tat ca ');
  End
  Else
  Begin
    Luu_bo := 0;
    KQ[0] := 0;
    Try(1);
    If Luu_bo=0 then Writeln(F2,0)
    Else
    Begin
      Writeln(F2,Luu_bo);
      For i:=1 to Luu_bo do
        Writeln(F2,T[LKQ[i]].x:3,T[LKQ[i]].y:3);
    End;
  End;
End;
BEGIN
  DocF;
  MoF_out;
  CauA;
  CauB;
  CauC;
  Close(F2);
END.

PHUSONG.TXT
5
1      1
5      5
2      2
2      4

```

3 1
3 4
5 2
5 4

PHUSONG.OUT

CAU A

1 0 0 0 0
0 1 0 1 0
1 0 0 1 0
0 0 0 0 0
0 1 0 1 1

CAU B

1

CAU C

4

1 1
5 5
3 1
3 4

Bài 6: (Số ổn định trong, ổn định ngoài , tập nhân)

Uses Crt;

Const Max = 100;

Fi = 'OnDinh2.inp';

Fo = 'OnDinh2.out';

Type Mang1 = Array[0..Max] of Integer;

Var A : Mang1;

N,k : Byte;

F,F2 : Text;

G : Array[1..Max,1..Max] of Integer;

Dem,Tong : LongInt;

Procedure DocF;

Var i,j : Byte;

Begin

Assign(F,Fi);

Reset(F);

Readln(F,N);

While Not Eof(F) do

Begin

Read(F,i);

While Not Eoln(F) do

Begin

Read(F,j);

G[i,j] := 1;

G[j,i] := 1;

End;

Readln(F);

End;

Close(F);

End;

Procedure Hien;

Var i : Byte;

Begin

```

    Inc(dem);
    For i:=1 to k do
        Write(F2,A[i]:4);
    Writeln(F2);
End;

```

```

Procedure Tao_Trong(i : Byte);
Var
    j : Byte;
Function KT_Trong (A : Mang1;h : Byte): Boolean;
Var
    x,y : Byte;
Begin
    For x:=1 to h-1 do
        For y:= x+1 to h do
            If G[A[x],A[y]]=1 then
                Begin
                    Kt_Trong := False;
                    Exit;
                End;
            KT_Trong := True;
        End;
    End;
Begin
    If i>k then
        Begin
            If KT_Trong(A,k) then Inc(Dem){Hien};
        End
    Else
        For j:=A[i-1]+1 to N-k+i do
            Begin
                A[i] := j;
                Tao_Trong(i+1);
            End;
        End;
End;

```

```

Procedure Tao_Ondinhtrong;
Begin
    Tong := 0;
    For k:=N downto 1 do
        Begin
            Dem := 0;
            FillChar(A,Sizeof(A),0);
            A[0] := 0;
            Tao_Trong(1);
            If Dem>0 then
                Begin
                    Writeln(F2,k);
                    { Tong := Tong +Dem; }
                    Break;
                End;
        End;
    End;
    { Writeln(F2,'Tong cong co ',Tong,' Tap on dinh trong . '); }

End;

```

```

Procedure Tao_Ngoai(i : Byte);

```

Var j : Byte;

Function KT_Ngoai (A : Mang1;h : Byte): Boolean;

Var x,y : Byte;

Function Khongthuoc : Boolean;

Var j : Byte;

Begin

For j:= 1 to h do

 If x=A[j] then

 Begin

 Khongthuoc := False;

 Exit;

 End;

 Khongthuoc := True;

End;

Function CoNoi : Boolean;

Var j : Byte;

Begin

For j:=1 to h do

 If G[x,A[j]]=1 then

 Begin

 CoNoi := True;

 Exit;

 End;

 CoNoi := False;

End;

Begin

For x:=1 to N do

 If Khongthuoc then

 If Not Conoi then

 Begin

 Kt_Ngoai := False;

 Exit;

 End;

 KT_Ngoai := True;

End;

Begin

 If i>k then

 Begin

 If KT_Ngoai(A,k) then Inc(Dem); { Hien };

 End

 Else

 For j:=A[i-1]+1 to N-k+i do

 Begin

 A[i] := j;

 Tao_Ngoai(i+1);

 End;

End;

Procedure Tao_OndinhNgoai;

Begin

 Tong := 0;

 For k:=1 to N do

 Begin

 Dem := 0;

```

    FillChar(A,Sizeof(A),0);
    A[0] := 0;
    Tao_Ngoai(1);
    If Dem>0 then
    Begin
        Writeln(F2,k);
        { Tong := Tong +Dem;}
        Break;
    End;
End;
{      Writeln(F2,'Tong cong co ',Tong,' Tap on dinh ngoai . ');}

End;

```

```

Procedure Vet_Nhan(i : Byte);
Var      j : Byte;
Function KT_Trong (A : Mang1;h : Byte): Boolean;
Var      x,y : Byte;
Begin
    For x:=1 to h-1 do
        For y:= x+1 to h do
            If G[A[x],A[y]]=1 then
                Begin
                    Kt_Trong := False;
                    Exit;
                End;
            KT_Trong := True;
        End;
    End;
Function KT_Ngoai (A : Mang1;h : Byte): Boolean;
Var      x,y : Byte;
Function Khongthuoc : Boolean;
Var      j : Byte;
Begin
    For j:= 1 to h do
        If x=A[j] then
            Begin
                Khongthuoc := False;
                Exit;
            End;
        Khongthuoc := True;
    End;
Function CoNoi : Boolean;
Var      j : Byte;
Begin
    For j:=1 to h do
        If G[x,A[j]]=1 then
            Begin
                CoNoi := True;
                Exit;
            End;
        CoNoi := False;
    End;
Begin
    For x:=1 to N do
        If Khongthuoc then

```

```

    If Not Conoi then
        Begin
            Kt_Ngoai := False;
            Exit;
        End;
    KT_Ngoai := True;
End;

```

```

Begin
    If i>k then
        Begin
            If KT_Ngoai(A,k) and KT_Trong(A,k) then Hien;
        End
    Else
        For j:=A[i-1]+1 to N-k+i do
            Begin
                A[i] := j;
                Vet_Nhan(i+1);
            End;
        End;
End;

```

```

Procedure Tao_Nhan;
Begin
    Tong := 0;
    For k:=1 to N do
        Begin
            Dem := 0;
            FillChar(A,Sizeof(A),0);
            A[0] := 0;
            Vet_Nhan(1);
            If Dem>0 then
                Begin
                    Writeln(F2,Dem,' Tap nhan ',k, ' phan tu .');
                    { Tong := Tong +Dem; }
                    Break;{ CHI TIM TAP NHAN IT PHAN TU NHAT }
                End;
            End;
        { Writeln(F2,'Tong cong co ',Tong,' Tap Nhan . ');}
    End;

```

```

BEGIN
    Clrscr;
    DocF;
    Assign(F2,Fo);
    Rewrite(F2);
    Writeln(F2,'***** SO ON DINH TRONG *****');
    Writeln(F2);
    Tao_Ondinhtrong;
    Writeln(F2);
    Writeln(F2,'***** SO ON DINH NGOAI *****');
    Writeln(F2);
    Tao_Ondinhngoai;
    Writeln(F2);
    Writeln(F2,'***** CAC TAP NHAN IT PHAN TU NHAT *****');
    Writeln(F2);
    Tao_Nhan;

```

Close(F2);
END.

Bài 7 : Tô màu

```

Uses    Crt;
Const   Max  = 20;
        Fi    = 'Tomau3.inp';
Var     A     : Array[1..Max,1..Max] of 0..1;
        Mau,LMau  : Array[1..Max] of Byte;
        N,i      : Integer;
        Somauxudung,SoMauMax : Integer;
Procedure TaoF;
Var i,j,x : Byte;f : Text;
Begin
    Assign(f,fi);
    Rewrite(f);
    Randomize;
    Writeln(f,Max);
    n := Max;
    For i:=1 to n-1 do
    For j:=i+1 to n do
    Begin
        x := random(2);
        If x =1 then Writeln(f,i:4,j:4);
    End;
    Close(f);
End;
Procedure NhapFile;
Var i,j : Integer;
    F : Text;
Begin
    FillChar(A,Sizeof(A),0);
    Assign(F,Fi);
    Reset(F);
    Readln(F,N);
    While not Eof(F) do
    Begin
        Read(F,i);
        While not eoln(F) do
        Begin
            Read(F,j);
            A[i,j] := 1;
            A[j,i] := 1;
        End;
        Readln(F);
    End;
End;
Procedure Hien;
Var i,j : Integer;
Begin
    Writeln;
    For i:=1 to N do
    Begin
        For j:=1 to N do Write(A[i,j]:4);
        Writeln;
    End;
End;

```



```

End;
End;
Procedure Khoitri;
Begin
    FillChar(Mau,sizeof(Mau),0);
    SoMauMax      := N;
    Somauxudung    := 1;
    Mau[1] := 1;
End;

Function Kt(x,m : Integer): Boolean;{ Mau m gan cho dinh x }
Begin
    For i:=1 to N do
        If (A[x,i]=1) and (m=Mau[i]) then
            Begin Kt := False;Exit;End;
        Kt := True;
    End;
Procedure Tomau(x : Integer); { To mau cho dinh x }
Var
    m,Luusomauxudung,Luumaux : Integer;
Begin
    If x=N+1 then
        Begin
            LMau      := Mau;
            SoMauMax := Somauxudung;
            Exit
        End;
    m := 1;
    While m<=SoMauMax do
        Begin
            If (KT(x,m)) then
                Begin
                    LuuMaux := Mau[x];
                    Mau[x] := m;
                    Luusomauxudung := Somauxudung;
                    If Somauxudung<m then Somauxudung := m;
                    Tomau(x+1);
                    Somauxudung := Luusomauxudung ;
                    Mau[x] := LuuMaux;
                End;
            Inc(m);
        End;
    End;
End;
Procedure Thongbao;
Var i : Integer;
Begin
    For i:=1 to N do
        Writeln( ' Diem ',i:2,'      to mau : ',LMau[i]);
    End;
BEGIN
    Clrscr;
    { TaoF; }
    NhapFile;
    Hien;
    Khoitri;

```

```

Tomau(2);
Thongbao;
END.

```

Bài 8 : Dãy đèn màu

```

Uses  Crt;
Const MaxK = 3500;
      Fi    = 'BL1.inp';
      Fo    = 'BL1.out';
Type  KM1   = Array[1..Maxk] of Record x,y : Byte End;
      KM2   = Array[1..Maxk] of Integer;
Var    Time : Longint Absolute $00:$46C;
      Tg    : Longint;
      B     : ^KM1;
      Mau,L Mau : ^KM2;
      N,M,K,dem,Sm1,Sm2,SmMax : Integer;

Procedure TaoFile;
  Var      F           : Text;
           i,j,L,xm,xn,xk : Integer;
           P           : KM1;
Function Ok : Boolean;
  Var      i,j : Integer;
  Begin
    Ok := True;
    For i:=1 to K-1 do
      For j:=i+1 to K do
        Begin
          If (P[j].x=P[i].x) and (P[j].y=P[i].y)
          then
            Begin
              Ok := False;
              Exit;
            End;
          End;
        End;
      End;
    End;
  End;
Begin
  Assign(F,Fi);
  Rewrite(F);
  Write('Nhap M,N,K (M,N<=200, K<250) : ');Readln(xM,xN,xK);
  Writeln(F,xM,' ',xN,' ',xK);
  Repeat
    For L:=1 to xk do
      Begin
        i := Random(200);
        j := Random(100)+Random(100);
        P[L].x := i;
        P[L].y := j;
      End;
    Until Ok;
    For i:=1 to xk do Writeln(F,P[i].x,' ',P[i].y);
  Close(F);
End;
Procedure NhapFile;

```

```

Var    i,j      : Integer;
        F       : Text;
Begin
    New(B);
    Assign(F,Fi);
    Reset(F);
    Readln(F,M,N,K); { M<=200,N<=200,K<3500 }
    For i:=1 to K do
        Begin
            B^[i].x := 0;
            B^[i].y := 0;
        End;
    For i:=1 to K do
        Readln(F,B^[i].x,B^[i].y);
    Close(F);
End;

```

```

Procedure Greedy;
Var    ii,i,j,Maxm : Integer;
        Lienquan   : KM2;
        Dato,chuato : Array[1..MaxK] of Boolean;

```

```

Procedure GhiGreedy;
Var    i      : Integer;
        F2    : Text;
Begin
    Assign(F2,Fo);
    ReWrite(F2);
    Writeln(F2,Sm1);
    Writeln(F2,'Hau an ');
    For i:=1 to k do
        Writeln(F2, ' Diem ',i:2,'    to mau : ',Mau^[i]);
    Writeln('Da ghi duoc 1 nghiem vao file ',Fo
        ,#13#10'... Bay gio tim nghiem tot hon ... ');
    Close(F2);
End;

```

```

Begin
    For i:=1 to k do Dato[i] := False;
    For i:=1 to k do Chuato[i] := True;
    For i:=1 to k do Mau^[i] := 0;
    Mau^[1]:=1;
    dato[1]:= True;
    chuato[1] := False;
    Maxm := 1;
    For i:=1 to k do
        Begin
            If chuato[i] then
                Begin
                    For j:=1 to k do Lienquan[j] := 0;
                    For j:=1 to k do
                        If (i<>j) and ((B^[i].x=B^[j].x) or (B^[i].y=B^[j].y))
                            and (Mau^[j]>0) then Lienquan[Mau^[j]] := 1;
                    For j:=1 to maxm+1 do
                        If Lienquan[j]=0 then

```

```

    Begin
        Sm1 := j;
        Break;
    End;
    If Sm1 <= maxm then Mau^[i] := Sm1
    Else
        Begin
            Inc(Maxm);
            Mau^[i] := Maxm ;
        End;
        Dato[i] := True;
        Chuato[i] := False;
    End;
End;
Sm1 := 0;
For i:=1 to k do If Mau^[i]>Sm1 then Sm1 := Mau^[i];
GhiGreedy;
End;

```

```

Procedure Vet;
Procedure Khoitri;
Var      i      : Integer;
Begin
    For i:=1 to K do Mau^[i] := 0;
    SmMax    := k;
    Sm2      := 1;
    Mau^[1] := 1;
End;
Function Kt(x,m : Integer): Boolean; { Mau m gan cho dinh x }
Var      i      : Integer;
Begin
    For i:=1 to k do
        If ((B^[i].x=B^[x].x)or(B^[i].y=B^[x].y)) and (m=Mau^[i]) then
            Begin Kt := False;Exit;End;
        Kt := True;
    End;
End;
Procedure GhiVet;
Var      i      : Integer;
        F2      : Text;
Begin
    Assign(F2,Fo);
    ReWrite(F2);
    Writeln(F2,'Vet - So mau : ',SmMax);
    For i:=1 to k do
        Writeln(F2, ' Diem ',i:2,'    to mau : ',LMau^[i]);
    Close(F2);
End;
Procedure Tomau(x : Integer); { To mau cho dinh x }
Var      m,luu,Luumaux : Integer;
Begin
    If x=K+1 then
        Begin
            LMau    := Mau;
            SmMax    := Sm2;
            If (Sm2<Sm1) and (dem=0) then

```

```

Begin
  Ghivet;
  Inc(dem);
End;
If ((Time-tg)/18.2)>30 then
Begin
  Ghivet;
  Writeln('Nghiem tot hon thay cho nghiem cu da ghi vao file ',Fo);
  Readln;
  Halt;
End
Else
Exit;
End;
m := 1;
While m<=SmMax do
Begin
  If (KT(x,m)) then
  Begin
    LuuMaux := Mau^[x];
    Mau^[x] := m;
    Luu := Sm2;
    If Sm2<m then Sm2 := m;
    Tomau(x+1);
    Sm2 := Luu;
    Mau^[x] := LuuMaux;
  End;
  Inc(m);
  If ((Time-tg)/18.2)>31 then
  Begin
    Writeln('Khong du thoi gian tim thay nghiem tot hon ');
    Readln;
    Halt;
  End
End;
End;
Begin
  Khoitri;
  Tomau(2);
End;

BEGIN
  Clrscr;
  { TaoFile; }
  New(Mau);
  New(LMau);
  Tg := Time;
  dem := 0;
  NhapFile;
  Greedy;
  Vet;
  Writeln('Nghiem toi uu thay cho nghiem truoc da ghi vao file ',Fo);
END.

```

Cách 2 (Bài làm của Lê Hồng Việt 11CT 1997-1998)

```

Uses  Crt;
Const  Fi      = 'Bl1.inp';
        Fo      = 'Bl1.out';
        Max     = 200;
        Maxsize=10000;

Type   Rec=   record
                x,y:byte;
            end;
        Mang   = array[1..maxsize] of rec;
        Mang2  = array[1..maxsize] of integer;
        Mang3  = array[1..max,1..max] of byte;
        Mang4  = array[1..max] of integer;

Var    F          : text;
        don,cot: mang4;
        A          : ^mang;
        tt,tm      : ^mang2;
        Mau        : mang3;
        M,N,K,maxmau : integer;

Procedure docF;
    Var i:integer;
    Begin
        Assign(f,fi);
        {$I-}reset(F);{$I+}
        If ioresult <>0 then
            Begin
                writeln('Loi file hoac khong tim thay file '+fi);
                readln;
                Halt;
            end;
        readln(f,m,n,k);
        for i:=1 to k do
            with A^[i] do
                read(f,x,y);
                close(F);
            end;
    end;

Procedure Hien;
    Var i:integer;
    Begin
        for i:=1 to k do
            with A^[i] do
                writeln(x,y:4)
            end;
    end;

Procedure Hienm(Var A:mang3);
    var i,j:integer;
    Begin

```

```

for i:=1 to m do
  begin
    for j:=1 to n do
      write(mau[i,j]:2);
    writeln;
  end;
end;

```

```

Procedure coc(i,j:integer);
Var c:rec;
  p:integer;
Begin
  c:=A^[i];
  A^[i]:=A^[j];
  A^[j]:=c;
  {p:=tt[i];
  tt[i]:=tt[j];
  tt[j]:=p;}
end;

```

```

Procedure trendongcot;
Var i:integer;
Begin
  fillchar(don,sizeof(don),0);
  fillchar(cot,sizeof(cot),0);
  for i:=1 to k do
    with A^[i] do
      Begin
        inc(don[x]);
        inc(cot[y]);
      end;
  end;
end;

```

```

Procedure Doidl;
var i:integer;
Begin
  fillchar(mau,sizeof(mau),0);
  for i:=1 to k do
    with A^[i] do
      mau[x,y]:=1;
  end;
end;

```

```

Procedure Init;
Begin
  Fillchar(tm^,sizeof(tm^),0);
end;

```

```

Function dembac(x,y:integer):integer;
Begin
  dembac:=don[x]+cot[y];
end;

```

```

Function timmax:integer;
Var i,m,li:integer;

```

```

Begin
  m:=0;
  li:=0;
  for i:=1 to k do
    If Mau[A^[i].x,A^[i].y]<>0 then
      If tt^[i]>m then
        Begin
          li:=i;
          m:=tt^[i];
        end;
      timmax:=li;
  end;

```

```

Procedure timbac;
Var i:integer;
Begin
  Fillchar(tt^,sizeof(tt^),0);
  for i:=1 to k do
    Begin
      tt^[i]:=Dembac(A^[i].x,A^[i].y);
    end
  end;

```

```

Procedure Bot(x,y:integer);
Begin
  dec(Don[x]);
  dec(cot[y]);
end;

```

```

Function Maumin(x,y:byte):integer;
var i:integer;
  P:array[1..max*max+1] of byte;
begin
  fillchar(p,sizeof(p),0);
  for i:=1 to k do
    If ((A^[i].x=x) and (A^[i].y<>y)) or ((A^[i].x<>x) and (A^[i].y=y)) then
      If Tm^[i]>0 then
        P[tm^[i]]:=1;
  i:=1;
  while p[i]<>0 do inc(i);
  maumin:=i;
end;

```

```

Procedure Tomau;
var i,li,j:integer;
Begin
  maxmau:=0;
  repeat
    timbac;
    i:=timmax;
    If i=0 then break;
    j:=maumin(A^[i].x,A^[i].y);
    If j>maxmau then maxmau:=j;
    Tm^[i]:=j;
    Mau[A^[i].x,A^[i].y]:=0;
  repeat

```



```

    bot(A^[i].x,A^[i].y);
until false;
end;

```

```

Procedure Hienkq;
var i:integer;
Begin
    Assign(f,fo);
    rewrite(f);
    writeln(f,Maxmau);
    for i:=1 to k do
        writeln(f,tm^[i]);
    Close(f);
end;

```

```

Procedure Batdau;
Begin
    New(a);
    New(tt); New(tm);
end;

```

```

Procedure Ketthuc;
Begin
    dispose(a);
    dispose(tt);
end;

```

```

Procedure Make;
Begin
    doidl;
    Hienm(mau);
    trendongcot;
    Tomau;
    Hienkq;
end;

```

```

BEGIN
    Clrscr;
    Batdau;
    Init;
    DocF;
    Hien;
    Make;
    Ketthuc;
END.

```

Cách làm 3 : Bài làm của Lê Sỹ Vinh 12 CT - 1997-1998

```

{ $A+,B+,D+,E+,F-,G-,I+,L+,N-,O-,P-,Q-,R+,S+,T-,V+,X+ }
{ $M 16384,0,655360 }
uses crt;
Const max =201;
    maxsize =1000;
    TimeOver =182*2;
    Input ='bl1.INP';
    Output ='bl1.Out';

```

```

type mang =array[0..max] of Integer;
Ta  =Array[0..max] of ^mang;
Tb  =Array[0..max] of Integer;

```

```

Var a   : Ta;
Th, Tc : Tb;
Cx  : Array[1..maxsize] of Byte;
M,N,Sd,Csh, Csc , maxmau, Liumaxmau , Time : Longint;

```

```

Procedure Read_Input;

```

```

var f : text;
i,j,x,y : Longint;
begin
  assign(f, Input); Reset(f);
  Readln(F, M,N, sd);
  For i:=0 to N Do New(A[i]);

  For i:=0 to N Do
    For j:=0 to m Do A[i]^j:=0;

  for i:=1 to Sd do
  begin
    readln(f, x,y);
    a[y]^x:=1;
  end;
  close(f);
end;

```

```

Procedure Hienm;

```

```

Var i,j : Longint;
Begin
  For i:=n downto 0 Do
  Begin
    For j:=0 to M Do Write(A[i]^j:3);
    Writeln;
  End;
  Writeln;
End;

```

```

procedure Greedy0;

```

```

Var i,j, St : Longint;
Begin
  If M>N Then maxmau:=M+1
  Else maxmau:=N+1;

  St:=0;
  For i:=0 to N Do
  Begin
    Inc(St);
    For j:=0 to M Do
    Begin
      If A[i]^j>0 Then
        If St+j>maxmau Then A[i]^j:=St+j-maxmau+1
        Else A[i]^j:=St+j+1;
    End;
  End;

```

```

    End;
    End;
    maxmau:=maxmau+1;
End;

Procedure Taomangthc;
Var i,j : Longint;
begin
    for i:=1 to m do th[i]:=0;
    for i:=1 to n do tc[i]:=0;

    For i:=0 to N Do
        For j:=0 to M Do
            Begin
                Th[i]:=th[i]+A[i]^j;
                Tc[j]:=tc[j]+A[i]^j;
            End;
        end;
    end;

procedure timhangmax;
var i : Longint;
begin
    csh:=0;
    For i:=1 to n Do
        if th[i]>th[csh] then csh:=i;
    end;

procedure timcotmax;
var i : Longint;
begin
    csc:=0;
    For i:=1 to m Do
        if tc[i]>tc[csc] then csc:=i;
    end;

procedure lesson1;
Var i,min : Longint;
Begin
    min:=0;
    For i:=0 to m do
        if (a[csh]^i=1) And (tc[i]>min) Then
            begin
                Csc:=i; min:=tc[i];
            end;
    End;

Procedure Lesson2;
Var i,min : Longint;
begin
    Min:=0;
    For i:=0 to n do
        if (a[i]^csc=1) and (th[i]>Min) then
            Begin
                Min:=Th[i]; Csh:=i;
            End;

```

end;

Procedure Tomaudiem;

Var i : Longint;

begin

Fillchar(Cx, Sizeof(cx),0);

For i:=0 to N Do

If $A[i]^{[csc]} > 1$ Then $Cx[A[i]^{[Csc]}] := 1$;

For i:=0 to M Do

IF $A[Csh]^i > 1$ then $Cx[A[Csh]^i] := 1$;

i:=1;

Repeat

i:=i+1;

Until $Cx[i] = 0$;

If $i > \text{maxmau}$ Then $\text{maxmau} := i$;

$A[Csh]^{[Csc]} := i$;

$Th[Csh] := Th[Csh] - 1$;

$tc[Csc] := Tc[Csc] - 1$;

End;

procedure Greedy1;

Var i,j : Longint;

Begin

taomangthc;

For i:=1 to Sd Do

Begin

timhangmax;

Timcotmax;

If $(A[Csh]^{[Csc]} \neq 1)$ Then

Begin

If $th[csh] > tc[csc]$ then lesson1

Else lesson2;

End;

ToMaudiem;

If $\text{Meml}[0:\$46c] - \text{time} > \text{TimeOver}$ Then Exit;

End;

End;

Procedure Hienkq;

Var f, fr : text;

i,x,y : Longint;

Begin

If $\text{maxmau} < \text{Liumaxmau}$ Then

Begin

$\text{Liumaxmau} := \text{maxmau}$;

Assign(Fr, Output); ReWRite(Fr);

WRiteln(fr, $\text{maxmau} - 1$);

Assign(F, Input); Reset(F);

Readln(f, M,N, Sd);

For i:=1 to Sd Do

Begin

Readln(F, x,y);

Writeln(fr, $a[y]^x - 1$);

End;

```

    Close(F);
    Close(Fr);
  End;
End;

procedure GiaiPhong;
  Var i : byte;
  Begin
    For i:=0 to N Do Dispose(A[i]);
  End;

begin
  Clrscr;
  Time:=Meml[0:$46c];
  liumaxmau:=maxint;

  Read_Input;
  Greedy0;
  Hienkq;
  GiaiPhong;

  Maxmau:=1;
  Read_Input;
  Greedy1;
  If meml[0:$46c]-Time<TimeOver Then Hienkq;
  GiaiPhong;
end.

```

CHƯƠNG I : DUYỆT KHÔNG ĐỆ QUY

I / Nhân xét :

Các chương trình có thể viết dưới dạng “ Duyệt bằng đệ quy “ khi nó phải thực hiện nhiệm vụ P có hình thức đệ quy sau đây :

$$P = (\text{Nếu } B_0 \text{ thì } S ; \text{Nếu } B_k \text{ thì } P)$$

trong đó S là một số công việc phải thực hiện khi có điều kiện kết thúc B_0 của đệ quy , còn B_k là điều kiện cần để thực hiện nhiệm vụ P ở bước thứ k . Trong mỗi bước gọi thực hiện P thì điều kiện B_k được thu hẹp dần để dẫn tới tình trạng kết thúc B_0 của quá trình duyệt .

Song do chương trình đệ quy được tổ chức bằng Stack (ngăn xếp) trong bộ nhớ có kích thước tối đa là 16kb nên khi gặp những chương trình đệ quy quá sâu thường bị tràn Stack của bộ nhớ (ngăn xếp của chương trình đệ quy không đủ chứa các hàm và thủ tục đệ quy của nó) . Trong những trường hợp như thế , người ta thường chuyển sang chương trình viết dưới dạng “Duyệt không đệ quy “ thay đệ quy bằng vòng lặp , dựa vào công thức sau :

$$P = (G_0 ; \text{Trong khi } B_k \text{ thì } P_k)$$

G_0 : một số lệnh gán trị ban đầu

B_k : điều kiện cần để thực hiện công việc P_k

II / Một số thí dụ :

Thí dụ 1 : Xây dựng hàm Fibonacci bằng đệ quy và không đệ quy

```

Function Fibonacci(N : Integer) : Integer;
Begin
    If N=0 then Fibonacci =1      { N=0 hoặc N=1 là điều kiện B0 }
    Else
    If N=1 then Fibonacci =1
    Else
    Fibonacci := Fibonacci(N-1)+ Fibonacci(N-2)
End;

```

```

Function Fibonacci(N : Integer) : Integer;
Var i,p,U0,U1 : Integer;
Begin
    i := 0;
    U0 := 0;
    U1 := 1;
    While i < N do
    Begin
        Inc(i);
        p := U1;
        U1 := U0+U1;
        U0 := p;
    End;
    Fibonacci := p;
End;

```

Thí dụ 2 : Sắp xếp mảng bằng thuật toán QuickSort :

Kiểu đệ quy

```

Program QSort;
{$R-,S-}
Uses Crt;
Const Max = 30000;
Type List = Array[1..Max] of Integer;
Var Data : List;
    I : Integer;

Procedure QuickSort(Var A: List; Lo, Hi: Integer);
Procedure Sort(L, r: Integer);
Var i, j, x, y: integer;
Begin
    i := L;
    j := r;
    x := a[(L+r) DIV 2];
    Repeat
        While a[i] < x do i := i + 1;
        While x < a[j] do j := j - 1;
        If i <= j then
            Begin
                y := a[i];
                a[i] := a[j];
                a[j] := y;
                i := i + 1;
                j := j - 1;
            End;
        Until i > j;
    End;
End;

```

```

        End;
        until i > j;
        If L < j then Sort(L, j);
        If i < r then Sort(i, r);
    End;
Begin
    Sort(Lo,Hi);
End;

BEGIN {QSort}
    Write('Hiện đang tạo ',max,', ' số ngẫu nhiên...');
    Randomize;
    For i := 1 to Max do Data[i] := Random(30000);
    Writeln;
    Write('Hiện đang sắp xếp các số...');
    QuickSort(Data, 1, Max);
    Writeln;
    For i := 1 to Max do Write(Data[i]:8);
    Readln;
END.

```

Kiểu không đệ quy

```

Uses  Crt;
Const MN  = 4000;
Type  cs   = 1..MN;
      Pt   = Record
              ma : Cs;
              gt : Integer;
            End;
      M1   = Array[1..MN] of pt;
      M2   = Array[1..MN] of Record tr,ph : cs End;
Var    i,N  : cs;
      A    : M1;
      B    : M2;
Procedure H;
    Var    s,i,j,tr,ph    : cs;
           x              : Integer;
           coc            : Pt;

    Begin

        s := 1;          { Công việc G0 : Nạp phần tử thứ nhất vào Stack B }
        B[s].tr := 1;
        B[s].ph := N;
        Repeat { Thực hiện cho đến gặp điều kiện kết thúc B0 : Stack rỗng ( s=0) }
            tr := B[s].tr;      { Lấy 1 phần tử ở đỉnh Stack }
            ph := B[s].ph;
            Dec(s);
            Repeat { Điều kiện thực hiện 1 lần sắp xếp là : tr < ph }
                i := tr;
                j := ph;
                x := A[(tr+ph) div 2].gt;
                Repeat
                    While A[i].gt < x do inc(i);
                    While A[j].gt > x do dec(j);
                    If i <= j then
                        Begin

```

```

        coc := A[i];
        A[i] := A[j];
        A[j] := coc;
        Inc(i);
        Dec(j);
    End;
Until i>j;
If i<ph then
Begin
    Inc(s);
    B[s].tr := i;
    B[s].ph := ph;
End;
ph := j;
Until tr >= ph;
Until s = 0;
End;
Procedure DocF;
Const Fi = 'qsort0dq.txt';
Var F : Text; i : cs;
Begin
    Assign(F,Fi);
    Reset(F);
    Readln(F,N);
    For i:=1 to N do
    Begin
        Readln(F,A[i].gt);
        A[i].ma := i;
    End;
    Close(F);
End;
Procedure Hienkq;
Var i : Cs;
Begin
    For i:=1 to N do Write(A[i].ma:4);
    Writeln;
    For i:=1 to N do Write(A[i].gt:4);
End;
Procedure TaoF;
Const Fi = 'qsort0dq.txt';
Var F : Text; i : cs;
Begin
    Assign(F,Fi);
    ReWrite(F);
    N := 4000;
    Writeln(F,N);
    For i:=1 to N div 2 do Writeln(F,i);
    For i:= N div 2+1 to N do Writeln(F,i-(N div 2));
    Close(F);
End;
Begin
    TaoF;
    DocF;
    H;
    Hienkq;

```


End.

Thí dụ 3 :

Cho 3 ký tự A,B,C . Hãy tạo xâu có độ dài $M \leq 250$ chỉ chứa 3 ký tự này có tính chất : Không có 2 xâu con liên nhau bằng nhau .

Kiểu đệ quy

```

Uses  Crt;
Const N      = 20;
Var    S      : String;
Function Kt(S : String) : Boolean;
    Var i,j : Byte;
    Begin
        Kt := True;
        For i:=1 to Length(S) div 2 do
            For j:=1 to Length(S)- 2*i+1 do
                If Copy(S,j,i)=Copy(S,j+i,i) then
                    Begin
                        Kt := False;
                        Exit;
                    End;
            End;
        End;
    End;
Procedure Tao(S : String);
    Var ch : Char;
    Begin
        If Length(S)=N then
            Begin
                Writeln(S);
                Readln;
                Halt;
            End;
        For ch:='A' to 'C' do      { Khởi tạo mọi khả năng }
            Begin
                S := S+ch;      { Thử chọn 1 khả năng }
                If Kt(S) then Tao(S) { Nếu thoả mãn điều kiện thì tìm tiếp }
                Else Delete(S,Length(S),1); { Nếu không thì trả về trạng thái cũ }
            End;
        End;
    End;
BEGIN
    Clrscr;
    S := "";
    Tao(S);
END.
```

Cách giải đệ quy ở trên chỉ áp dụng đ-ợc khi $\text{Length}(S) \leq 20$. Sau đây là cách giải không đệ quy , có thể áp dụng với S có $\text{Length}(S) \leq 250$.

Kiểu không đệ quy

```

Uses  Crt;
Const    Max    = 100; { co the toi 250 }
Var      A      : Array[1..Max] of Integer;
          S      : String;
          i,j    : Integer;
```

Function Duoc(S : String):Boolean;

Var i,j : Integer;

S1,S2 : String;

Begin

Duoc := False;

S1 := "";

S2 := "";

For i:=1 to Length(S) div 2 do { do dài của các xâu con }

Begin

For j:=1 to (Length(S)-2*i+1) do { điểm đầu của xâu con S1 }

Begin

S1 := Copy(S,j,i);

S2 := Copy(S,j+i,i);

If S1=S2 then Exit;

End;

End;

Duoc := True;

End;

Procedure Tim;

Begin

For i:=1 to Max do A[i] := 1;

i := 1;

S := 'A';

While (Length(S)<Max) and (i>0) do

Begin

If A[i]<4 then { A[i]<4 cho biết còn ký tự cho vào S[i+1] }

Begin

If Duoc(S+Char(A[i]+64)) then

Begin

S := S + Char(A[i]+64);

A[i] := A[i]+1;

Inc(i);

End

Else

Inc(A[i]);

End

Else { A[i]=4 : mọi ký tự 'A','B','C' cho vào S[i+1] không thành công, phải xóa S[i] đi, quay lui }

Begin

Delete(S,Length(S),1);

A[i] := 1;

Dec(i);

End;

End;

Writeln;

If i=0 then Writeln('Không có xâu dài ', Max , ' thỏa mãn ')

Else Writeln(s);

End;

BEGIN

Clrscr;

Tim;

Readln;

END.

BÀI TỐP VỬ NHÀ

- 1) Viết chương trình tạo các hoán vị của bộ $(1, 2, 3, \dots, 9)$ bằng duyệt không đệ quy
- 2) Xâu nhị phân là xâu chỉ chứa các ký tự 1 và 0. Xâu nhị phân S được gọi là không lặp bậc L nếu: Các xâu con có độ dài L của nó đều khác nhau từng đôi một. Xâu nhị phân không lặp bậc L được gọi là cực đại nếu việc bổ xung vào bên trái hoặc bên phải của xâu một ký tự 1 hoặc 0 thì sẽ phá vỡ tính không lặp bậc L của xâu.

Viết chương trình xác định xâu nhị phân không lặp bậc L cực đại, ngắn nhất bằng duyệt đệ quy và duyệt không đệ quy.

Cho một bảng hình chữ nhật kích thước $M \times N$, M, N nguyên dương, ($M, N \leq 50$). Hình chữ nhật này được chia thành $M \times N$ ô vuông bằng nhau bởi các đường song song với các cạnh trên ô vuông $[i, j]$ ghi số $A[i, j] \leq 50$, từ bảng A ta lập bảng B mà $B[i, j]$ được tính như sau: Biểu diễn $A[i, j]$ thành tổng nhiều nhất các số nguyên tố trong đó có nhiều nhất 1 số được xuất hiện nhiều nhất là 2 lần, $B[i, j]$ bằng số số hạng của biểu diễn này kể cả số bội. Ví dụ: $A[i, j] = 10 = 2 + 3 + 5$ thì $B[i, j] = 3$, $A[i, j] = 12 = 2 + 2 + 3 + 5$ thì $B[i, j] = 4$.

- 1) Nhập từ File INPUT.TXT trong đó dòng đầu ghi 2 số M, N . M dòng sau ghi M dòng của mảng A (Không cần kiểm tra dữ liệu) ghi ra File OUT.TXT mảng B, mỗi dòng 1 dòng của bảng.
- 2) Tìm hình chữ nhật lớn nhất gồm các ô của bảng B ghi các số như nhau.

BÀI CHÈA

Bài 1:*Kiểu đệ quy*

```

Uses    Crt;
Const   N      = 9;
        TF     = 'hoanvi.txt';
Type    TS     = String[N];
Var     S      : TS;
        d, Lt  : Longint;
        F      : Text;
        T      : LongInt Absolute $0000:$046C;

Procedure Doi(Var a, b : Char);
    Var p : Char;
    Begin
        p := a; a := b; b := p;
    End;

Procedure Hien(S : TS);
    Begin
        Inc(d); Write(F, S, ' ');
    End;

```

```

    If (d mod 10 = 0) then Writeln(F);
  End;
Procedure Tao(S : String; i : Byte);
  Var   j       : Byte;
        p       : Char;
  Begin
    If i=N then Hien(S);
    For j:=i to N do
      Begin
        Doi(S[i],S[j]);
        Tao(S,i+1);
      End;
    End;
  End;
BEGIN
  Clrscr;
  S := '123456789';
  S := Copy(S,1,N);
  d := 0;
  LT := T;
  Assign(F,TF);
  ReWrite(F);
  Tao(S,1);
  Close(F);
  Writeln(#13#10,'So hoan vi la   ',d);
  Writeln('Mat thoi gian la : ',((T-Lt)/18.2):10:2,' giay');
  Readln;
END.

```

Kiểu không đệ quy

```

Uses  Crt;
Const Max  = 9;
      Fo    = 'hoanvi.txt';
Type   K1   = Array[1..Max] of Integer;
Var    F     : Text;
      N,i,j   : Integer;
      V       : K1;
      dem     : LongInt;

Procedure Tao;
  Var j,k  : Integer;

  Procedure Hien;
  Var j : Byte;
  Begin
    Begin
      For j:=1 to N do Write(F,V[j]);Write(F,' ');
      Inc(dem);
      If (dem mod (79 div (N+1)))=0) then Writeln(F);
      Dec(k);
    End
  End;

  End;

Procedure TaoVk;
Var Ok : Boolean;

```

```

Begin
  Repeat
    j := 1;
    While V[k] <> V[j] do Inc(j);
    If j=k then Ok := True
    Else
      Begin
        Ok := False;
        Inc(V[k]);
      End
    Until Ok;
End;

```

```

Begin
  Assign(F,Fo);
  ReWrite(F);
  For k:=1 to N do V[k] := -1;
  V[1] := 1;
  k := 2;
  Repeat
    If k>N then Hien
    Else
      If V[k]=-1 then
        Begin
          V[k] := 1;
          TaoVk;
          Inc(k);
        End
      Else
        Begin
          Inc(V[k]);
          TaoVk;
          If V[k]<=N then Inc(k)
          Else
            Begin
              V[k] := -1;
              Dec(k);
            End;
        End;
    Until k=0;
End;

```

```

BEGIN
  Repeat
    Clrscr;
    dem := 0;
    Write('Tao cac hoan vi cua N chu so lien tiep 1..N . Nhap N = ');
    Readln(N);
    Tao;
    Writeln(F);
    Writeln(F,'So hoan vi la : ',dem );
    Close(F);
    Writeln('ESC thoat ');
  Until ReadKey=#27;
END.

```

Bài 2 :*Kiểu đệ quy*

```

Uses  Crt;
Const  Max  = 13;
Var  L  : Byte;
      S  : String;
Procedure Nhap;
  Var  Ok : Boolean;
  Begin
    Write('Nhập bậc L của xâu nhị phân không lặp , L = ');
    Repeat
      {$i-}Readln(L);{$i+}
      Ok := (Ioresult=0) and (L<=Max);
      If Not Ok then Writeln('Nhập lại ');
    Until Ok;
  End;
Procedure Taoxau;
Function Ktra1(S : String): Boolean;
  Var i,j : Byte;
  Begin
    Ktra1 := True;
    If Length(S)>=L then
      For i:=1 to Length(S)-L+1 do
        For j:=i+1 to length(S)-L+1 do
          If copy(S,i,L)=copy(S,j,L) then
            Begin
              Ktra1 := False;
              Exit;
            End;
          End;
  End;
Function Ktra2: Boolean;
  Begin
    Ktra2 := False;
    If (Not Ktra1('0'+S)) and (Not Ktra1('1'+S)) and
      (Not Ktra1(S+'0')) and (Not Ktra1(S+'1')) then
      Ktra2 := True;
  End;
Procedure Tim;
  Var i : Byte;
  Begin
    If Ktra2 then
      Begin
        Writeln('Xâu nhị phân không lặp bậc L cực đại, ngắn nhất : ');
        Writeln(S);
        Exit;
      End;
    For i:=0 to 1 do
      Begin
        S := S+Char(i+48);
        If Ktra1(S) then Tim
        Else Delete(S,length(S),1);
      End;
  End;

```

```

End;
Begin
  S := "";
  Tim;
End;
BEGIN
  Clrscr;
  Repeat
    Nhap;
    Taoxau;
    Writeln('ESC thoat ... ');
  Until Readkey=#27;
END.

```

Kiểu không đệ quy :

```

Uses  Crt;
Const Max  = 255;
Var  L      : Byte;
      S      : String;
Procedure Nhap;
  Var  Ok : Boolean;
  Begin
    Write('Nhap bac L cua xau nhi phan khong lap , L = ');
    Repeat
      {$i-}Readln(L);{$i+}
      Ok := (Ioresult=0) and (L<=Max);
      If Not Ok then Writeln('Nhap lai ');
    Until Ok;
  End;
Procedure Taoxau;
Function Ktra1(S : String): Boolean;
  Var i,j : Byte;
  Begin
    Ktra1 := True;
    If Length(S)>=L then
      For i:=1 to Length(S)-L+1 do
        For j:=i+1 to length(S)-L+1 do
          If copy(S,i,L)=copy(S,j,L) then
            Begin
              Ktra1 := False;
              Exit;
            End;
          End;
  End;
Function Ktra2: Boolean;
  Begin
    Ktra2 := False;
    If (Not Ktra1('0'+S)) and (Not Ktra1('1'+S)) and
      (Not Ktra1(S+'0')) and (Not Ktra1(S+'1')) then
      Ktra2 := True;
  End;
Procedure Tim;
  Var i,k : Byte;
      Ok : Boolean;
  Begin

```

```

S := "";
Repeat
  Ok := False;
  i := 0;
  While (i<2) and (Not Ok) do
    Begin
      Ok := Ktra1(S+char(i+48));
      If Ok then S := S + Char(i+48);
      Inc(i);
    End;
  Until Ktra2;
End;
Begin
  S := "";
  Tim;
  Writeln(S);
End;
BEGIN
  Repeat
    Clrscr;
    Nhap;
    Taoxau;
    Writeln('ESC thoát ... ');
  Until Readkey=#27;
END.

```

PHẦN 2 : ÔN THỰC HÀNH, NÂNG CẤP CHU TRÌNH EULER - CHU TRÌNH HAMINTON

I / Định nghĩa :

- 1 - Trong đồ thị vô hướng : Đường đi qua tất cả các cạnh, mỗi cạnh qua đúng 1 lần , gọi là đường đi Euler. Chu trình đi qua tất cả các cạnh, mỗi cạnh qua đúng 1 lần , gọi là chu trình Euler.
- 2 - Đồ thị vô hướng có đường đi Euler gọi là đồ thị nửa Euler
Đồ thị vô hướng có chu trình Euler gọi là đồ thị Euler
- 3 - Định lý Euler : Đồ thị vô hướng, liên thông G là đồ thị Euler khi và chỉ khi mọi đỉnh đều có bậc chẵn .
Đồ thị vô hướng , liên thông là đồ thị nửa Euler khi và chỉ khi nó có không quá 2 đỉnh bậc lẻ .
- 4 - Trong đồ thị có hướng : Mạch đi qua mọi cung, mỗi cung chỉ 1 lần gọi là mạch Euler
Đồ thị có hướng , nếu tại mỗi đỉnh số cung đi vào bằng số cung đi ra thì ta gọi đồ thị này là tựa đối xứng .
Định lý : Đồ thị có hướng, liên thông và tựa đối xứng thì có mạch Euler
- 5 - Trong đồ thị có hướng : Mạch đi qua tất cả các đỉnh , mỗi đỉnh chỉ 1 lần , gọi là mạch Haminton ; nếu mạch này đóng thì gọi là mạch đóng Haminton . Dây chuyền đơn đi qua tất cả các đỉnh , mỗi đỉnh chỉ 1 lần , gọi là dây chuyền đơn Haminton . đồ thị gọi là nửa Haminton .
- 6 - Trong đồ thị vô hướng : Đường đi qua tất cả các đỉnh , mỗi đỉnh chỉ 1 lần , gọi là đường đi Haminton ; chu trình đi qua tất cả các đỉnh , mỗi đỉnh chỉ 1 lần (trừ đỉnh đầu trùng đỉnh cuối) gọi là chu trình Haminton ; đồ thị tựa đối xứng cũng gọi là đồ thị nửa Haminton (vô hướng) hoặc Haminton (vô hướng)
- 7 - Định lý : (Koríc) Nếu đồ thị đầy đủ (giữa 2 đỉnh bất kỳ đều có ít nhất 1 cung) thì tồn tại mạch Haminton

8 - Định lý : (Dirak) Đơn đồ thị vô hướng G có n đỉnh ($n \geq 3$) có bậc của mọi đỉnh đều $\geq n/2$ thì đồ thị là Haminton.

Đồ thị có hướng G có n đỉnh ($n \geq 3$) liên thông mạnh và có bán bậc vào, bán bậc ra của mọi đỉnh đều $\geq n/2$ thì đồ thị là Haminton.

9 - Định lý :

Nếu đỉnh x chỉ có cung đi ra thì mọi mạch Haminton có đỉnh x là nút đầu tiên

Nếu đỉnh y chỉ có cung đi vào thì mọi mạch Haminton có đỉnh y là nút cuối cùng

10 - Định lý : Nếu x là đỉnh treo (chỉ có 1 cung duy nhất dính với nó - đi tới nó hoặc từ nó đi ra -) thì mọi đường đi Haminton M đều có nút đầu tiên hoặc cuối cùng là x . Đỉnh kề với x trong đồ thị G cũng là đỉnh kề với x trong mạch Haminton M

II / Thuật toán Fleury tìm chu trình Euler (trong đồ thị vô hướng):

B- ớc 1 : Xuất phát từ 1 đỉnh x_i tùy ý .

B- ớc 2 : Vòng lặp

+ Chọn 1 cạnh xuất phát từ x_i tới x_k có tính chất : nếu xoá nó khỏi đồ thị thì phần đồ thị còn lại vẫn liên thông . (gọi là tính chất A)

+ Xoá cạnh đã chọn .

+ Gán $x_i := x_k$

+ B- ớc 2 đ- ọc lặp cho đến khi không chọn đ- ọc cạnh có tính chất A nêu trên ; lúc này hoặc là hết cạnh , hoặc cạnh đó là cầu sang vùng liên thông mới . Nếu hết cạnh thì kết thúc còn không thì sang b- ớc 3

B- ớc 3 : Qua cầu , xoá điểm cô lập (hoặc xử lý gián tiếp : tăng số vùng liên thông) , về b- ớc 2.

III / Tìm đường đi Haminton bằng đệ quy:

Giả sử đã tìm đ- ọc mạch k đỉnh , cần bổ xung đỉnh thứ $k+1$ vào chỗ thích hợp của mạch này , ta chọn 1 trong 3 tr- ờng hợp sau :

+ Tr- ờng hợp 1 : có cung nối x_k với x_{k+1} thì cho mạch đi tiếp tới x_{k+1}

+ Tr- ờng hợp 2 : có cung nối x_{k+1} tới x_1 thì thêm cung (x_{k+1}, x_1) vào đầu mạch

+ Tr- ờng hợp 3 : soát từ x_k về đầu mạch cho đến khi gặp x_m mà có cung nối x_m với x_{k+1} thì chèn vào giữa mạch : cung (x_m, x_{k+1}) và cung (x_{k+1}, x_{m+1}) , bỏ cung (x_m, x_{m+1})

IV / Bài tập cơ bản :

1) Cho đồ thị vô hướng

Câu a) Tìm các cầu của đồ thị .

Câu b) Hãy kiểm tra xem :

b1 - Có phải là đồ thị nửa Euler không ? Nếu là đồ thị nửa Euler thì hiện đường đi Euler

b2 - Có phải là đồ thị Euler không ? Nếu là đồ thị Euler thì hiện chu trình Euler.

2) Cho đồ thị có hướng . Tìm mạch Haminton nếu có .

Bài 1 :

Uses Crt;

Const Max = 100;

Fi = 'cau.inp';

Fo = 'cau.out';

Type Mang = Array[1..Max, 1..max] of Integer;

T_Q = Array[1..Max*max] of Byte;

T_D = Array[1..Max] of Integer;

Var A : Mang;

```

    N,sv   : Byte;
    Q       : T_Q;
    D       : T_D;
    F       : Text;
Procedure MoFGhi;
Begin
    Assign(F,Fo);
    Rewrite(F);
End;
Procedure DocF;
Var F      : Text;
    i,j     : Byte;
Begin
    Assign(F,Fi);
    Reset(F);
    Readln(F,n);
    For i:=1 to n do
        Begin
            For j:=1 to n do Read(F,A[i,j]);
            Readln(F);
        End;
    Close(F);
End;
Procedure HienF;
Var i,j : Byte;
Begin
    For i:=1 to n do
        Begin
            For j:=1 to n do Write(A[i,j]:2);
            Writeln;
        End;
    End;
Procedure Loang(i : Byte);
Var    dau,cuoi,j,k    : Byte;
Begin
    cuoi := 0;
    dau  := 0;
    Inc(cuoi);
    Q[cuoi] := i;
    D[i]    := sv;
    While (dau+1<=cuoi) do
        Begin
            Inc(dau);
            j := Q[dau];
            For k:=1 to N do
                If (D[k]=0) and (A[j,k]=1) then
                    Begin
                        Inc(cuoi);
                        Q[cuoi] := k;
                        D[k]    := sv;
                    End;
            End;
        End;
End;
Function stplt : Integer;
Var    i,j    : Byte;

```

```

Ok      : Boolean;
Begin
  sv := 0;
  FillChar(D,sizeof(D),0);
  Repeat
    Ok := True;
    i := 0;
    For j:=1 to n do
      If D[j]=0 then
        Begin i := j;Break;End;
    If i>0 then
      Begin
        Inc(sv);
        Loang(i);
        Ok := False;
      End;
  Until Ok;
  stplt := sv;
End;
Procedure Cau;
  Var i,j      : Byte;
      s,s2     : Integer;
Begin
  Writeln(F,'Cac cau cua do thi : ');
  s := stplt;
  For i:=1 to n do
    For j:= 1 to n do
      If (A[i,j]=1) then
        Begin
          A[i,j] := 0;
          s2 := stplt;
          If s2 = s+1 then
            Writeln(F,('i:2,',j:2,));
          A[i,j] := 1;
        End;
  End;
End;
Function Sobacle : Integer;
  Var i      : Byte;
      sbl     : Integer;
Function Bac(i : Byte) : Integer;
  Var j,b : Integer;
  Begin
    b := 0;
    For j:=1 to n do Inc(b,A[i,j]);
    Bac := b;
  End;
Begin
  Sbl := 0;
  For i:=1 to n do
    If (Bac(i) mod 2 = 1) then Inc(sbl);
  Sobacle := sbl;
End;
Procedure ChutrinhEuler;
  Var i,j,dem : Byte;
      Lt      : Integer;

```

```

    chtr    : Array[1..Max] of Byte;
    Ok      : Boolean;
Function Ketthuc : Boolean;
Var        i,j : Byte;
Begin
    For i:=1 to n do
        For j:=i+1 to n do
            If A[i,j]=1 then
                Begin
                    Ketthuc := False;
                    Exit;
                End;
            Ketthuc := True;
        End;
    End;
Begin
    FillChar(chtr,Sizeof(chtr),0);
    i := 1;
    dem := 1;
    chtr[dem] := i;
    Lt := 1;
    Repeat
        Ok := False;
        j := 1;
        While (j<=n) do
            Begin
                If A[i,j]=1 then
                    Begin
                        A[i,j] := 0; { xoa canh }
                        A[j,i] := 0;
                        If stplt=Lt then { da xoa dung canh khong la cau }
                            Begin
                                Inc(dem);
                                chtr[dem]:= j;
                                i := j;
                                Ok := True;
                                Break;
                            End
                        Else { da xoa nham canh la cau, phai xay lai canh }
                            Begin
                                A[i,j] := 1;
                                A[j,i] := 1;
                            End;
                        End;
                    End;
                Inc(j);
            End;
        If Not Ok then
            { Phai qua cau, sang vung lien thong moi }
        Begin
            For j:=1 to n do { Tim lai cau de qua }
                If A[i,j]=1 then
                    Begin
                        A[i,j] := 0; { Qua cau }
                        A[j,i] := 0;
                        Inc(dem);
                        chtr[dem] := j;
                    End;
            End;
        End;
    End;
End;

```

```

        i      := j;
        Inc(Lt); { Gian tiep xoa diem co lap moi }
        Break;  { Thoat sang vung moi thi quay ve B2 }
    End;
End;
Until Ketthuc;
Writeln(F,'Chu trình Euler : ');
For i:=1 to dem-1 do Write(F,chtr[i]:2,' ->');
Writeln(F,chtr[dem]:2);
End;

Procedure Phanloai;
Var sbl : Integer;
Begin
    If stplt>1 then Writeln(F,'Do thi khong lien thong ')
    Else
        Begin
            sbl := sobacle;
            If sbl=0 then
                Begin
                    Writeln(F,'Do thi Euler ');
                    ChutrinhEuler;
                End
            Else
                If sbl=2 then Writeln(F,'Do thi nua Euler ')
                Else
                    Writeln(F,'Do thi lien thong , khong Euler , khong nua Euler ');
        End;
    End;
End;
BEGIN
    Clrscr;
    DocF;
    MoFghi;
    Cau;
    Phanloai;
    Close(F);
END.

```

Bài 2 :

```

Uses  Crt;
Const Max  = 20;
      Fi    = 'HMT.inp';
      Fo    = 'HMT.out';
Type  M1    = Array[1..Max,1..Max] of 0..1;
      M2    = Array[1..max] of Byte;
      M3    = Array[1..Max] of Boolean;
Var    A     : M1;
      KQ    : M2;
      KT    : M3;
      N     : Integer;

Procedure DocF;
Var    i,j   : Byte;
      F     : Text;

```

```

Begin
  Assign(F,Fi);
  Reset(F);
  Read(F,N);
  For i:=1 to N do
    Begin
      For j:=1 to N do Read(F,A[i,j]);
      Readln(F);
    End;
  Close(F);
End;
Function Ra(i : Byte) : Boolean;
Var      j      : Byte;
Begin
  Ra := True;
  For j:=1 to n do
    If KT[j] and (A[i,j]=1) then Exit;
  Ra := False;
End;
Function Vao(i : Byte) : Boolean;
Var      j      : Byte;
Begin
  Vao := True;
  For j:=1 to n do
    If KT[j] and (A[j,i]=1) then Exit;
  Vao := False;
End;
Procedure HienKQ;
Var      j      : Byte;
        F      : Text;
Begin
  Assign(F,Fo);
  Rewrite(F);
  Writeln(F,'Mach Haminton : ');
  For j:=1 to N do Write(F,KQ[j]:4);
  Close(F);
End;
Procedure Lam;
Var      Ok      : Boolean;
        i,d,c    : Byte;
Procedure Tim (i,d : Byte);
Var      j      : Byte;
Begin
  If d=c then
    Begin
      HienKq;
      Halt;
    End
  Else
    For j:=1 to N do
      If KT[j] and (A[i,j]=1 ) then
        Begin
          KT[j] := False;
          KQ[d] := j;
          Tim(j,d+1);
        End;
    End;
End;

```

```

    KT[j] := True;
End;
End;
Begin
    FillChar(KT,Sizeof(KT),True);
    OK := True;
    d := 0;
    c := N+1;
    While OK do
        Begin
            Ok := False;
            For i:=1 to N do { Tim dau mach }
                If KT[i] and (Not Vao(i)) and (Ra(i)) then
                    Begin
                        Ok := True;
                        KT[i] := False;
                        Inc(d);
                        Kq[d] := i;
                    End
                Else { Tim cuoi mach }
                    If KT[i] and (Vao(i)) and (Not Ra(i)) then
                        Begin
                            Ok := True;
                            KT[i] := False;
                            Dec(c);
                            Kq[c] := i;
                        End
                    End;
            If d=0 then Tim(1,1) { Tiep tuc tim tu dau mach }
            Else
                Tim(Kq[d],d+1); { Tiep tuc tim tu giua mach }
        End;
    BEGIN
        Repeat
            Clrscr;
            DocF;
            Lam;
            Writeln('Khong ton tai mach Haminton ! . An phim ESC : thoat ');
        Until ReadKey=#27;
    END.

```

BÀI TẬP

1) Tìm mạch Euler trong đồ thị có hướng, liên thông, tựa đối xứng.

2) Trong một nhà máy hoá chất, chỉ dùng 1 thiết bị sản xuất (thí dụ nh- : lò phản ứng hoá chất) để lần lượt điều chế N hoá chất, mỗi lần chuyển từ công việc điều chế hoá chất H_i sang điều chế hoá chất mới là H_k , phải điều chỉnh lại thiết bị sản xuất cho phù hợp điều chế hoá chất mới. Gọi chi phí điều chỉnh từ H_i sang H_k là P_{ik} . Giả sử chi phí điều chỉnh P_{ik} chỉ nhận giá trị 0, 1 với ý nghĩa: $P_{ik}=0$ nếu không phải điều chỉnh, $P_{ik}=1$ nếu phải điều chỉnh. Hãy tìm một quy trình sản xuất, để sản xuất đủ N hoá chất, mỗi hoá chất 1 lần, mà không tốn chi phí điều chỉnh thiết bị sản xuất.

3) Một nhà máy in sử dụng 2 máy A và B để hoàn thành N cuốn sách: Máy A in sách, máy B đóng sách. Thời gian làm cuốn sách k trên máy A và B tương ứng là a_k và b_k ($k=1..n$) với điều kiện phải qua máy A rồi mới qua máy B (in cuốn sách k xong rồi mới đóng nó). Ng-ời ta chứng minh đ-ợc định lý sau: Nếu $\min\{a_k, b_m\} \leq \min\{a_m, b_k\}$ thì phải làm cuốn sách k trước cuốn m.

Hãy tìm một trình tự in sách để tổng thời gian chờ đợi của máy B là ít nhất.

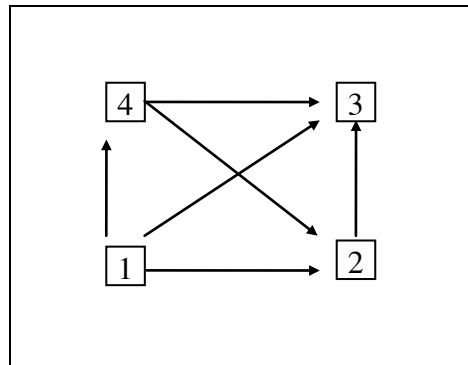
Gợi ý: Mỗi cuốn sách là 1 đỉnh đồ thị, thứ tự in là cung.

Từ bảng A, B, dựa vào định lý trên, lập đồ thị G, cung (k, m) thể hiện cuốn sách k làm trước cuốn sách m.

Vì phải hoàn thành toàn bộ các cuốn sách nên ta phải tìm mạch Hamilton của đồ thị.

Thí dụ:

T/T	A	B
1	0.5	1
2	2	1.5
3	1.5	1
4	2	3



$\min(a_1, b_4) = 0.5$ $\min(a_4, b_1) = 1$ Do đó sách 1 làm trước sách 4

Đáp số: Thứ tự làm các cuốn sách theo mạch Hamilton:

4) Tìm xâu nhị phân dài nhất mà mọi xâu con gồm k ký tự liên tiếp của nó chỉ xuất hiện đúng 1 lần

Gợi ý: Bài toán tìm mạch Euler, tạo đồ thị gồm 2^{k-1} đỉnh là các xâu nhị phân gồm k-1 ký tự 0, 1; các cung là xâu nhị phân k ký tự đ-ợc lập theo quy tắc:

Nếu cung (i, j) là xâu ($a_1 a_2 \dots a_{k-1}, a_k$) thì đỉnh i là xâu ($a_1 a_2 \dots a_{k-1}$), đỉnh j là xâu ($a_2 a_3 \dots a_k$)

Thí dụ: cung (i, j) = 0001 thì đỉnh i là 000, đỉnh j là 001.

Do đồ thị liên thông và giả đối xứng nên tồn tại mạch Euler, từ đó theo mạch tạo đ-ợc xâu nhị phân thoả mãn đề bài (xâu này dài 2^k ký tự)

Chú ý: Để giải bài toán 3 (N chi tiết máy trên 2 máy) còn thuật toán JonhSon

Tên chi tiết	1	2	3	4
Thời gian trên máy A	0.5	2	1.5	2
Thời gian trên máy B	1	1.5	1	3
Thứ tự thực hiện các chi tiết	1	4	2	3

Tìm giá trị nhỏ nhất trong tất cả các giá trị thời gian thực hiện trên máy A , máy B của các chi tiết còn lại , nếu giá trị nhỏ nhất này thuộc về máy A thì xếp tiếp tên chi tiết máy vào đoạn đầu hành trình , ngược lại nếu thuộc về máy B thì xếp tiếp tên chi tiết máy vào phần cuối hành trình , sẽ được kết quả là dòng 4 trong bảng trên : $1 \rightarrow 4 \rightarrow 2 \rightarrow 3$

5) Cho đồ thị có hướng, liên thông , tựa đối xứng , trên mỗi cung (i,k) có trọng số C_{ik} là chi phí từ đỉnh i tới đỉnh k . Tìm mạch Hamilton có tổng chi phí là ít nhất .

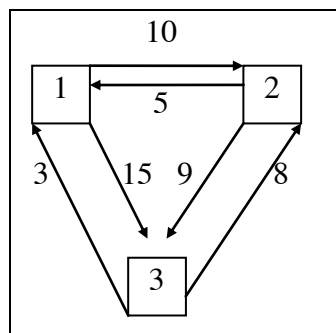
Gợi ý : Dùng phương pháp quy hoạch động : Giải bài toán kích cỡ lớn dựa vào bài toán tương tự nhỏ hơn bằng công thức sau :

$$G(i,T) = \text{Min} \{ C_{ik} + G(k, T-[k]) \}$$

i : đỉnh cuối của hành trình trong giai đoạn đang tìm đỉnh k tiếp theo , T : tập đỉnh còn lại chưa qua .

Theo công thức này, ta tìm được $G(1, T-[1])$ nếu biết $G(k, T-[1,k]) \forall k \in T-[1]$, để biết $G(k, T-[1,k])$ ta lại tìm $G(j, T-[1,k,j])$ quá trình tiếp tục cho đến khi đỉnh cuối cùng của hành trình là đỉnh i và tập các đỉnh còn lại là tập \emptyset , khi đó ta quy ước $G(i, \emptyset)$ là C_{ii} vì tới đỉnh cuối cùng là i thì chỉ còn cạnh $(i,1)$ chưa qua .

Thí dụ :



Ma trận $C(3,3)$

	0	10	15
5	0	9	
3	8	0	

$$G(2, \emptyset) = 5 ; G(3, \emptyset) = 3$$

$$G(2, [3]) = C_{23} + G(3, \emptyset) = 12; G(3, [2]) = C_{32} + G(2, \emptyset) = 13$$

$$G(1, [2,3]) = \text{Min} \{ C_{12} + G(2, [3]), C_{13} + G(3, [2]) \} = 22$$

Đường đi : $1 \rightarrow 2 \rightarrow 3 \rightarrow 1$

Bài 1) Lời giải Lê Hồng Việt (11 CT 1997-98) :

{ \$A+,B-,D+,E+,F-,G-,I-,L+,N-,O-,P-,Q-,R-,S+,T-,V+,X+}
{ \$M 16384,0,655360}

Program MachEuler;

Uses crt;

Const Max = 100;

Fi = 'Euler.inp';

Fo = 'Euler.out';

Type Mtk = Array[1..max,1..max] of 0..1;

MQ = Array[1..max] of byte;

Mdd = Array[1..max+1] of boolean;

Mkq = Array[1..max] of record d,c : Byte; end;

Msc = Array[1..max] of byte;

```

Var    A      : Mtk;
      N,maxkq  : Byte;
      Kq      : Mkq;
      Sc      : Msc;

```

```

Procedure Docf;

```

```

Var    F      : Text;
      i,j     : Byte;

```

```

Begin

```

```

  Assign(F,Fi);

```

```

  Reset(F);

```

```

  If Ioresult<>0 then

```

```

    Begin

```

```

      Writeln('Loi file hoac khong tim thay file ',Fi );

```

```

      Readln;

```

```

      Halt;

```

```

    End;

```

```

  Readln(f,n);

```

```

  For i:=1 to n do

```

```

    Begin

```

```

      For j:=1 to n do

```

```

        Begin

```

```

          Read(f,a[i,j]);

```

```

          If A[i,j]=1 then inc(sc[i]);

```

```

        End;

```

```

      Readln(f);

```

```

    End;

```

```

  Close(f);

```

```

end;

```

```

Function Slt:byte;

```

```

Var    Q      : Mq;

```

```

      Dx      : Mdd;

```

```

      d,c,i,j,lt: Byte;

```

```

      TT      : Boolean;

```

```

Begin

```

```

  Lt:=0;

```

```

  TT:=false;

```

```

  Fillchar(dx,sizeof(dx),false);

```

```

  i:=1;

```

```

  Repeat

```

```

    i:=1;

```

```

    While dx[i] do inc(i);

```

```

    If i>n then tt:=true;

```

```

    If not tt then

```

```

      Begin

```

```

        D:=0;c:=1;q[c]:=i;dx[i]:=true;

```

```

        While D<c do

```

```

          Begin

```

```

            Inc(d);

```

```

            For i:=1 to n do

```

```

              If ((a[q[d],i]=1) or (A[i,q[d]]=1) ) and (not dx[i]) then

```

```

                Begin

```

```

                  Inc(c);

```

```

                  Q[c]:=i;

```

```

                Dx[i]:=true;
            End;
        End;
        Inc(lt);
    End;
    Until tt=true;
    Sl:=lt;
end;

Function Euler:boolean;
Var i,j,va,ra:byte;
Begin
    Euler:=false;
    If slt<>1 then exit;
    For i:=1 to n do
        Begin
            Ra:=0;Va:=0;
            For j:=1 to n do
                Begin
                    If a[i,j]=1 then inc(ra);
                    If a[j,i]=1 then inc(va);
                end;
            If Ra<>va then exit;
        End;
    Euler:=true;
End;

Function Con:boolean;
Var i,j:byte;
Begin
    Con:=true;
    For i:=1 to n do
        For j:=1 to n do
            If A[i,j]=1 then exit;
        Con:=false;
    end;

Procedure TimMachEuler;
    Var    i,j,dd,llt,li1,lj1 : Integer;
          Tt,tt1 : Boolean;
Begin
    Dd:=0;
    I:=1;
    Llt:=slt;
    While con do
        Begin
            j:=1;
            Repeat
                While j<=n do
                    If (a[i,j]=1) { or (a[j,i]=1) } then
                        Begin
                            a[i,j]:=0;
                            If (sLt<>llt) then
                                Begin
                                    li1:=i;

```

```

        lj1:=j;
        A[i,j]:=1;
        inc(dd);
        inc(j);
    End
Else
    Begin
        inc(maxKq);
        Kq[maxkq].D:=i;
        Kq[maxkq].C:=j;
        Dec(sc[i]);
        i:=j;
        j:=1;
        dd:=0;
        Break;
    End;
End
Else inc(j);
If dd>=sc[i] then
    Begin
        i:=li1;
        j:=lj1;
        inc(maxKq);
        Kq[maxkq].D:=i;
        Kq[maxkq].C:=j;
        Dec(sc[i]);
        A[i,j]:=0;
        Dec(sc[i]);
        i:=j;
        llt:=slt;
        If i=1 then break;
        dd:=0;
    End;
    j:=1;
Until (dd=sc[i])
End;
End;

```

Procedure Hien;

```

Var    F      : Text;
      i      : Integer;

```

```

Begin
    Assign(f,fo);
    Rewrite(f);
    For i:=1 to maxkq do
        Writeln(f,kq[i].d:4,kq[i].c:4);
    Close(F);
end;

```

BEGIN

```

    Clrscr;
    DocF;
    If not Euler then
        Begin
            Writeln('Do thi khong phai Euler');

```

```

    Readln;
    Halt;
End;
TimMachEuler;
Hien;
END.

```

Bài 3) Giải bằng thuật toán JonhSon :

{ \$A+,B-,D+,E+,F-,G-,I-,L+,N-,O-,P-,Q-,R+,S+,T-,V+,X+ }
 { \$M 16384,0,655360 }

```

Program Js;
Uses crt;
const max=100;
    Fi='Johnson.inp';
    Fo='Johnson.out';

```

```

Type mang=array[1..2,1..max] of Real;
    MKq=array[1..max] of Byte;
    Mdx=array[1..max] of boolean;

```

```

Var A:mang;
    Kq:Mkq;
    Dx:Mdx;
    N:byte;

```

```

Procedure DocF;
Var f:text;
    i,j:byte;
Begin
    Assign(f,fi);
    reset(f);
    Readln(f,n);
    For j:=1 to n do
        begin
            For i:=1 to 2 do
                Read(f,a[i,j]);
            readln(f);
        end;
    close(f);
end;

```

```

Function Min(var p:Byte): Byte;
Var i,j,lc:byte;Lgt : Real;
Begin
    Lgt:=MaxInt;
    For i:=1 to 2 do
        For j:=1 to n do
            If (a[i,j]<lgt) and not dx[j] then
                Begin
                    P:=i;
                    lc:=j;
                    Lgt:=a[i,j];
                end;
        Min:=lc;
    end;
end;

```

```

Procedure Xepmay;
Var i,j,d,c,dem:byte;
Begin
  Fillchar(Dx,sizeof(dx),false);
  D:=0;C:=n+1;
  repeat
    j:=min(i);
    If i=1 then
      Begin
        Inc(d);
        Kq[d]:=j;
        Dx[j]:=true;
      end
    else
      Begin
        dec(c);
        Kq[c]:=j;
        Dx[j]:=true;
      end;
  until d=c-1;
end;

```

```

Procedure Hien;
Var f:text;
  i:byte;
Begin
  Assign(f,fo);
  rewrite(f);
  For i:=1 to n do
    Write(f,Kq[i]:4);
  close(f);
end;

```

```

BEGIN
  Clrscr;
  DocF;
  Xepmay;
  Hien;
END.

```

Bài 4)

Cách 1 : áp dụng bài tìm mạch Euler (bài 1) cho đồ thị có $(1 \text{ shl } (n-1)).(1 \text{ shl } (n-1))$ đỉnh đ-ợc xây dựng nh- đã nêu ở phần h- ớng dẫn ngay sau đề bài .

Cách 2 : Đệ quy xây dựng dãy nhị phân X gồm $2^n + n - 1$ số 0,1 :
+ n phần tử đầu là 0

+ phần tử thứ i ($n+1 \leq i \leq 2^n + n - 1$) chọn trong 2 khả năng 0,1 sao cho dãy :

$X[i-n+1], X[i-n+2], \dots, X[i]$ là 1 dãy nhị phân có n phần tử ch- a có mặt lần nào kể từ vị trí 1 tới i .

Cách 3 : Nh- cách 2 , nh- ng dùng vòng lặp thay đệ quy .

Cách 1 ch- ơng trình chỉ chạy tới $N=7$

Cách 2 ch- ơng trình chỉ chạy tới $N=10$

Cách 3 ch- ơng trình có thể chạy tới $N=15$

Lời giải bài 4 (cách 2) :

```
{ $A+,B-,D+,E+,F-,G-,I+,L+,N-,O-,P-,Q-,R+,S+,T-,V+,X+ }
{ $M 16384,0,655360 }
{ Cách giải đệ quy , xây dựng cấu trúc phân dải ( $2^n + N - 1$ ) thỏa mãn yêu cầu đề bài. }
```

```
Uses    Crt;
Const   Max    = 1 Shl 10;
        Output = 'MachOle.dat';

Type    Mang    = Array[0..max] of Shortint;
        TroM    = ^Mang;
Var      A,Dd    : TroM;
        N       : Byte;
        F       : Text;
        i       : Integer;

Procedure Nhap;
Begin   Write('Nhap N : '); Readln(N); End;

Function Tinh(k : Word) : Word;
Var     x,i : Integer;
Begin
    x:=0;
    For i:=k Downto k-N+1 Do
        If A^[i]=1 then      x:=x or (1 Shl (k-i));
    Tinh:=x;
End;

Procedure GhiF;
Begin
    Assign(f,Output); Rewrite(F);
    Writeln(F,'Do dai cua xau : ',1 Shl N+N-1 );
    For i:=1 to 1 Shl N+N-1 do Write(F,A^[i]);
    Writeln(F);
    Close(f);
    Halt;
End;

Procedure Xaydung(i : Integer);
Var     j      : Byte;
        gt     : Integer;
Begin
    If i>((1 SHL N)+N-1) then GhiF
    Else
        For j:=0 to 1 do
            If A^[i]=-1 then
                Begin
                    A^[i] := j;
                    GT := Tinh(i);
                    Inc(DD^[GT]);
                    If DD^[GT]=1 then Xaydung(i+1);
                    Dec(DD^[GT]);
                    A^[i] := -1;
                End;
        End;
End;
```

```
BEGIN
    Clrscr;
    New(A);
    New(DD);
    Nhap;
    Fillchar(A^,Sizeof(A^),0);
    For i:=N+1 to 1 Shl N+N-1 do A^[i]:=-1;
    Fillchar(DD^,Sizeof(DD^),0);
    DD^[0] := 1;
    Xaydung(N+1);
    Dispose(A);
    Dispose(DD);
END.
```

Sau đây là ch- ơng trình giải bài 4 (cách 3) : (Lời giải Lê Sỹ Vinh - 12 CT 1997-1998)

```
{ $A+,B-,D+,E+,F-,G-,I+,L+,N-,O-,P-,Q-,R+,S+,T-,V+,X+ }
{$M 16384,0,655360}
```

```
Uses  Crt;
Const Max   = 1 Shl 14+15;
       Output = ";
```

```
Type  Mang  = Array[0..max] of Shortint;
```

```
Var    A,Dd   : Mang;
       N      : Byte;
       F      : Text;
```

```
Procedure Nhap;
Begin
    Write('Nhap K : '); Readln(N);
End;
```

```
Function Tinh(k : Word) : Word;
Var x,i : Word;
Begin
    x:=0;
    For i:=k downto k-N+1 do
        If A[i]=1 then x:=x or (1 Shl (k-i));
    Tinh:=x;
End;
```

```
Procedure Working;
Var  i, Gt   : Word;
     F       : Text;
Begin
    Fillchar(dd,Sizeof(dd),0);
    Fillchar(A,Sizeof(a),0);
    For i:=N+1 to 1 Shl N+N-1 do A[i]:=-1;
    Dd[0]:=1;
    i:=N+1;
    While i<=1 Shl N+N-1 do
        Begin
            If A[i]=1 Then
```



```

      Begin
        A[i]:=-1; Dec(i);
      End
    Else
      Begin
        If A[i]>-1 then Dec(Dd[Tinh(i)));
        A[i]:=A[i]+1;
        Gt:=Tinh(i);
        Inc(dd[Gt]);
        If dd[Gt]<=1 then i:=i+1;
      End;
    End;
  Assign(f,Output); Rewrite(F);
  WRiteln(F,1 Shl N+N-1 );
  For i:=1 to 1 Shl N+N-1 Do Write(F,A[i]);
  Close(f);
End;

BEGIN
  Clrscr;
  Nhap;
  Working;
END.

```

Bài 5 :

Sau đây là 2 cách giải của Phạm phú Trung 11CT 1997-1998

Cách 1 : Độ quy (chỉ chạy với đồ thị số đỉnh nhỏ) .

Program Haminton;

Uses Crt;

Const Fi = 'Haminton.dat';

Fo = 'Vet.out';

max = 100;

Var A : Array [1..max,1..max] Of Integer;

TT : Array [1..max] Of 0..1;

Kq,Lkq : Array [1..max] Of Integer;

N : integer;

F : Text;

lt,t,cs : Integer;

Procedure Taofile;

Var i,j : Integer;

Begin

End;

Procedure Readfile;

Var i,j : Integer;

Begin

Assign(F,Fi);

Reset(F);

Readln(F,N);

For i:=1 to N do

```

Begin
  For j:=1 to N do
    Read(F,A[i,j]);
    Readln(F);
  End;
  Close(F);
End;

```

```

Procedure Hienfile;
  Var i,j : Integer;
  Begin
    Writeln('File');
    For i:=1 to N do
      Begin
        For j:=1 to N do
          Write(A[i,j]:4);
          Writeln;
        End;
      End;
    End;
  End;

```

```

End;

```

```

Procedure Init;
  Var i : Integer;
  Begin
    For i:=1 to N do TT[i]:=0;
    t:=0;
    lt:=maxint;
    cs:=1;
    Kq[1]:=1;
    TT[1]:=1;
  End;

```

```

Procedure Try(k : Integer);
  Var i : Integer;
  Begin
    For i:=1 to N do
      If (TT[i]=0) and (A[k,i]>0) then
        Begin
          t:=t+A[k,i];
          TT[i]:=1;
          Inc(cs);
          Kq[cs]:=i;
          If cs=N then
            Begin
              If t+A[Kq[cs],1]<lt then
                Begin
                  lt:=t+A[Kq[cs],1];
                  Lkq:=kq;
                End;
            End
          Else If cs<N then Try(i);
          t:=t-A[k,i];
          TT[i]:=0;
          Dec(cs);
        End;
      End;
    End;
  End;

```

End;

Procedure Inkq;

Var i : Integer;

Begin

Assign(F,Fo);

Rewrite(F);

Writeln(F,'Chi phi min la : ',lt);

For i:=1 to N do Write(F,Lkq[i]:4); Writeln(F,1:4);

Close(F);

End;

BEGIN

Clrscr;

Readfile;

Hienfile;

Init;

Try(1);

Inkq;

Writeln('Da xong ');

Readln;

END.

Cách 2 : Quy hoạch động (chạy đ- ọc đồ thị khoảng 60 đỉnh)

Program Haminton;

Uses Crt;

Const Fi = 'Haminton1.dat';

Fo = 'Haminton1.out';

max = 60;

Type Kmang = Record

ten,gt : integer;

TH : Set of 1..max;

End;

Var B : Array [1..max,1..max] Of Kmang;

A : Array [1..max,1..max] Of Integer;

N : Integer;

F : Text;

Procedure Taofile;

Var i,j : integer;

Begin

Randomize;

Write('Nhap N : ');

Readln(N);

For i:=1 to N do

For j:=1 to N do A[i,j]:=Random(10)+1;

For i:=1 to N do A[i,i]:=0;

Assign(F,Fi);

Rewrite(F);

Writeln(F,N);

For i:=1 to N do

Begin

For j:=1 to N do Write(F,A[i,j]:4);

Writeln(F);

```

    End;
    Close(F);
End;
Procedure Readfile;
    Var      i,j : Integer;
    Begin
        Assign(F,Fi);
        Reset(F);
        Readln(F,N);
        For i:=1 to N do
            Begin
                For j:=1 to N do Read(F,A[i,j]);
                Readln(F);
            End;
        Close(F);
    End;
Procedure Hien;
    Var      i,j : Integer;
    Begin
        Writeln('File');
        For i:=1 to N do
            Begin
                For j:=1 to N do Write(A[i,j]:4);
                Writeln;
            End;
        End;
Procedure Khoitao;
    Var      i,j : integer;
    Begin
        For j:=1 to N do
            Begin
                B[1,j].gt:=0;
                B[1,j].ten:=j;
                B[1,j].th:=[1..N]-[j];
            End;
        End;
Procedure Work;
    Var i,j,k,min,lk : Integer;
    Begin
        Khoitao;
        For i:=2 to N do
            For j:=1 to N do
                Begin
                    min:=maxint;
                    For k:=1 to N do
                        If (A[B[i-1,j].ten,k]>0) and (k in B[i-1,j].Th) then
                            If (A[B[i-1,j].ten,k]+B[i-1,j].gt<min) then
                                Begin
                                    lk:=k;
                                    min:=A[B[i-1,j].ten,k]+B[i-1,j].gt;
                                End;
                            B[i,j].gt:=min;
                            B[i,j].ten:=lk;
                            B[i,j].Th:=B[i-1,j].Th-[lk];
                        End;
                    End;
                End;
            End;
        End;
    End;

```

```

End;
Procedure Lannguoc;
  Var      min,i,j,lj : Integer;
  Begin
    min:=maxint;
    For j:=1 to N do
      If (A[B[N,j].ten,j]>0) and (A[B[N,j].ten,j]+B[N,j].gt<min) then
        Begin
          min:=A[B[N,j].ten,j]+B[N,j].gt;
          lj:=j;
        End;
    Assign(F,Fo);
    Rewrite(F);
    Writeln(F,'Chu trình haminton : ',min);
    For i:=1 to N do Write(F,B[i,lj].ten:4); Writeln(F,lj:4);
    Close(F);
    Writeln('Xem ket qua trong file ',fo );
  End;
BEGIN
  Clrscr;
  Taofile;
  Readfile;
  Hien;
  Work;
  Lannguoc;
  Readln;
END.

```

PHẦN 3 CÂY - CÂY KHUNG NGUYÊN NHẤT

I / Định nghĩa :

Cây là đồ thị hữu hạn , vô hướng , liên thông , không có chu trình , có ít nhất 2 đỉnh .

II / Tính chất :

1 - Định lý 1 :

Nếu H là cây có N đỉnh thì H có các tính chất sau đây :

- a) Thêm vào H một cạnh nối 2 đỉnh bất kỳ không kề nhau , H sẽ xuất hiện chu trình .
- b) Bớt đi 1 cạnh trong H thì H không liên thông
- c) Giữa 2 đỉnh bất kỳ của H luôn tồn tại 1 đường đi duy nhất (vậy H là đồ thị đơn)
- d) H có N-1 cạnh

2 - Định lý 2 :

Nếu đồ thị G liên thông có N đỉnh và N-1 cạnh thì G là cây .

Vậy cây là đồ thị liên thông có chu số bằng 0 (suy từ công thức Ôle)

3 - Ghi chú :

Từ 1 đồ thị có thể hình thành nhiều cây khác nhau (gọi là các cây khung của đồ thị) . Trong số các cây khung của đồ thị , có 1 cây đặc biệt tạo ra một cách đơn giản như sau : nối 1 đỉnh với n-1 đỉnh còn lại !

Số cây khung của 1 đồ thị đầy đủ là N^{n-2} (N số đỉnh)

Số cây khung của một đồ thị có hữu hạn đỉnh là một số hữu hạn ,nên luôn tìm được ít nhất 1 cây khung có tổng độ dài nhỏ nhất (nguyên lý biên) . Ta gọi cây khung này là cây khung ngắn nhất .

Bài toán tìm cây khung ngắn nhất là một bài toán gặp trong thực tế :

Thí dụ : Xây dựng mạng dây điện thoại nối N thành phố sao cho 2 thành phố bất kỳ liên lạc được với nhau và tổng độ dài dây điện ngắn nhất. Đó là bài toán tìm cây khung ngắn nhất. Ngược lại : Xây dựng mạng dây điện thoại nối N thành phố sao cho 2 thành phố bất kỳ liên lạc được với nhau và tổng độ tin cậy trên các đường dây điện là lớn nhất. Đó là bài toán tìm cây khung dài nhất.

III / Thuật toán Prim tìm cây khung nhỏ nhất :

B-ớc 1 : Khởi tri - Lấy 1 đỉnh i tùy ý đưa vào tập đỉnh của cây. Khi đó tập đỉnh của cây là $\mathcal{D} = \{i\}$. Tập cạnh của cây là $C = \emptyset$ (Tập rỗng)

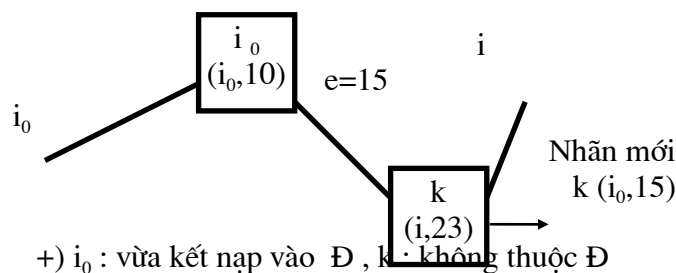
B-ớc 2 : Gán nhãn - Với mỗi đỉnh k không thuộc \mathcal{D} , ta gán cho nó nhãn $k(i, d)$ trong đó i là tên đỉnh thuộc \mathcal{D} , kề với k , gần k nhất, còn d là khoảng cách giữa i và k . Nếu trong \mathcal{D} không tìm được đỉnh i kề với k thì gán cho k nhãn $k(0, \infty)$.

B-ớc 3 : Kết nạp - Chọn đỉnh k không thuộc tập \mathcal{D} , có nhãn d nhỏ nhất, kết nạp k vào \mathcal{D} . Vậy $\mathcal{D} = \mathcal{D} + \{k\}$. Nhãn của k là $k(i, d)$ thì kết nạp cạnh (i, k) vào tập cạnh C . Vậy $C = C + \{\text{cạnh}(i, k)\}$. Gọi đỉnh k vừa kết nạp là i_0 .

Nếu số đỉnh của \mathcal{D} bằng N thì kết thúc, còn không chuyển sang b-ớc 4

B-ớc 4 : Sửa nhãn - Với mọi đỉnh k chưa thuộc \mathcal{D} có nhãn là $k(i, d)$ mà k kề với i_0 - là đỉnh vừa được kết nạp vào tập đỉnh ở b-ớc 3 - ta sửa lại nhãn của k theo nguyên tắc sau : Gọi độ dài cung (i_0, k) là e

Nếu $d > e$ thì đỉnh k có nhãn mới là $k(i_0, e)$



Procedure Prim(w,n,s)

$\{v(i)=1$ nếu đỉnh i được nạp vào cây, $v(i)=0$ nếu đỉnh i chưa được nạp vào mst $\}$

begin

for $i:=1$ to n do $v(i) := 0$

$v(s) := 1$ { đánh dấu đã nạp đỉnh s vào mst }

$E := \&$ { ban đầu tập cạnh của mst là rỗng }

for $i:=1$ to $n-1$ do { lần lượt đặt $n-1$ cạnh vào mst }

begin

min := :

for $j := 1$ to n do

if $v(j)=1$ then { j là đỉnh thuộc mst }

for $k:= 1$ to n do

if $v(k)=0$ and $w(j,k)<\text{min}$ then

begin

luuk := k

```

        e := (j,k)
        min := w(j,k)
    end
    v(luuk) := 1
    E := E U {e}
end
return(E)
end

```

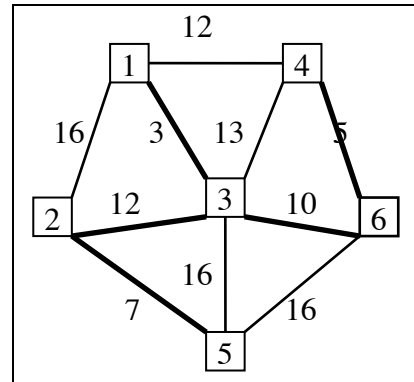
Thí dụ :

File dữ liệu vào : PRIM.INT

```

6
0 16 3 12 0 0
16 0 12 0 7 0
3 12 0 13 16 10
12 0 13 0 0 5
0 7 16 0 0 16
0 0 10 5 16 0

```



File dữ liệu ra : PRIM.OUT

(1, 3)= 3 (3, 6)= 10 (6, 4)= 5 (3, 2)= 12 (2, 5)= 7
 Tổng giá trị cây khung nhỏ nhất là 37

```

Uses Crt;
Const Fi = 'prim.txt';
      Fo = 'prim.out';
      Max = 200;
Var A : Array[1..Max,1..Max] of Byte;
    D : Array[1..Max] of Boolean;
    C : Array[0..Max] of record x1,x2 : Byte; end;
    Nh : Array[1..Max] of record truoc,giatri : Byte; end;
    N,dd,socanh : Byte;
    { canh : Integer; }
{-----}
Procedure DocF;
Var f : Text;
    i,j : Byte;
Begin
    Assign(f,fi);
    Reset(f);
    Readln(f,n);
    For i:=1 to n do
        Begin
            For j:=1 to n do read(f,a[i,j]);
            Readln(f);
        End;
    Close(f);
End;
{-----}
Procedure Napdinh1;
Begin
    Fillchar(d,sizeof(d),False);
    d[1] := True;
    dd := 1;
End;

```

```

{-----}
Function Min(xj : Byte): Byte;
Var   xi,p,i : Byte;
Begin
  xi := 0; p := 255;
  For i:=1 to N do
    If d[i] then
      If (p>a[i,xj]) and (a[i,xj]>0) then
        Begin
          xi := i;   p := a[i,xj];
        End;
      Min := xi;
    End;
  End;
{-----}
Procedure Gannhan;
Var   xi,xj : Byte;
Begin
  For xj:=1 to N do
    If not d[xj] then
      Begin
        xi := Min(xj);
        If (xi>0) and (A[xi,xj]>0) then
          Begin
            nh[xj].truoc := xi;
            nh[xj].giatri:= A[xi,xj];
          End
        Else
          If xi=0 then
            Begin
              nh[xj].truoc := 0;
              nh[xj].giatri:= 255;
            End;
          End;
      End;
    End;
  End;
{-----}
Procedure Ketnapthem;
Var   p,j,xj : Byte;
Begin
  p := 255;
  For j:=1 to n do
    If not d[j] then
      Begin
        If (nh[j].giatri<p) then
          Begin
            xj := j;
            p := nh[j].giatri;
          End;
        End;
      End;
    d[xj] := True;
    Inc(socanh);
    c[socanh].x1 := nh[xj].truoc;
    c[socanh].x2 := xj;
    dd := xj;
  End;
{-----}

```


Procedure Suanhan;

Var xj : Byte;

Begin

For xj:=1 to N do

If (not D[xj]) and (A[xj,dd]>0) then

Begin

If Nh[xj].giatri>A[xj,dd] then

Begin

Nh[xj].truoc := dd;

Nh[xj].giatri:= A[xj,dd];

End;

End;

End;

{-----}

Procedure Hiencanh;

Var i,p : Byte;f : Text;

Begin

Assign(f,fo);

Rewrite(f);p:=0;

For i:=1 to n-1 do

Begin

p := A[c[i].x1,c[i].x2]+p;

Write(f>(' ,c[i].x1:2,',c[i].x2:2,')= ',A[c[i].x1,c[i].x2]:3,' ':3);

End;

Writeln(f);

Writeln(f,'Tong gia tri cay khung ngan nhat la ',p);

Close(f);

End;

{-----}

Procedure TT_Prim;

Var Ok : Boolean;

Begin

SoCanh := 0;

Fillchar(nh,sizeof(nh),0);

Napdinh1;

Gannhan;

Ok := False;

Repeat

Ketnapthem;

If Socanh=N-1 then Ok:= True

Else Suanhan;

Until Ok;

Hiencanh;

End;

{-----}

BEGIN

Clrscr;

DocF;

TT_Prim

END.

Chương trình viết thu gọn :

uses crt;

const max = 100;

fi = 'prim.inp';

```

fo      = 'prim.out';
type    m1    = array[1..max,1..max] of integer;
        m2    = array[1..max] of 0..1;
        cung  = record i,j : byte end;
        th    = array[1..max] of cung;
var      w      : m1;
        d      : m2;
        e      : th;
        n,s    : byte;
procedure docf;
  var f : text;
      i,j : byte;
  begin
    assign(f,fi);
    reset(f);
    read(f,n,s);
    for i:=1 to n do
      for j:=1 to n do w[i,j] := 1000;
      while not eof(f) do
        begin
          read(f,i,j,w[i,j]);
          w[j,i] := w[i,j];
        end;
      close(f);
    end;
procedure hienf;
  var i,j : byte;
  begin
    for i:=1 to n do
      begin
        for j:=1 to n do write(w[i,j]:5);
        writeln;
      end;
    end;
procedure prim;
  var i,j,k,lk : byte;
      c      : cung;
      min    : integer;
  begin
    for i:=1 to n do d[i] := 0;
    d[s] := 1;
    fillchar(e,sizeof(e),0);
    for i := 1 to n-1 do
      begin
        min := maxint;
        for j:=1 to n do
          if d[j] = 1 then
            for k:=1 to n do
              if (d[k]=0) and (w[j,k]<min) then
                begin
                  lk := k;
                  min := w[j,k];
                  c.i := j;
                  c.j := k;
                end;
      end;

```

```

    e[i] := c;
    d[lk] := 1;
  end;
end;
procedure hien cay;
  var i : byte;
  begin
    for i:=1 to n-1 do write(e[i].i,'-',e[i].j,' ');
  end;
BEGIN
  docf;
  clrscr;
  prim;
  hien cay;
END.

```

Ôn tập THỰC HÀNH

I / Định nghĩa đồ thị 2 phía , định nghĩa cặp ghép:

- a) Cho 2 tập điểm X và Y , tập cung E gồm các cung $e=(x,y)$ mà $x \in X, y \in Y$.
 Đồ thị $G(X \cup Y, E)$ đ-ợc gọi là đồ thị 2 phía .
- b) Tập M gồm các cung thuộc E của đồ thị 2 phía G nêu trên mà các cung này không có đỉnh nào chung thì tập M đ-ợc gọi là cặp ghép. Số cung của M gọi là lực l-ợng của cặp ghép .
 Sau đây là 2 bài toán th-ờng gặp :
 1 - Bài toán tìm cặp ghép M có lực l-ợng cực đại .
 2 - Bài toán tìm cặp ghép M sao cho tổng trọng số trên các cung của M có giá trị lớn nhất (hoặc nhỏ nhất) .

II / Bài toán tìm cặp ghép M có lực l-ợng cực đại :

Những cung đã đ-ợc nạp vào cặp ghép ta qui - ớc là cung tô đậm , những cung ch- a đ-ợc ghép là cung tô nhạt . Những mút của cung đậm là đỉnh đậm , những đỉnh còn lại là đỉnh nhạt .

a) Định lý : Cặp ghép M có lực l-ợng cực đại khi và chỉ khi trong M không tìm thấy đ-ờng đi từ 1 đỉnh nhạt của X tới 1 đỉnh nhạt của Y.

b) Thuật toán :

```

+ Xây dựng cặp ghép ban đầu ( một số cung nào đó )
+ Stop := False
+ While Not Stop do
  Begin
    + Tìm đ-ờng đi P từ đỉnh i là nhạt của X tới đỉnh k là nhạt của Y
    ( gọi là đ-ờng tăng cặp ghép )
    + Nếu thấy P thì tăng cặp ghép : thêm cung  $e=(i,k)$  của E vào M
    Else Stop := True;
  End

```

Về tổ chức dữ liệu :

Dùng 2 mảng A và B quản các đỉnh của đồ thị. Cung đậm của dây chuyền là (i,j) với đỉnh i đi-ọc quản trên mảng A, đỉnh j đi-ọc quản trên mảng B, sẽ đi-ọc biểu diễn bằng cách gán $A[i] = j$ và $B[j] = i$. Các đỉnh k quản trên mảng A nếu $A[k] \neq 0$ thì đỉnh k là đỉnh nhậ trên A, Các đỉnh k đi-ọc quản trên mảng B nếu $B[k] \neq 0$ thì đỉnh k là đỉnh nhậ trên B

Để biểu diễn h-ớng trên cung ta dùng mảng TR, thí dụ để ghi nhận có cung đi từ đỉnh i tới đỉnh j của dây chuyền ta gán $TR[j]=i$

III / Bài toán tìm cặp ghép M sao cho tổng trọng số trên các cung của M có giá trị nhỏ nhất (hoặc lớn nhất). Còn gọi là bài toán tìm cặp ghép tối - u .

Ph-ơng pháp 1 : Chỉ giải quyết số điểm của X bằng N và số điểm của Y cũng bằng N và trên các cung $e=(i,j)$ với $i \in X, j \in Y$ có một trọng số $C[i, j] > 0$. Cặp ghép gồm các cung đậm nối đủ N điểm của X với N điểm của Y (không có 2 cung đậm nào có đỉnh chung) đi-ọc gọi là cặp ghép đầy đủ .

Giả sử M là một cặp ghép đầy đủ trên đồ thị 2 phía $G(X \cup Y, E)$. Cặp ghép này có thể ch- a là cặp ghép tối - u . Từ đồ thị vô h-ớng G ta xây dựng đồ thị G_M có h-ớng nh- sau :

Trên cung tô đậm $e=(i,j) \in E \setminus M$ ($i \in X, j \in Y$), xác định cung (j,i) chiều từ j tới i , với trọng số bằng $-C[i, j]$. Trên các cung nhậ, xác định chiều từ X sang Y với trọng số nh- cũ .

a) Định lý : M là cặp ghép tối - u khi và chỉ khi trong G_M không có chu trình âm (tổng các trọng số trên các cung của chu trình là số âm)

Dựa vào định lý trên, ta có thể giải bài toán cặp ghép có tổng trọng số nhỏ nhất bằng thuật toán sau :

b) Thuật toán :

```

+ Xây dựng một cặp ghép đầy đủ M trên đồ thị 2 phía vô h-ớng G
+ Stop := False
+ While Not Stop do
  Begin
    + Xây dựng đồ thị có h-ớng  $G_M$  từ đồ thị vô h-ớng G
    + Tìm chu trình âm trên  $G_M$ 
    + Nếu có chu trình âm thì khử chu trình âm ( bằng cách đổi dấu các trọng
số của các cung của chu trình, sẽ có chu trình d-ơng )
    Else Stop := True
  End

```

Trong tr-ờng hợp cần tìm cặp ghép có tổng trọng số trên các cung là lớn nhất thì làm nh- hệt bài toán trên, song khi đọc mảng c-ớc phí $C[i,j]$ thì đổi lại dấu, đồng thời tổng trọng số tối - u cuối cùng cũng đổi lại dấu là xong .

Ph-ơng pháp 2 : (M thợ, N việc, $C[i,j]$ tiền do thợ i làm việc j có thể là số âm hoặc d-ơng)

Thuật toán tìm tổng trọng số trên cặp ghép lớn nhất :

Gọi tập đỉnh thợ là X, tập đỉnh công việc là Y .

Động tác 1 :

Xây dựng các hàm F_x, F_y sao cho $F_x[i] + F_y[j] \geq C[i,j]$ (i thuộc X, j thuộc Y) . Khởi trị các hàm F_x, F_y nhận giá trị ban đầu :

$F_x[i] = \max \{ C[i,j], \text{ với mọi } j \text{ thuộc } Y \}$ với mọi i thuộc X

$F_y[j] = 0$

Nh- vậy bảo đảm đi-ọc tính chất cung (i,j) thuộc cặp ghép thì $F_x[i] + F_y[j] = C[i,j]$

Động tác 2 : Tìm một đỉnh u thuộc tập X ch- a đ- ọc ghép cặp

Động tác 3 : Xây dựng đồ thị có h- ớng G1 (so dinh =M+N) theo quy cách là :

Nếu $F_i[i] + F_j[j] = C[i,j]$ nghĩa là có thể ghép (i,j) thì xác nhận có cung (i,M+j) trong G1

Động tác 4 : Tìm đ- ờng tăng cặp ghép (LOANG trên đồ thị G1)

Xuất phát từ một đỉnh u thuộc tập X ch- a đ- ọc ghép cặp , tìm dây chuyền tới một đỉnh v thuộc Y ch- a đ- ọc ghép cặp .

Động tác 5 : Tăng cặp ghép thực hiện khi trong động tác 4 tìm đ- ọc dây chuyền

Động tác 6 : Điều chỉnh lại các hàm F_x, F_y (gọi là sử nhân)

Tìm $d = \min(F_i[i] + F_j[j] - C[i,j])$

i thuộc tập X và đã xét , j thuộc tập Y và ch- a xét

Điều chỉnh lại :

$F_i[i] := F_i[i] - d$ Voi mọi i THUOC X DA xet (Neu tim MIN thi +d)

$F_j[j] := F_j[j] + d$ Voi mọi j THUOC Y DA xet (Neu tim MIN thi -d)

Cong viec nay giup ta tang duoc so canh cua do thi G

Neu ban dau co duong di tu $i \rightarrow j$ tuc la $F_i[i] + F_j[j] = C[i,j]$

thi dieu nay luon duoc bao dam vi $(F_i[i] - d) + (F_j[j] + d) = C[i,j]$

Mat khac sau khi giam $F_i[i]$ Voi mọi i Thuoc X da xet di d_{\min}

thi so canh cua do thi tang len ≥ 1 canh

Quay lại LOANG cho đến khi tìm được cách Ghep

BÀI TẬP

1) Một xí nghiệp có N công nhân , và dây chuyền sản xuất gồm N vị trí . Công nhân i nếu đứng ở vị trí j của dây chuyền thì tạo lãi C_{ij} . Hãy bố trí công nhân sao cho mỗi công nhân 1 vị trí và 1 vị trí chỉ có 1 công nhân mà tổng số lãi thu đ- ọc tốt nhất .

2)

a) Cho M ng- ời thợ , nhận làm N công việc ($M \leq N$), thợ i ($1 \leq i \leq M$) nếu làm việc j ($1 \leq j \leq N$) thì tạo lợi nhuận C_{ij} . Hãy sắp xếp sao cho M thợ làm đ- ọc nhiều lợi nhuận nhất (mỗi thợ chỉ làm 1 việc) .

b) Nh- trên nh- ng thay từ lợi nhuận bằng chi phí cho sản xuất , tìm sắp xếp M thợ làm sao cho chi phí ít nhất

3) Cho N thành phố . Khoảng cách giữa 2 thành phố là C_{ij} . Có K nhân viên tiếp thị hiện đang ở K thành phố trong N thành phố trên . Hãy chuyển K nhân viên tiếp thị này đến K thành phố mới trong N thành phố này sao cho tổng khoảng cách di chuyển là ít nhất .

INPUT

10 4

0	7	7	1	2	1	1	5	1	3
2	0	1	1	1	1	5	4	1	7
1	1	0	1	1	1	3	7	2	4
5	2	4	0	2	4	10	1	7	1
7	1	3	7	0	10	2	4	1	1
10	1	1	2	1	0	1	4	2	1

```

1 1 4 1 1 3 0 1 10 1
7 1 7 1 1 3 4 0 1 1
7 7 1 2 1 1 4 2 0 10
1 3 4 1 2 4 1 1 1 0
1 2 3 4
10 9 8 7

```

OUTPUT

```

5
1 7
2 9
3 4 8
4 10

```

BÀI CHÉA 2:

```

{$A+,B-,D+,E+,F-,G-,I+,L+,N-,O-,P-,Q-,R+,S+,T-,V+,X+}
{$M 16384,0,655360}
Program Cap_Ghep_Cuc_dai; { Do Duc Dong 11 CT Nguyen Hue 1998-1999 }

Uses Crt;

Const      Max      = 102;
           Fi       = 'cgm.i35';
           Fo       = 'cg.OUT';

Type       K1        = Array[1..Max,1..Max] of Integer;
           K2        = Array[1..Max] of Longint;
           K3        = Array[1..2*Max] of Byte;
           K4        = Array[1..Max] of Byte;

Var        C         : K1; {Mang Trong so}
           FX,FY     : K2; {Ham F Chap nhan duoc}
           Tr        : K3; {Mang Truoc}
           Dx,Dy,    {Danh dau dinh da xet tung phia}
           Right,Left: K4;{Cap ghep}
           M,N       : Byte;
           Ok        : Boolean;{Neu tim thay duong tang cap ghep thi Ok=True}

Procedure Input;
Var F      :Text;
    i,j    :Byte;
    Maxso  :Integer;
Begin
    Assign(F,Fi);
    Reset(F);
    ReadLn(F,M,N);

    For i:=1 to M do
        Begin
            Maxso:=-MaxInt;
            For j:=1 to N do
                Begin

```

```

        Read(F,C[i,j]);
        If C[i,j]>Maxso then Maxso:=C[i,j];
    End;
    FX[i]:=Maxso;{Xay dung F chap nhan duoc}
End;
FillChar(FY,Sizeof(FY),0);

Close(F);
End;

Procedure Thay_doi_lai_cac_cung(j :Byte);
{j dinh cuoi cung nam ben Y .Tang so cap ghep:cung dam->nhat,nhat->dam}
Var i      :Byte;
Begin
    Repeat
        i      := Tr[j];
        Right[i] := j-M;
        Left[j-M] := i;
        j      := Tr[i];
    Until j=0;
End;

Procedure Loang(i : Byte);
Var j,dau,cuoi    : Byte;
    D,Q           : K3;{Mang Q de loang}
Begin
    Ok:=False;
    FillChar(D,Sizeof(D),0);
    FillChar(Dx,Sizeof(Dx),0);
    FillChar(Dy,Sizeof(Dy),0);
    FillChar(Tr,Sizeof(Tr),0);
    FillChar(Q,Sizeof(Q),0);
    dau:=1;cuoi:=1;Q[1]:=i;D[i]:=1;
    Dx[i]:=1;{Danh dau dinh i ben Right da xet}
    While dau<=cuoi do
        Begin
            For j:=1 to M+N do
                If D[j]=0 then
                    Begin
                        If j>M then {Dinh o ben Left}
                            Begin{Dinh o ben Right}                {Chap nhan duoc}
                                If (Q[dau]<=M) And (FX[Q[dau]]+FY[j-M]=C[Q[dau],j-M]) then
                                    Begin
                                        Inc(cuoi);
                                        Q[cuoi]:=j;
                                        D[j]:=1;
                                        Tr[j]:=Q[dau];
                                        Dy[j-M]:=1;{Danh dau dinh ben Left da xet}
                                        If Left[j-M]=0 then {Dinh nay chua duoc ghep}
                                            Begin
                                                Ok:=True;
                                                Thay_doi_lai_cac_cung(j);
                                                Exit;
                                            End;
                                        End;
                                    End;
                                End;
                            End
                        Else
                            Begin{Dinh o ben Left}                {Dinh nay da duoc ghep voi j}
                                If (Q[dau]>M) And (Left[Q[dau]-M]=j) then
                                    Begin
                                        Inc(cuoi);
                                        Q[cuoi]:=j;
                                        D[j]:=1;
                                        Tr[j]:=Q[dau];
                                        Dx[j]:=1;{Danh dau dinh ben Right da xet}

```

```

        {Break;Vi chi co mot dinh di tu j}
      End;
    End;
  End;
  Inc(dau);
End;
End;

Function Min:Longint;
Var i,j   : Byte;
    Ph    : Integer;
Begin
  Ph:=MaxInt;
  For i:=1 to M do
    If Dx[i]=1 then {Dinh da xet ben X}
      For j:=1 to N do
        If Dy[j]=0 then {Dinh chua duoc xet ben Y}
          If FX[i]+FY[j]-C[i,j]<Ph then Ph:=FX[i]+FY[j]-C[i,j];
        Min:=Ph;
      End;
    End;

Procedure Thay_doi_lai_do_thi;{Tang so canh}
Var k      : Byte;
    d      : Integer;
Begin
  d:=Min;
  For k:=1 to M do
    If Dx[k]=1 then Dec(FX[k],d);

    For k:=1 to N do
      If Dy[k]=1 then Inc(FY[k],d);
    End;

Procedure Work;
Var k      : Byte;
Begin
  FillChar(Right,Sizeof(Right),0);
  FillChar(Left,Sizeof(Left),0);
  For k:=1 to M do
    If Right[k]=0 then{Tim dinh chua gep cap}
      Begin
        Ok:=False;
        While Ok=False do{Lam den khi ghep duoc}
          Begin
            LOANG(k);
            If Ok=False then Thay_doi_lai_do_thi;
            {Neu chua tim thay thi Left tang so canh}
          End;
        End;
      End;

    End;

  End;

End;

Procedure Output;
Var F      :Text;
    k      :Byte;
    chiphi : longint;
Begin
  Assign(F,Fo);
  Rewrite(F);
  chiphi := 0;
  For k:=1 to M do
    begin
      WriteLn(F,k,#32,Right[k]);
      chiphi := chiphi+ C[k,Right[k]];
    end;
  write(F,chiphi);

```



```
Close(F);
End;
```

```
BEGIN
  Input;
  Work;
  Output;
```

```
END.
```

```
DT2P.INP
```

```
DT2P.OUT
```

```
4 4
2 5 1 6
8 7 6 4
6 9 3 5
5 1 2 7
```

```
4 5
7 8 9 4 7
5 0 7 5 2
3 1 2 0 3
1 2 3 0 4
```

BÀI CHÈA 3 :

```
{ $A+,B-,D+,E+,F-,G-,I+,L+,N-,O-,P-,Q-,R+,S+,T-,V+,X+ }
{ $M 65384,0,655360 }
```

```
Uses Crt;
```

```
Const Max = 101;
      Input = 'bai1.inp';
      Output = 'bai1.out';
      MaxK = 51;
```

```
Type
```

```
  Mang = Array[1..Max,1..Max] of Integer;
  Bang = Array[1..MaxK,1..MaxK] of Integer;
```

```
Var
```

```
  C      : Mang;
  T      : Array[1..Max,1..Max] of Byte;
  N,K    : Byte;
  A      : Bang;
  Nhan   : Array[1..Max] of Integer;
  Ra,Vao,Cu,Moi,Truoc,IVao,Ng      : ArRay[1..Max] of Byte;
```

```
(*-----*)
```

```
Procedure Nhap;
```

```
  Var Inp : Text;
      i,j : Byte;
```

```
  Begin
```

```
    Assign(inp,input);
```

```
    Reset(inp);
```

```
    Readln(inp,N,K);
```

```
    For i:=1 to N do
```

```
      Begin
```

```
        For j:=1 to N do Read(inp,C[i,j]);
```

```

    Readln(inp);
    End;
    For i:=1 to N do C[i,i]:=0;
    For i:=1 to K do Read(inp,Cu[i]);
    Readln(inp);
    For i:=1 to K do Read(inp,Moi[i]);
    Close(inp);
    End;
(*-----*)

Procedure TinhCP; { Dung Ford-Bellman tinh duong di ngan nhat i-j }
    Var i,j,k : Byte;
    Begin
        Fillchar(T,sizeof(T),0);
        For k:=1 to N do
            For i:=1 to N do
                For j:=1 to N do
                    If C[i,k]+C[k,j]<C[i,j] then
                        Begin
                            C[i,j]:=C[i,k]+C[k,j];
                            T[i,j]:=k;
                        End;
                End;
            End;
        End;
(*-----*)

Procedure TaoMT; { Khoi tao do thi 2 phia vo huong E : k-k }
    Var i,j : Byte;
    Begin
        For i:=1 to K do
            For j:=1 to K do
                A[i,j]:=C[Cu[i],Moi[j]];
            End;
        End;
(*-----*)

Procedure NghiemDau; { Khoi tao do thi 2 phia co huong Em : k-k }
    Var i : Byte;
    Begin
        For i:=1 to k do
            Begin
                Ra[i] := i; { ghep i-i }
                Vao[i] := i;
                A[i,i] := -A[i,i];
            End;
        End;
(*-----*)

Procedure KhoiTao;
    Begin
        Fillchar(nhan,sizeof(nhan),0);
        Fillchar(Truoc,sizeof(truoc),0); { Luu 1 hanh trinh hien tai }
    End;
(*-----*)

Function CT_am(x:Byte):Boolean; { Tim chu trinh am }
    Var Luu : Byte;

```

```

Begin
  Luu:=x;
  Repeat
    Luu := Truoc[Luu];
    If Luu=0 then
      Begin
        CT_am:=false;
        Exit;
      End;
    Luu := Vao[Luu];
    If Luu=x then
      Begin
        CT_am:=true;
        Exit;
      End;
    Until false;
  End;
(*-----*)

```

[illegible]

```

Var      Luu,p  : Byte;
Begin
  LVao:=Vao;
  Luu:=x;
  Repeat
    { Doi dau trong so cac cung to net dam }
    p    := Truoc[Luu];
    A[Luu,p] := -A[Luu,p];

    Vao[p] := Luu;
    Ra[Luu] := p;

    { Doi dau trong so cac cung to net nhut }
    Luu    := LVao[p];
    A[Luu,p] := -A[Luu,p];
  Until Luu=x;
End;
(*-----*)

```

```

Function Tang:Boolean; { Tang them cap ghep moi }
Var
    Kethuc : Boolean;
    p,i,j   : Byte;

Begin
    KhoiTao;
    Repeat
        kethuc:=true; { Khong sua nhan duoc }
        For p:=1 to K do
            Begin
                j := Ra[p];
                For i:=1 to K do
                    If (i<>p) and
                        { Sua nhan tot hon }
                        (Nhan[i]>Nhan[p]+A[p,j]+A[j,i]) then
                            Begin

```

```

Nhan[i] := Nhan[p]+A[p,j]+A[j,i];
Truoc[i] :=j;
kethuc:=false; { Con sua nhan }

```

```

If CT_am(i) then
  Begin
    DoiDau(i);
    Tang:=true; { Con tang duoc }
    Exit;
  End;
End;

```

```

End;

```

```

End;

```

```

Until kethuc;

```

```

Tang:=false;

```

```

End;

```

```

(*-----*)

```

```

Procedure Hien;

```

```

  Var    i,j : Byte;

```

```

  Begin

```

```

    For i:=1 to K do

```

```

      Begin

```

```

        For j:=1 to K do Write(A[i,j]:3);

```

```

        Writeln;

```

```

      End;

```

```

    End;

```

```

(*-----*)

```

```

Function Tinh:Integer;

```

```

  Var    i,j : Byte;

```

```

        sum : Integer;

```

```

  Begin

```

```

    sum:=0;

```

```

    For i:=1 to K do

```

```

      For j:=1 to K do

```

```

        If A[i,j]<0 then Inc(sum,abs(A[i,j]));

```

```

      Tinh:=Sum;

```

```

    End;

```

```

(*-----*)

```

```

Procedure HienKQ;

```

```

  Var out : Text;

```

```

    i,j : Integer;

```

```

    dem : Byte;

```

```

Procedure Tim(x,y:Byte);

```

```

  Var Tg : Byte;

```

```

  Begin

```

```

    Tg := T[x,y]; { Lan nguoc theo cung trung gian - Ford Bellman }

```

```

    If Tg=0 then

```

```

      Begin

```

```

        If (dem=0) or ((dem>0) and (x<>Ng[dem])) then

```

```

          Begin

```

```

            Inc(dem);

```

```

            Ng[dem]:=x;

```

```

          End;

```

```

        Inc(dem);
        Ng[dem]:=y;
    End
Else
    Begin
        Tim(x,tg);
        Tim(tg,y);
    End;
End;
Begin
    Assign(out,output);
    Rewrite(out);
    Writeln(out,Tinh);
    For i:=1 to K do
        Begin
            dem:=0;
            Tim(Cu[i],Moi[Ra[cu[i]]]);
            { Xay dung Ng : duong di tu cu[i] toi moi[Ra[cu[i]]] }
            For j:=1 to dem do Write(out,ng[j],' ');
            Writeln(out);
        End;
    CClose(out);
End;
(*-----*)

```

```

Procedure Lam;
Begin
    TinhCP;
    TaoMT;
    NghiemDau;
    Repeat Until Not Tang;
    HienKQ;
End;
(*-----*)

```

```

Procedure Test;
Var i,j : Byte;
    inp : Text;
Begin
    Randomize;
    N:=10;
    k:=4;
    Assign(inp,input);
    Rewrite(inp);
    Writeln(inp,N,' ',K);
    For i:=1 to N do
        Begin
            For j:=1 to N do
                If i<>j then Write(inp,Random(4)*Random(4)+1:4)
                Else Write(inp,0:4);
            Writeln(inp);
        End;
    For i:=1 to K do Write(inp,i,' ');
    Writeln(inp);
    For i:=N downto N-k+1 do Write(inp,i,' ');

```

```
    Close(inp);  
    End;  
(*-----*)
```

```
BEGIN  
    Clrscr;  
    {Test;}  
    Nhap;  
    Lam;  
END.
```

Bài toán tìm cặp ghép với tổng trọng số lớn nhất :
{ \$A+,B-,D+,E+,F-,G-,I+,L+,N-,O-,P-,Q-,R+,S+,T-,V+,X+ }

{ \$M 16384,0,655360 }

Program Cap_Ghep_Cuc_dai;

Uses Crt;

Const Max =100;

Fv ='DT2P.INP';

Fr ='DT2P1.OUT';

Var C :Array[1..Max,1..Max]of Integer;{ Mang Trong so }

Fi,Fj :Array[1..Max]of Integer;{ Ham F Chap nhan duoc }

Tr,Q :Array[1..2*Max]of Byte;{ Mang Truoc,Mang Q de loang }

S,T :Array[1..Max]of Byte;{ Danh dau dinh da xet tung phia }

Trai,Phai :Array[1..Max]of Byte;{ Cap ghep }

M,N :Byte;

Ok :Boolean;{ Neu tim thay duong tang cap ghep thi Ok=True }

dau,cuoi :Byte;

Procedure Input;

Var F :Text;

i,j :Byte;

Maxso :Integer;

Begin

Assign(F,Fv);

Reset(F);

ReadLn(F,M,N);

FillChar(Fj,Sizeof(Fj),0);

For i:=1 to M do

Begin

Maxso:=-MaxInt;

For j:=1 to N do

Begin

Read(F,C[i,j]);

If C[i,j]>Maxso then Maxso:=C[i,j];

End;

Fi[i]:=Maxso;{ Xay dung F chap nhan duoc }

End;

Close(F);

End;

Procedure Thay_doi_lai_cac_cung(j :Byte);

{ Tang so cap ghep:cung dam->nhat,nhat->dam }

Var i :Byte;

Begin

Repeat

i:=Tr[j];

Trai[i]:=j-M;

Phai[j-M]:=i;

j:=Tr[i];

Until j=0;

End;

Procedure LOANG(i :Byte);

Var j :Byte;

D :Array[1..2*Max]of Byte;

```

Begin
  Ok:=False;
  FiLLChar(D,Sizeof(D),0);
  FiLLChar(S,Sizeof(S),0);
  FiLLChar(T,Sizeof(T),0);
  FiLLChar(Tr,Sizeof(Tr),0);
  FiLLChar(Q,Sizeof(Q),0);
  dau:=1;cuoi:=1;Q[1]:=i;D[i]:=1;
  S[i]:=1;{ Danh dau dinh i ben trai da xet }
  While dau<=cuoi do
    Begin
      For j:=1 to M+N do
        If D[j]=0 then
          Begin
            If j>M then { Dinh o ben Phai }
              Begin{ Dinh o ben Trai }      { Chap nhan duoc }
                If (Q[dau]<=M) And(Fi[Q[dau]]+Fj[j-M]=C[Q[dau],j-M]) then
                  Begin
                    Inc(cuoi);
                    Q[cuoi]:=j;
                    D[j]:=1;
                    Tr[j]:=Q[dau];
                    T[j-M]:=1;{ Danh dau dinh ben phai da xet }
                    If Phai[j-M]=0 then { Dinh nay chua duoc ghep }
                      Begin
                        Ok:=True;
                        Thay_doi_lai_cac_cung(j);
                        Exit;
                      End;
                    End;
                  End
                Else
                  Begin{ Dinh o ben Phai }    { Dinh nay da duoc ghep voi j }
                    If (Q[dau]>M) And (Phai[Q[dau]-M]=j) then
                      Begin
                        Inc(cuoi);
                        Q[cuoi]:=j;
                        D[j]:=1;
                        Tr[j]:=Q[dau];
                        S[j]:=1;{ Danh dau dinh ben trai da xet }
                        { Break;Vi chi co mot dinh di tu j }
                      End;
                    End;
                  End;
                Inc(dau);
              End;
            End;
          End;
        End;
      End;
    End;
  End;

Function Min:Integer;
Var i,j :Byte;
    Ph :Integer;
Begin
  Ph:=MaxInt;
  For i:=1 to M do
    If S[i]=1 then{ dinh da xet ben trai }

```



```

For j:=1 to N do
  If T[j]=0 then {dinh chua duoc xet ben phai}
    If Fi[i]+Fj[j]-C[i,j]<Ph then Ph:=Fi[i]+Fj[j]-C[i,j];
  Min:=Ph;
End;

```

```

Procedure Thay_doi_lai_do_thi; {tang so canh}

```

```

Var k   :Byte;
    dd  :Integer;
Begin
  dd:=Min;
  For k:=1 to M do
    If S[k]=1 then Dec(Fi[k],dd);

    For k:=1 to N do
      If T[k]=1 then Inc(Fj[k],dd);
  End;

```

```

Procedure Work;

```

```

Var k   :Byte;
Begin
  FiLLChar(Trai,Sizeof(Trai),0);
  FiLLChar(Phai,Sizeof(Phai),0);
  For k:=1 to M do
    If Trai[k]=0 then {Tim dinh chua gep cap}
      Begin
        Ok:=False;
        While Ok=False do {Lam den khi ghep duoc}
          Begin
            LOANG(k);
            If Ok=False then Thay_doi_lai_do_thi;
            {Neu chua tim thay thi phai tang so canh}
          End;
        End;
      End;
  End;

```

```

Procedure Output;

```

```

Var F   :Text;
    k   :Byte;
Begin
  Assign(F,Fr);
  ReWrite(F);
  For k:=1 to M do WriteLn(F,k,#32,Trai[k]);
  Close(F);
End;

```

```

BEGIN

```

```

  Input;
  Work;
  Output;
END.
DT2P.INP
DT2P.OUT
4 4
2 5 1 6

```

8 7 6 4

6 9 3 5

5 1 2 7

4 5

7 8 9 4 7

5 0 7 5 2

3 1 2 0 3

1 2 3 0 4

{ Thuật toán tìm cặp ghép cực đại:Lon nhất

M đỉnh voi N đỉnh($M \leq N$)

Trong số có thể tìm -> Một cách đơn giản để tìm cặp ghép Min là đổi dấu trong số $C[i,j] := -C[i,j]$ rồi tìm như cặp ghép cực đại. Cũng có một cách khác nữa để tìm cặp ghép min.

Gọi đồ thị bên Trái là :X

Gọi đồ thị bên Phải là :Y

Bước 1: Xây dựng hàm F_i, F_j chấp nhận được

$F_i[i] = \text{MAX}(C[i,j], \text{Voi mọi } j \text{ thuộc } Y) \text{ Voi mọi } i \text{ thuộc } X$
(Nếu tìm cặp ghép min thì $F_i[i] = \text{MIN}(C[i,j])$)

$F_j[j] = 0 \text{ Voi mọi } j \text{ thuộc } Y$

(F_j điều chỉnh sao cho phù hợp với F_i để ta luôn có
 $F_i[i] + F_j[j] \geq C[i,j]$)

Bước 2: Tìm một đỉnh thuộc tập X chưa được ghép cặp

Bước 3: Xây dựng đồ thị G (số đỉnh $= M+N$)

Nếu $F_i[i] + F_j[j] = C[i,j]$ thì có cung đi từ i -> (M+j)

Nếu $F_j[j] = i$ thì có cung đi từ (M+j) -> i

Bước 4: Tìm đường tăng cặp ghép (Dùng thuật toán LOANG với đồ thị G)

Xuất phát từ một đỉnh chưa được ghép cặp.

Nhưng đỉnh đã xét bên tập X ta sẽ đánh dấu bằng mảng S

Nhưng đỉnh đã xét bên tập Y ta sẽ đánh dấu bằng mảng T

Nếu LOANG thấy một đỉnh thuộc Y chưa ghép cặp thì tăng cặp ghép và thoát và Quay Về bước 2

Nếu không tìm thấy tức là số cung của đồ thị G chưa đủ để ghép khi đó ta sẽ phải điều chỉnh lại đồ thị G.

* Ta tìm:

$d = \text{MIN}(F_i[i] + F_j[j] - C[i,j])$

i THUỘC X ĐÃ xét, j THUỘC Y CHƯA xét

(Nếu tìm cặp Ghép min thì $d = \text{MIN}(C[i,j] - F_i[i] - F_j[j])$)

* Thay đổi:

$F_i[i] := F_i[i] - d \text{ Voi mọi } i \text{ THUỘC X ĐÃ xét (Nếu tìm MIN thì } +d)$

$F_j[j] := F_j[j] + d \text{ Voi mọi } j \text{ THUỘC Y ĐÃ xét (Nếu tìm MIN thì } -d)$

Công việc này giúp ta tăng được số cạnh của đồ thị G

Neu ban đầu có đường đi từ $i \rightarrow j$ tức là $F_i[i] + F_j[j] = C[i,j]$
 thì điều này luôn được bảo đảm vì $(F_i[i] - d) + (F_j[j] + d) = C[i,j]$
 Mặt khác sau khi giảm $F_i[i]$ Với mọi i Thuộc X đã xét đi d_{\min}
 thì số cạnh của đồ thị tăng lên ≥ 1 cạnh

Quay lại LOANG lại cho đến khi tìm được cách Ghep

BÀI TOÁN LUỒNG

I / Một số khái niệm :

a) Định nghĩa mạng :

Mạng là đồ thị có hướng $G(V, E)$, V là tập đỉnh, E là tập cung thỏa mãn các điều kiện sau đây :

- + Tồn tại duy nhất 1 đỉnh S không có cung vào (bán bậc vào bằng 0)
- + Tồn tại duy nhất 1 đỉnh T không có cung ra (bán bậc ra bằng 0)
- + Mỗi cung e thuộc E ứng với 1 số không âm $A(e)$

b) Định nghĩa luồng :

Cho mạng $G(V, E)$ với ma trận trọng số A .

Luồng là 1 ánh xạ F từ tập cung E vào tập số thực

$$F : \begin{matrix} E \rightarrow R \\ e \rightarrow F(e) \end{matrix}$$

thỏa mãn các tính chất sau đây :

- + $F(e) \geq 0 \quad \forall e$
- + $A(e) \geq F(e) \quad \forall e$
- + $W(i) = \sum F(e^+) - \sum F(e^-) = 0 \quad \forall$ đỉnh i khác S và T (e^+ là mọi cung ra khỏi đỉnh i , e^- là mọi cung đi tới i). Ngoài ra nếu đặt $W(S) = W$ thì $W(T) = -W$.

$W(i)$ gọi là thông lượng của luồng tại đỉnh i .

$F(e)$ gọi là giá trị của luồng trên cung e .

W là giá trị của luồng.

II / Bài toán luồng thứ nhất :

1) Bài toán : Tìm luồng có giá trị lớn nhất (giá trị W) trong tất cả các luồng xác định trên mạng.

2) Ý nghĩa thực tế : Tìm lưu lượng lớn nhất của hàng hoá vận chuyển trên mạng giao thông.

3) Thuật toán : Dựa trên định lý của Ford Fulkerson “ giá trị của luồng cực đại bằng khả năng thông qua của lát cắt hẹp nhất “. người ta xây dựng thuật toán tìm luồng cực đại.

Trước hết ta định nghĩa nhãn của các đỉnh như sau

- + Nhãn của đỉnh i là $i (+j, v)$ nghĩa là : có thể tăng giá trị luồng trên cung (j, i) một lượng không vượt quá v
- + Nhãn của đỉnh i là $i (-j, v)$ nghĩa là : có thể giảm giá trị của luồng trên cung (i, j) một lượng không vượt quá v .

Để thực hiện thuật toán, người ta sử dụng các động tác sau :

* **Khởi trị** : tạo 1 luồng ban đầu trên mạng (có thể chọn luồng tầm thường là F sao cho $F(e) = 0 \quad \forall e$. Giá trị của luồng là $W=0$

Đầu tiên tất cả các đỉnh chưa có nhãn, và đánh dấu là chưa xét

Gán nhãn $S(+S, \infty)$. Cho S vào stack.

* **Sửa nhãn** : dùng đỉnh j (lấy từ đỉnh stack) để sửa nhãn cho các đỉnh i chưa đánh dấu và kề với j :

Giả sử nhãn đỉnh j $(+k, v)$ hoặc $j(-k, v)$.

+ Nếu cung $(j, i) \in E$, $F[j, i] < A[j, i]$ thì nhãn mới của i là $i(+j, v_0)$, ở đây $v_0 = \min(v, A[j, i] - F[j, i])$

+ Nếu cung $(i, j) \in E$, $F[i, j] > 0$ thì nhãn mới của i là $i(-j, v_0)$, ở đây $v_0 = \min(v, F[j, i])$

Sửa xong nhãn thì cho đỉnh i vào stack

Cuối cùng, sau khi tất cả các đỉnh i đã được sửa nhãn, ta đánh dấu đỉnh j là đã được dùng (để sửa nhãn cho các đỉnh i).

• Điều chỉnh luồng :

+ Xuất phát việc điều chỉnh từ đỉnh T (gán $i := T$)

+ Vòng lặp

$j := i$;

$i :=$ nhãn 1 của j ;

Nếu $i > 0$ thì $F[i, j]$ tăng thêm một lượng v (là nhãn 2 của T)

Nếu $i < 0$ thì $F[j, -i]$ giảm một lượng v

$i := \text{Abs}(i)$;

Lặp cho đến khi $i = S$;

Thuật toán tìm luồng có giá trị lớn nhất :

Repeat

Khởi_trị;

While Stack khác rỗng thực hiện

Begin

Lấy j ở đỉnh Stack;

Nếu còn đỉnh ch- a đã- ọc đánh dấu thì Sửa_nhãn(j)

End;

Nếu đỉnh T đã- ọc đánh dấu thì Điều_chỉnh_luồng ;

Until đỉnh T không thể đánh dấu ;

Cuối cùng, để tìm giá trị cực đại của luồng, ta tính tổng các giá trị của luồng trên các cung xuất phát từ S (nghĩa là ta xét luồng chảy qua 1 lát cắt hẹp nhất, trong lát cắt này tập đỉnh đã- ọc chia thành 2 tập : tập 1 gồm 1 đỉnh duy nhất là S , tập 2 gồm các đỉnh còn lại.)

Uses Crt; Const Max = 100; Fi = 'Luongcd.txt'; Type Kpt = Record
Byte;

delta : Integer;

End;

Knhan = Array[1..Max] of Kpt;

KStack = Array[1..Max] of Byte;

Kdasuanhan = Array[1..Max] of Boolean;

Kmang = Array[1..Max, 1..Max] of Integer;

Var NH : Knhan;

S : Kstack;

A, F : Kmang;

```

D      : Kdasuanhan;
N,Top  : Byte;
Procedure DocF;
  Var i,j : Byte; F : Text;
  Begin
    Assign(F,Fi);
    Reset(f);
    Readln(f,N);
    For i:=1 to N do
      Begin
        For j:=1 to N do Read(f,A[i,j]);
        Readln(f);
      End;
    Close(f);
  End;
Procedure HienF;
  Var i,j : Byte;
  Begin
    For i:=1 to N do
      Begin
        For j:=1 to N do Write(A[i,j]:4);
        Writeln;
      End;
    End;
Function Min(a,b : Integer): Integer;
  Begin
    If a<b then Min:=a else Min:=b;
  End;
Procedure Khoitao;
  Begin
    Fillchar(D,sizeof(D),False);
    FillChar(S,Sizeof(S),0);
    With NH[1] do
      Begin
        truoc := +1;
        delta := MaxInt div 2;
      End;
    D[1] := True;
    Top := 1;
    S[Top] := 1;
  End;
Procedure Suanhan(j : Byte);
  Var i : Byte;
  Begin
    For i:=1 to N do
      If not D[i] then
        Begin
          If (A[j,i]<>0) and (F[j,i]<A[j,i]) then
            Begin
              With NH[i] do
                Begin
                  Truoc := +j;
                  Delta := Min(NH[j].delta,A[j,i]-F[j,i]);
                End;
              D[i] := True;
            End;
          End;
        End;
      End;
    End;
  End;

```

```

    Inc(top);
    S[top] := i;
End
Else
If (A[i,j]<>0) and (F[i,j]>0) then
Begin
    With NH[i] do
    Begin
        Truoc := -j;
        Delta := Min(NH[j].delta,F[i,j]);
    End;
    D[i] := True;
    Inc(top);
    S[top] := i;
End
End;
End;
Procedure Dieuchinh;
Var i,j : Byte;
Begin
    i := N;
    Repeat
        j := i;
        i := NH[j].truoc;
        If i>0 then F[i,j] := F[i,j]+NH[n].delta
        Else
            If i<0 then F[j,-i] := F[j,-i]-NH[n].delta;
            i := abs(i);
        Until i=1;
    End;
Procedure Xaydung;
Var i,j : Byte;
Function Consua : Boolean;
Var i : Integer;
Begin
    For i:=1 to N do
        If Not D[i] then
            Begin
                Consua := True;
                Exit;
            End;
    Consua := False;
End;
Begin
Repeat
    Khoitao;
    While top<>0 do
        Begin
            j := S[top];
            Dec(Top);
            If consua then Suanhan(j);
        End;
        If D[n] then Dieuchinh;
    Until Not D[n];
End;

```

Procedure HienKQ;

Var i,j : Byte; T : Integer;

Begin

For i:=1 to N do

For j:=1 to N do

If $F[i,j] < 0$ then

Writeln('i:2','j:2,') = ',F[i,j]);

T := 0;

For i:=1 to N do

If $F[1,i] < 0$ then Inc(T,F[1,i]);

Writeln('Gia tri luong cuc dai la : ',T);

End;

BEGIN

Clrscr;

DocF; HienF;

Xaydung;

Hienkq;

Writeln('Da xong ');

Readln;

END.

III / Bài toán luồng thứ 2 :

1) Bài toán : Cho đồ thị N đỉnh , thông lượng hàng hoá tối đa trên cung $e(i,j)$ là $A[i,j]$ (hay viết cho gọn là $A[e]$), sức chứa hàng hoá của đỉnh i là $P[i]$ với quy định : nếu $P[i] > 0$ thì đỉnh i gọi là đỉnh thu , $P[i] < 0$ thì i gọi là đỉnh phát , còn khi $P[i] = 0$ thì đỉnh i gọi là đỉnh trung gian (không phát , không thu) . Tìm cách vận chuyển đi - ọc nhiều hàng hoá nhất .

File input Luong2.inp

+ Dòng đầu là số N

+ N dòng tiếp theo là ma trận $A(N,N)$ + Dòng cuối cùng là N số $P[i]$ ($i = 1,2,... N$)

File Output : Luong2.out

Hiện lần lượt các dòng , mỗi dòng 3 số $i,j,F[i,j]$ (ý nghĩa : chuyển $F[i,j]$ hàng từ i tới j)

Dòng cuối cùng là tổng số hàng đi - ọc vận chuyển

2) ý nghĩa : Trong thị trường gặp bài toán tìm cách điều hoà hàng hoá từ nơi này đến nơi khác sao cho sự lưu thông hàng hoá trong toàn thể khu vực chuyển từ các nơi phát đến các nơi thu là tối đa trong điều kiện cho phép . Bài toán luồng thứ 2 này khác bài toán luồng thứ nhất ở chỗ :

+ Có nhiều đỉnh thu và nhiều đỉnh phát

+ Tại mỗi đỉnh có chỉ số dung lượng phát hoặc dung lượng thu tối đa

Còn điểm giống nhau là trên mỗi cung từ đỉnh này sang đỉnh khác vẫn quy định thông lượng tối đa

3) Thuật toán :*a) Một số định nghĩa :*+ Thông lượng tại đỉnh i là $W[i] = \sum F[j,i] - \sum F[i,j]$: Tổng hàng hoá đến i - Tổng hàng hoá ra khỏi i+ Đỉnh thoả mãn là đỉnh i nếu $|W[i]| = |P[i]|$ + Đỉnh ch- a thoả mãn là đỉnh i nếu $|W[i]| < |P[i]|$

+ Luồng thị trường thích trên mạng là luồng thoả mãn các tính chất sau :

1 - $0 \leq F(e) \leq A(e)$ với mọi cung e của mạng2 - $W[i].P[i] \geq 0$ 3 - $|W[i]| \leq |P[i]|$

+ Một dãy chuyển ch- a bão hoà là dãy chuyển đi từ một đỉnh phát ch- a thoả mãn tới một đỉnh thu ch- a thoả mãn , đồng thời trên các cung thuận (hướng trên dãy chuyển đi từ đỉnh phát tới thu) giá trị của luồng < giá trị dung lượng tối đa của cung , còn trên các cung ngược (hướng đi ngược lại) thì giá trị của luồng > 0 .

b) Cơ sở thuật toán : Dựa trên định lý Luồng t-ơng thích đạt cực đại khi không còn dây chuyền ch-a bão hoà đi từ đỉnh phát ch-a thoả mãn đến đỉnh thu ch-a thoả mãn .

c) Thuật toán :

Repeat

Khởi trị : các đỉnh ch- a đánh dấu ($D[i] := -$ vô cùng)

Tìm đỉnh i là đỉnh phát ch- a thoả mãn

Nếu tìm đ- ọc i (nghĩa là $i < 0$) thì

Tìm dây chuyền ch- a bão hoà xuất phát từ i

Nếu tìm đ- ọc dây chuyền thì Điều chỉnh luồng

Until Không tìm đ- ọc dây chuyền ch- a bão hoà

Hai động tác chính trong thuật toán là : Tìm dây chuyền , Điều chỉnh luồng

Tìm dây chuyền xuất phát từ đỉnh i :

+ Đánh dấu đỉnh i đã xét ($D[i] := 0$)

+ Cho i vào Stack

+ While Stack ch- a rỗng và

dây chuyền ch- a kết thúc (nghĩa là ch- a gặp đỉnh thu ch- a thoả mãn) thì

Begin

+ Lấy đỉnh k từ đỉnh Stack

+ Vòng lặp For : xét các đỉnh j ch- a đ- ọc đánh dấu

Nếu việc tìm dây chuyền ch- a kết thúc thì

Begin

Nếu (k,j) là cung thuận ch- a bão hoà thì

Begin

+ Nạp j vào Stack

+ Đánh dấu đã xét j ($D[j] := k$)

+ Nếu j là đỉnh thu ch- a thoả mãn thì kết thúc dây chuyền

End;

Nếu (j,k) là cung ng- ợc ch- a bão hoà thì

Begin

+ Nạp j vào Stack

+ Đánh dấu đã xét j ($D[j] := - k$)

+ Nếu j là đỉnh thu ch- a thoả mãn thì kết thúc dây chuyền

End;

End;

End;

Điều chỉnh luồng :

Lấy một đỉnh i từ Stack

Repeat

j := i;

i := $D[i]$ (Đỉnh kế tr- ớc của i trong dây chuyền là $D[i]$)

Nếu $i > 0$ thì tăng luồng trên cung thuận (i,j) 1 đơn vị

Nếu $i < 0$ thì giảm luồng trên cung ng- ợc (j,i) 1 đơn vị

Until Lấy hết các đỉnh của dây chuyền ch- a bão hoà (chứa trong Stack)


```

Uses Crt;
Const Max = 100;
  Fi = 'Luongl2.txt';
  Fo = 'Luongl2.out';
Type Ta = Array[1..Max,1..Max] of Integer;
  Tb = Array[1..Max] of Integer;
Var A : Ta; { Thông lượng tối đa trên các cung }
  F : Ta; { Lương }
  P : Tb; { Số chứa tại mỗi đỉnh }
  S : Tb; { Stack }
  D : Tb; { Mảng đánh dấu đồng thời theo dõi đỉnh trước }
  N,Top : Integer;
  out : Text;
  Ok : Boolean;
Procedure Nhap;
  Var i,j : Byte;
  F : Text;
  Begin
    Assign(F,Fi);
    Reset(F);
    Readln(F,N);
    For i:=1 to N do
      Begin
        For j:=1 to N do Read(F,A[i,j]);
        Readln(F);
      End;
    For i:=1 to N do
      Read(F,P[i]);
    Close(F);
  End;
Procedure Hien;
  Var i,j : Byte;
  Begin
    For i:=1 to N do
      Begin
        For j:=1 to N do Write(A[i,j]:4);
        Writeln;
      End;
    Writeln;
    For i:= 1 to N do Write(P[i]:4);
    Writeln;
  End;
Function Giatri : Integer;
  Var i,j,gt : Integer;
  Begin
    gt := 0;
    For i:=1 to n do
      For j:=1 to n do
        If P[j]<>0 then Inc(gt,F[i,j]);
    Giatri := gt;

```

```

End;
Procedure HienKq;
  Var i,j : Byte;
  Begin
    For i:=1 to n do
      Begin
        For j:=1 to n do
          If P[j]<>0 then Write(out,F[i,j]:4)
          Else Write(out,0:4);
        Writeln(out);
      End;
    Writeln(out);
    Writeln(out,'Gia tri luong : ',Giatri);
  End;
Function Thongluong(i : Byte) : Integer;
  Var j : Byte;
  thlg : Integer;
  Begin
    Thlg := 0;
    For j:=1 to N do
      Begin
        If A[i,j]>=0 then Inc(thlg,F[i,j]);
        If A[j,i]>0 then Dec(thlg,F[j,i]);
      End;
    Thongluong := thlg;
  End;
Function Thoaman(i : Byte) : Boolean;
  Begin
    If Abs(Thongluong(i))<Abs(P[i]) then Thoaman := False
    Else Thoaman := True;
  End;

Function TimPhat : Byte;
  Var i,j : Byte;
  Begin
    TimPhat := 0;
    For i:=1 to N do
      If D[i]=-MaxInt then
        If P[i]<0 then
          If Not Thoaman(i) then
            Begin
              Timphat := i;
              Exit;
            End;
    End;
  End;
Procedure Daychuyen(i : Byte);
  Var j,k : Byte;
  Begin
    D[i] := 0;
    Top := 1;
    S[Top] := i; {Lan luot cho cac dinh cua day chuyen vao Stack }
    While (Top<>0) and (Not Ok) do
      Begin
        k := S[top];
        Dec(Top);
      End;
  End;

```

```

For j:=1 to N do
  If (D[j]=-MaxInt) then
    Begin
      If Not Ok then { Not Ok:Chua ket thuc day chuyen }
      Begin
        If (A[k,j]>F[k,j]) then
          Begin
            D[j] := k;
            Inc(Top);
            S[Top] := j;
            Ok := (P[j]>0) and (Not Thoaman(j));
          End
        Else
          If (A[j,k]>=0) and (F[j,k]>0) then
            Begin
              D[j] := -k;
              Inc(Top);
              S[Top] := j;
              Ok := (P[j]>0) and (Not Thoaman(j));
            End;
          End;
        End;
      End;
    End;
  End;
End;
Procedure Dieuchinh;
  Var i,j : Byte;
  Begin
    i := S[Top]; { Lan nguoc day chuyen , bat dau tu dinh stack }
    Repeat
      j := i;
      i := D[i];
      If i>0 then Inc(F[i,j]);
      If i<0 then Dec(F[j,-i]);
      i := Abs(i);
    Until i=0;
  End;
Procedure Luongl2;
  Var i : Byte;
  Begin
    Repeat
      Ok := False;
      For i:=1 to N do D[i]:=-MaxInt;
      i := TimPhat; { Tim dinh phat chua thoa man }
      If i<>0 then
        Begin
          Daychuyen(i); { Ok = Tim duoc day chuyen chua bao hoa }
          If Ok then Dieuchinh;
        End;
      Until Not Ok;
      HienKq;
    End;
  BEGIN
    Clrscr;
    Nhap;
    Hien;

```

```
Assign(out,Fo);
ReWrite(out);
Luong12;
Close(out);
Writeln('Da xong ');
END.
```

Bài tập về qui hoạch động

Bài Mã vạch :

Cho bộ 3 số (N,M,K) nguyên không âm ($N \leq 100, M, K \leq 33$). Ng-ời ta định nghĩa mỗi bộ 3 số trên ứng với 1 mã là một xâu kí tự dạng nhị phân thoả mãn :

- + Chứa đúng N chữ số
- + Các chữ số 0 liên nhau hoặc các chữ số 1 liên nhau gọi là 1 vạch , phải có đúng M vạch
- + Số chữ số trong 1 vạch gọi là độ rộng của vạch . Độ rộng tối đa của vạch là K
- + Vạch đầu tiên của mã phải là vạch gồm các chữ số 1.

Lập trình thực hiện các yêu cầu sau :

1) Lấy dữ liệu từ File 'MV.INP' tổ chức như sau :

- Dòng đầu là 3 số N, M, K
- Dòng thứ 2 là số p
- p dòng tiếp theo : mỗi dòng là một mã M_i ($0 < i < p+1$) của bộ mã (M, N, K)

2) Thông tin ra gửi vào File 'MV.OUT' :

- Dòng đầu là số nêu tổng số mã của bộ mã (N, M, K)
- Tiếp theo gồm p dòng , mỗi dòng ghi 1 số là vị trí của mã M_i trong tự điển xếp tăng các mã của bộ mã (N, M, K) .

Thí dụ

File 'MV.INP'

```
7 4 3
6
1110100
1101100
1001000
1000100
1101110
1110110
```

File 'MV.OUT'

```
16
15
12
3
1
13
16
```

```

Const  Fi      = 'Mv.inp';
       Fo      = 'Mv.out';
       MaxN    = 100;
       MaxM    = 33;
Type   Pt      = Array[1..13] of Byte;
       Ma      = Array[1..104] of 0..1;
       Bang    = Array[0..MaxM,0..MaxN] of Pt;
Var    N,M,K : Byte;
       F      : Bang;
       X      : Ma;
       P      : Pt;
Procedure Dan(P : Pt;Var X : Ma);
  Var i,j,t,tg : Byte;
  Begin
    FillChar(X,Sizeof(X),0);
    T := 0;
    For i:=1 to 13 do
      For j:=0 to 7 do
        Begin
          Inc(T);
          X[t] := (P[i] SHR j) and 1;
        End;
      End;
    End;
  End;
Procedure Nen(X : Ma;Var P : Pt);
  Var i,j,t,tg : Byte;
  Begin
    FillChar(P,Sizeof(P),0);
    T := 0;
    For i:=1 to 13 do
      Begin
        Tg := 0;
        For j:=0 to 7 do
          Begin
            Inc(T);
            Tg := Tg+X[t] SHL j;
          End;
        End;
        P[i] := Tg;
      End;
    End;
  End;
Procedure Cong(Var A : Ma;B : Ma);
  Var i,t,nho : Byte;
  Begin
    Nho := 0;
    For i:= 1 to 104 do
      Begin
        T := A[i]+B[i]+Nho;
        A[i] := T mod 2;
        Nho := T div 2;
      End;
    End;
  End;
Procedure TaoBang; { F[x,y]=So luong cac ma co x vach , dai y ki tu }
  Var i,j : Byte;F3 : Text;
  Procedure Xaydung(x,y:Byte);
    Var i : Byte; A,B : Ma;
    Begin

```

```

    Dan(F[x,y],A);
    For i:=1 to k do
        If i<y then
            Begin
                Dan(F[x-1,y-i],B);
                Cong(A,B);
            End;
        Nen(A,F[x,y]);
    End;
Begin
    FillChar(F,Sizeof(F),0);
    For i:=1 to M do F[i,i,1] := 1;
    For i:=1 to K do F[1,i,1] := 1;
    For i:=2 to M do
        For j:=i+1 to N do
            If i*k>=j then Xaydung(i,j);
    End;
Procedure Nhan(Var S : String;T : Byte);
    Var i,tg,nho,L : Byte;
    Begin
        L := Length(S);
        While(L>1) and (S[1]='0') do
            Begin
                Dec(L);
                Delete(S,1,1);
            End;
        Nho := 0;
        For i:= Length(S) downto 1 do
            Begin
                Tg := (Ord(S[i])-48)*T+Nho;
                S[i] := Char(Tg mod 10 + 48);
                Nho := Tg div 10;
            End;
        If Nho<>0 then S := Char(Nho+48)+S;
    End;
Procedure CongS(Var S1 : String;S2 : String);
    Var i,tg,nho,L1,L2,L : Byte;
    Begin
        Nho := 0;
        L1 := Length(S1);
        L2 := Length(S2);
        If L1<L2 then L := L2 Else L := L1;
        While Length(S1)<L do S1 := '0'+S1;
        While Length(S2)<L do S2 := '0'+S2;
        For i:=L downto 1 do
            Begin
                Tg := Ord(S1[i])+Ord(S2[i])-96+Nho;
                S1[i] := Char(Tg mod 10 +48);
                Nho := Tg div 10;
            End;
        If Nho<>0 then S1 := Char(Nho+48)+S1;
    End;
Function Doi(P : Pt) : String; { Doi mang P dang nhi phan thanh xau }
    Var
        X      : Ma;
        i,j    : Byte;

```

```

    S,LT,SP : String;
Begin
    Dan(P,X);
    Lt := '1';
    S := '0';
    j := 104;
    While X[j]=0 do Dec(j);
    For i:=1 to j do
    Begin
        Sp := LT;
        Nhan(Sp,X[i]);
        CongS(S,Sp);
        Nhan(Lt,2);
    End;
    Doi := S;
End;
Procedure Vitri(S : String;Var P : Pt);
Var
    Ch      : Char;
    i,j,d,L : Byte;
    A,B     : ma;
Begin
    FillChar(A,Sizeof(A),0);
    D := Length(S);
    For i:=M downto 2 do
    Begin
        Ch := S[i];
        L := 0;
        While (D>0) and (S[i]=ch) do
        Begin
            Inc(L);
            Delete(S,i,1);
            Dec(D);
        End;
        Case ch of
            '1' : For j:=2 to L do
                Begin
                    Dan(F[i-1,D+L-j+1],B);
                    Cong(A,B);
                End;
            '0' : For j:=k-L downto 1 do
                Begin
                    Dan(F[i-1,D-j],B);
                    Cong(A,B);
                End;
        End;
    End;
    Nen(A,P);
End;
Procedure Lam;
Var
    F1,F2 : Text;
    S      : String;
    P      : Pt;
    H,i    : Integer;
Begin
    Assign(F1,Fi);

```

```

Reset(F1);
Assign(F2,Fo);
Rewrite(F2);
Readln(F1,N,M,K); { Ma : N kí tự, có M vạch, độ rộng max của vạch : k }
TaoBang;
S := Doi(F[M,N]); { Ghi tổng số ma }
Writeln(F2,S);
Readln(F1,H); { Đọc số lượng các ma cần chuyển từ ma thành vitri }
For i:=1 to H do
Begin
    Readln(F1,S);
    Vitri(S,P);
    S := Doi(P);
    CongS(S,'1');
    Writeln(F2,S);
End;
Close(f2);      Close(F1);
End;
BEGIN
    Clrscr;      Lam;      Writeln('Xong');      Readln;
END.

```

ÔỦ BÀI :

Cho một hình chữ nhật $n \times m$ ô vuông, mỗi ô vuông nhận giá trị 0 hoặc 1. Vùng các ô có giá trị 1 chung cạnh gọi là một vùng liên thông. Nếu trong hình chữ nhật này chỉ có một vùng liên thông thì vùng này gọi là một mẫu.

Câu a : Nhập từ file SOMAU.INP hai số nguyên m, n và hai hình chữ nhật. Thông báo hai hình chữ nhật đó có phải là hai mẫu không.

Câu b : Hai mẫu gọi là t-ơng đ-ơng nếu diện tích của chúng bằng nhau. Nếu câu a đ-ợc hai mẫu thì hai mẫu đó đ-ợc t-ơng đ-ơng không.

Câu c : Đặt hai mẫu trên cùng một hệ trục tọa độ, nếu tịnh tiến dọc các trục mà hai mẫu trùng khít lên nhau thì ta nói hai mẫu đó bằng nhau. Nếu câu b đ-ợc hai mẫu t-ơng đ-ơng thì hai mẫu đó bằng nhau hay không?

Câu d : Nếu kết hợp thực hiện tịnh tiến dọc các trục tọa độ và phép quay một mẫu, một góc 90° mà hai mẫu trùng khít lên nhau thì ta nói hai mẫu bằng nhau kiểu 2. Kiểm tra hai mẫu đã nhập trong file có bằng nhau kiểu 2 hay không?

LỖI GIỎI:

(học sinh tự làm câu d)

```

Uses Crt;
Const Max = 50;
      Fi = 'somau.inp';
Type Pt = Record x,y : Byte ; End;
      MangM = Array[0..Max,0..Max] of Byte;
      MangQ = Array[1..Max*Max] of Pt;
      MangD = Array[1..Max,1..Max] of Pt;
Var N,M : Byte;
    A,B : MangM;
    Q : MangQ;
    D : MangD;
    S1,S2 : Integer;
(*-----*)

```


Procedure NhapFile;

Var i,j : Byte;

F : Text;

Begin

FillChar(A,Sizeof(A),3);

FillChar(B,Sizeof(B),3);

Assign(F,Fi);

Reset(F);

Readln(F,M,N);

For i:=1 to M do

For j:=1 to N do Read(F,A[i,j]);

For i:=1 to M do

For j:=1 to N do Read(F,B[i,j]);

Close(F);

End;

(*-----*)

Function Tim1(Var MX : MangM; Var x,y : Byte): Boolean;

Var i,j : Byte;

Begin

For i:=1 to M do

For j:=1 to N do

If MX[i,j]=1 then

Begin

x := i;

y := j;

Tim1 := True;

Exit;

End;

Tim1 := False;

End;

(*-----*)

Function Mau(Var MX : MangM;Var DT : Integer) : Boolean;

Var Ok : Boolean;

F,L : Integer;

x,y,k : Byte;

Procedure Loang(Var MX : MangM; x,y,i,j : Byte);

Begin

If MX[i,j]=1 then

Begin

MX[i,j] := 2;

Inc(L);

Q[L].x := i;

Q[L].y := j;

D[i,j].x := x;

D[i,j].y := y;

End

Else

If (MX[i,j]=2) and ((i<>D[x,y].x) or (j<>D[x,y].y))

then Ok := True;

End;

Begin

Ok := False;

If Tim1(MX,x,y) then

Begin

F := 0;

```

    L := 1;
    Q[L].x := x;
    Q[L].y := y;
    MX[x,y] := 2;
    Repeat
        Inc(F);
        x := Q[F].x;
        y := Q[F].y;
        Loang(MX,x,y,x-1,y);
        Loang(MX,x,y,x+1,y);
        Loang(MX,x,y,x,y-1);
        Loang(MX,x,y,x,y+1);
    Until F=L;
    If Tim1(MX,x,y) then Ok := False;
    Mau := Ok;
    DT := L;
End;
End;
Function Thongbao(Var X : MangM; Var S : Integer) : Boolean;
Begin
    S := 0;
    If Not Mau(X,S) then
        Begin
            Writeln('Du lieu khong dung ');
            Thongbao := False;
            Exit;
        End;
    Thongbao := True;
End;
Procedure Timkhung(Var X : MangM; Var x1,y1,x2,y2 : Byte);
Function MinD : Byte;
Var i,j : Byte;
Begin
    For i:=1 to M do
        For j:=1 to N do
            If X[i,j]=2 then
                Begin
                    MinD := i;
                    Exit;
                End;
        End;
    End;
End;
Function MaxD : Byte;
Var i,j : Byte;
Begin
    For i:=M downto 1 do
        For j:=1 to N do
            If X[i,j]=2 then
                Begin
                    MaxD := i;
                    Exit;
                End;
        End;
    End;
End;
Function MaxC : Byte;
Var i,j : Byte;
Begin

```

```

For j:=N downto 1 do
  For i:=1 to M do
    If X[i,j]=2 then
      Begin
        MaxC := j;
        Exit;
      End;
  End;
End;
Function MinC : Byte;
Var i,j : Byte;
Begin
  For j:=1 to N do
    For i:=1 to M do
      If X[i,j]=2 then
        Begin
          MinC := j;
          Exit;
        End;
    End;
  End;
Begin
  x1 := MinD;
  x2 := MaxD;
  y1 := MinC;
  y2 := MaxC;
End;
(*-----*)
Function Trung : Boolean;
Var xa1,xa2,xb1,xb2,ya1,ya2,yb1,yb2,i,j : Byte;
Ok : Boolean;
L1,L2,x,y : Byte;
Begin
  TimKhung(A,xa1,ya1,xa2,ya2);
  TimKhung(B,xb1,yb1,xb2,yb2);
  L1 := Abs(xa1-xb1);
  L2 := Abs(ya1-yb1);
  Trung := True;
  If (xa2-xa1)*(ya2-ya1)=(xb2-xb1)*(yb2-yb1) then
    Begin
      For i:= xa1 to xa2 do
        For j:= ya1 to ya2 do
          If A[i,j]=2 then
            Begin
              If xa1<xb1 then x := i+L1
              Else
                If xa1>xb1 then x := i-l1
                Else x := i;
              If ya1<yb1 then y := j+L2
              Else
                If ya1>yb1 then y := j-L2
                Else y := j;
              If A[i,j]<>B[x,y] then
                Begin
                  Trung := False;
                  Exit;
                End;
            End;
          End;
        End;
      End;
    End;
  End;
End;

```

```

End;
End
Else
Trung := False;
End;
BEGIN
NhậpFile;
Clrscr;
If Thongbao(A,S1) and Thongbao(B,S2) then
If S1=S2 then
Begin
Writeln('Hai mau tuong duong ve mat dien tich ');
If Trung then Writeln('Hai mau co the tinh tien trung nhau ')
Else Writeln('Hai mau khong the tinh tien trung nhau ');
End
Else
Writeln('hai mau khong tuong duong, khong trung nhau ');
Writeln('ENTER thoat ');
Readln;

END.

```

TỔNG N

M_N : THỜI KÌ THUẬT TOÁN

I/ Dynamic programing

a) Gán nhãn (Dijkstra) Tìm đường đi ngắn nhất trên đồ thị có trọng số không âm từ đỉnh u (nguồn) tới mọi đỉnh d (đích). Trọng số $C[i,j]$ là trọng số từ đỉnh i tới đỉnh j .

Trước hết ta gọi nhãn của đỉnh i ($\forall i : 1 \leq i \leq N$) là cặp số (b,v) với ý nghĩa : b là đỉnh kề ngay trước i của đường đi ngắn nhất từ u tới i , v là giá trị đường đi ngắn nhất từ u tới i . Ký hiệu $i(b,v)$

+ khởi trị nhãn :

- * nhãn mọi đỉnh i là : $i(0, \text{Max}) \forall i : 1 \leq i \leq N$
- * nhãn đỉnh xuất phát là : $u(u,0)$
- * Ghi nhận đỉnh $x = u$ và kết nạp x vào tập đỉnh đã xét : $ex[x] = 1$

+ Trong khi $x \neq d$ (đích) và ($x \neq 0$) thực hiện vòng lặp :

begin

- * sửa nhãn các đỉnh i (b_i, v_i) chưa kết nạp và có đường đi từ x tới i theo nguyên tắc : giả sử nhãn x là $x(b_x, v_x)$, nếu $b_x + C[x,i] < b_i$ thì đỉnh i có nhãn mới là $i(x, b_x + C[x,i])$
- * Chọn đỉnh i_0 có nhãn nhỏ nhất trong các đỉnh chưa kết nạp vào tập đỉnh đã xét , nếu tìm được thì kết nạp i_0 vào tập đỉnh đã xét , gán $x = i_0$. Nếu không chọn được thì $x = 0$

end;

+ Lặp lại theo nhãn thứ nhất để tìm đường đi

$i = d$

Trong khi $i \neq u$ thực hiện vòng lặp

Begin

- + ghi lại u i vào mảng kết quả
- + i nhận giá trị nhãn thứ nhất của i

end;

```

uses crt;const max = 100;          fi = 'dijskra.001';type tc = array[1..max,1..max]
of integer;{ cost } tb = array[1..max] of shortint; { befor } tv = array[1..max] of
integer; { value } tr = array[1..max] of char; { result }
      tex = array[1..max] of 0..1; { examined : da xem xet }
var c : tc;
    t : tb;
    v : tv;
    rs : tr;
    ex : tex;
    n , u , d ,x : byte;

```

```

procedure docf;
var i,j : byte;
    f : text; begin
fillchar(c,sizeof(c),0);
assign(f,fi);
reset(f);
readln(f,n,u,d);
while not eof(f) do
begin
    readln(f,i,j,c[i,j]);
    c[j,i] := c[i,j];
end;
close(f);
end;

```

```

procedure hienf;
var i,j : byte;
begin
    writeln(n,' ',u,' ',d);
    for i:=1 to n do
    begin
        for j:=1 to n do write(c[i,j]:5);
        writeln;
    end;
end;

```

```

procedure khoitrinhan;
var i : byte;
begin
    fillchar(ex,sizeof(ex),0);
    for i:=1 to n do
    begin
        t[i] := 0;
        v[i] := maxint;
    end;
    t[u] := u;
    v[u] := 0;
    x := u;
    ex[u] := 1;
end;

```

```

procedure suanhan;
var i : byte;
begin
    for i:=1 to n do
        if c[x,i]>0 then
            if ex[i]=0 then

```

```

begin
    if v[x]+c[x,i]<v[i] then
        begin
            v[i] := v[x] + c[x,i];
            t[i] := x;
        end;
    end;
end;
function chon : byte;
var i,li    : byte;
    min     : integer;
begin
    min := maxint;
    li := 0;
    for i:=1 to n do
        if ex[i]=0 then
            if v[i]<min then
                begin
                    min := v[i];
                    li := i;
                end;
            chon := li;
        end;
end;
procedure suanhhan_ketnap;
begin
    suanhhan;
    ex[x] := 1;
    x := chon;
end;
procedure thuchien;
begin
    khoitrinhan;
    while (x<>d) and (x<>0) do
        suanhhan_ketnap;
    end;
end;
procedure lannguoc;
var i,j,dem : byte;
begin
    i := d;
    dem := 0;
    while i<>u do
        begin
            inc(dem);
            rs[dem] := char(i);
            i := t[i];
        end;
    inc(dem);
    rs[dem] := char(u);
    for i:=dem downto 1 do write(ord(rs[i]),' ');
end;
BEGIN
    clrscr;
    docf;
    hienf;
    thuchien;

```

lannguoc;
END.

Input

6 1 4 { 6 đỉnh , xuất phát từ đỉnh 1 , tới đỉnh 4 } 1 2 4 1 6 2 2 3 5 2 6 1 3 4 6 3 5 2 3 6 8 4 5 3 5 6 10

Output : 1 6 2 3 5 4

b) Bài toán 0/1 knapsack :

Cho n đồ vật , đồ vật thứ i có trọng lượng là w_i , giá trị là v_i .Ng- ời ta xếp các đồ vật vào 1 chiếc va ly có sức chứa tối đa là $limw$. Hãy chọn những đồ vật nào xếp vào va ly để giá trị va ly là lớn nhất .

Đây là bài toán tìm véc tơ $x = (x_1, x_2, \dots, x_n)$ với x_i chỉ nhận giá trị 0,1 , sao cho

$\sum x_i . w_i \leq limw$ và $\sum x_i . v_i$ đạt max .

Cách giải :

$$V_{max} = \text{Max}(V_1, V_2)$$

Trong đó $V_1 = V_{max} (M, N-1)$
 V

```
{ xếp các đồ vật vào va ly, mỗi loại chỉ chọn tối đa là 1 vật }
uses crt; const mn = 100;
mw = 300;
fi = 'knapsack.inp';
fo = 'knapsack.out';
type tf = array[0..mn, 0..mw] of integer;
twv = array[1..mn] of integer;
tkq = array[1..mn] of byte;
var f : tf; g : text; w, v : twv; tong : integer;
mt, luumt, n, limw : integer;
procedure docf;
var i, j : integer;
f : text;
begin
assign(f, fi); reset(f);
read(f, n, limw);
for i:=1 to n do read(f, w[i]);
for i:=1 to n do read(f, v[i]);
close(f);
end;
procedure hienf;
var i, j : integer;
begin
write(n, ' ', limw); writeln;
for i:=1 to n do write(w[i]:4); writeln;
for i:=1 to n do write(v[i]:4); writeln;
end;
procedure taobang;
var i, j : integer;
function max2(x, y : integer) : integer;
begin
if x < y then max2 := y else max2 := x;
end;
begin
for i:=0 to n do
```

```

    for j:=0 to limw do f[i,j] := -1;
    for i:=0 to n do
    for j:=0 to limw do f[i,j] := -1;
    for j:=0 to limw do f[0,j] := 0;

    for i:=0 to n do f[i,0] := 0;
    for i:=1 to n do
    for j:=1 to limw do
    begin
        if f[i,j]=-1 then
        if (j-w[i]>=0) then
            f[i,j] := max2(f[i-1,j],f[i-1,j-w[i]]+v[i])
        else f[i,j] := f[i-1,j];
    end;
    end;
procedure timkq(i,j : Integer);
begin
    if (i<>0) and (j<>0) then
    begin
        if f[i,j]=f[i-1,j] then timkq(i-1,j)
        else
        begin
            writeln(g,'vat thu ',i:4,' : w =':8,w[i]:4,'v =' :8,v[i]:4);
            timkq(i-1,j-w[i]);
            tong := tong+w[i];
        end;
    end;
end;
BEGIN
    clrscr;
    docf;
    hienf;
    taobang;
    tong := 0;
    assign(g,fo);
    rewrite(g);
    timkq(n,limw);
    Writeln(g,'tong gia tri va ly : ',f[n,limw]);
    Writeln(g,'tong trong luong : ',tong);
    writeln('da chay xong chuong trinh ');
    close(g);
    readln;
END.

```


II / Đề quy**Bài tập 2 : Mã đi tuần :**

Cách 1 : Đề quy tìm mọi nghiệm , chỉ chạy đ- ọc với n khoảng 6 hoặc 7

```

uses crt; const max = 10;      dy : array[1..8] of -2..2 = (-1, 1, 2, 2, 1, -1, -2, -2);
  dx : array[1..8] of -2..2 = (-2, -2, -1, 1, 2, 2, 1, -1);
  fo = 'nnn.dat';
var a : array[-1..max, -1..max] of shortint;
  m, n, x, y, i, sn : integer;
  f : text;
procedure nhap;
begin
  write('m,n = '); readln(m,n);
  write('Toa do (x,y) cua o xuất phát : '); readln(x,y);
end;
procedure khoitri;
var i, j : integer;
begin
  for i := -1 to m+2 do
    for j := -1 to n+2 do a[i,j] := -1;
  for i := 1 to m do
    for j := 1 to n do a[i,j] := 0;
  a[x,y] := 1;
end;
procedure hien;
var i, j : integer;
begin
  inc(sn);
  writeln(f, sn);
  for i := 1 to m do
    begin
      for j := 1 to n do Write(f, a[i,j]:6);
      writeln(f);
    end;
end;
procedure vet(i, x, y : integer);
var j, u, v : integer;
begin
  if i > m*n then hien;
  for j := 1 to 8 do
    begin
      u := x + dx[j];

```

```
        v := y + dy[j];
        if (a[u,v]=0) then
        begin
            a[u,v] := i;
            vet(i+1,u,v);
            a[u,v] := 0;
        end;
    end;
end;
BEGIN
    clrscr;
    nhap;
    khoitri;
    sn := 0;
    i := 2;
    assign(F,Fo);
    rewrite(F);
    vet(i,x,y);
    if sn=0 then writeln(f,'vo nghiem ');
    close(F);
END.
```

III / Tham lam :

Bài mã đi tuần (Cách 2) Tham lam , tìm 1 nghiệm chạy đ- ọc với n khoảng 30 hoặc 40

```

{$A+,B-,D+,E+,F-,G-,I+,L+,N-,O-,P-,Q-,R+,S+,T-,V+,X+}{$M 56384,0,655360}Uses crt;Const
    Max    = 50;          dx      : Array[1..8] of integer=(-2,-2,-1,1, 2, 2,1,-1);      dy      :
Array[1..8] of integer=( -1,1, 2,2,1,-1,-2,-2);
Var    N,x,y  : Integer;
        A      : Array[-1..max+2,-1..max+2] of Integer;
        dem    : Integer;
        F      : Text;
Procedure Nhap;
Begin
    Write('Nhap kich thuoc ban co = ');
    Readln(n);
    Write('Nhap toa do xuất phát x,y = ');
    Readln(x,y);
End;
Procedure Hien;
Var    i,j    : Integer;
Begin
    Inc(dem);
    For i:=1 to n do
        Begin
            For j:=1 to n do write(F,a[i,j]:4);
            Writeln(F);
        End;
    End;
End;
Procedure Hangrao;
Var    i,j    : Integer;
Begin
    Fillchar(a,sizeof(a),0);
    For i:=-1 to n+2 do
        For j:=1 to 2 do
            Begin
                A[i,1-j]:=-1;
                A[i,n+j]:=-1;
                A[1-j,i]:=-1;
                A[n+j,i]:=-1;
            End;
        End;
    End;
End;
Function Bac(x,y:integer) : Integer;
Var    i,dem  : Integer;
Begin
    dem:=0;
    For i:=1 to 8 do
        If a[x+dx[i],y+dy[i]]=0 then inc(dem);
        Bac:=dem;
    End;
End;
Procedure Vet(so,i,j:integer);
Var    k,lk ,Ldem,p : Integer;
Begin
    If so>n*n then
        Begin
            Clrscr;
            Hien;
        End;
    Ldem:=9;
    For k:=1 to 8 do

```

```

    If A[i+dx[k],j+dy[k]]=0 then
      Begin
        P := Bac(i+dx[k],j+dy[k]);
        If (Ldem>P) and (P>=0) then
          Begin
            Lk := k;
            Ldem := p;
          End;
        End;
      End;

    If Ldem = 9 then exit;
    If Ldem<9 then
      Begin
        A[i+dx[Lk],j+dy[Lk]] := So;
        Vet(so+1,i+dx[Lk],j+dy[Lk]);
        A[i+dx[Lk],j+dy[Lk]] := 0;
      End;
    End;
  End;
Procedure Lam;
  Begin
    Hangrao;
    A[x,y]:=1;
    Vet(2,x,y);
  End;
BEGIN
  Clrscr;
  Nhap;
  Assign(F,'Ma.txt');
  ReWrite(F);
  dem := 0;
  Lam;
  If dem=0 then Writeln(F,'Vo nghiem ');
  Close(F);
  Writeln('Da xong');
  Readln;
END.

```

Cách 2b : Tham lam , chỉ tìm 1 nghiệm , chạy đ- ọc với n khoảng 100 .

```

{$A+,B-,D+,E+,F-,G-,I+,L+,N-,O-,P-,Q-,R+,S+,T-,V+,X+} {$M 56384,0,655360}uses crt;const
  max = 100; fo = 'banco.out';
  dx : array[1..8] of integer=(-2,-1,1,2,2,1,-1,-2);
  dy : array[1..8] of integer=(1,2,2,1,-1,-2,-2,-1);
type mang = array[1..max,1..max] of integer;
var f : text;
    a : mang;
    x,y,u,v,n,m : integer;
procedure nhap;
begin
  write('m,n = ');readln(m,n);
  write('x,y = ');readln(x,y);
end;
function trong(x,y:integer):boolean;
begin

```

```

    trong := (x>0) and (y>0) and (x<m+1) and (y<n+1);
end;
function bac(x,y : integer) : integer;
var i,j,dem : integer;
    lx,ly : integer;
begin
    dem:=0;
    for i:=1 to 8 do
        begin
            lx := x+dx[i];
            ly := y+dy[i];
            if (trong(lx,ly)) and (a[lx,ly]=0) then inc(dem);
        end;
    bac := dem;
End;
procedure chon(x,y : integer;var u,v:integer);
var i,b,lb,lx,ly : integer;
begin
    lb:=255;
    u:=0;v:=0;
    for i:=1 to 8 do
        begin
            lx:=x+dx[i];
            ly:=y+dy[i];
            If(trong(lx,ly)) and (a[lx,ly]=0) then
                begin
                    b:= bac(lx,ly);
                    if b<lb then
                        begin
                            lb := b;
                            u := lx;
                            v := ly;
                        end;
                    end;
                end;
        end;
    end;
end;

procedure lam;
var sb : integer;
    procedure hien;
    var i,j : integer;
    begin
        assign(f,fo);
        rewrite(f);
        writeln(f,sb-1);
        for i:=1 to m do
            begin
                for j:=1 to n do
                    write(f,a[i,j]:7);
                    writeln(f);
                end;
            close(f);
        end;
    begin
        a[x,y]:=1;

```

```

sb:=1;
chon(x,y,u,v);
while (u<>0) and (v<>0) do
begin
  x := u;
  y := v;
  inc(sb);
  a[x,y] := sb;
  chon(x,y,u,v);
end;
hien;
end;
BEGIN
  nhap;
  lam;
END.

```

IV Backtracking : Th- ờng dùng với lớp các bài toán tìm kiếm thoả 2 tính chất :

+ Không có bản đồ tìm kiếm xác định

+ Tại mỗi b- ớc tìm kiếm có 1 tập hữu hạn các khả năng $Pset(i) = A_i \cup B_i$

Mỗi tập khả năng của b- ớc i gồm 2 tập con không giao nhau A_i và B_i . Trong đó A_i là tập cá khả năng đã duyệt , B_i ch- a duyệt . Nếu $B_i = \Phi$ (mọi khả năng của b- ớc i đã duyệt hết) mà ch- a đạt kết quả thì lùi một b- ớc trở về b- ớc tr- ớc . Ng- ợc lại khi B_i khác rỗng thì ta chọn một khả năng của B_i , cho đi tiếp . Thuật toán kết thúc khi gặp kết quả .

Ng- ợc lại , sau khi thăm hết mọi khả năng của mọi b- ớc mà không đạt kết quả ta cũng dừng thuật toán .

Các bài toán loại này kết quả th- ờng chứa 2 điều kiện P và Q . Khi tìm kiếm ta th- ờng tạm bỏ qua 1 điều kiện , thí dụ nh- bỏ điều kiện P , tại mỗi b- ớc tìm kiếm ta chỉ cần khảo sát các khả năng thoả mãn điều kiện Q .

Sơ đồ giải tìm 1 nghiệm :

```

Khởi trị mảng chứa kết quả V thoả mãn điều kiện P
Repeat
  If gặp Đích then begin Hiện nghiệm ; exit ; end;
  If Thất bại then begin Thông báo vô nghiệm ; exit ; end;
  If Có đ- ờng then Tiến
    Else Lui
Until false;

```

Sơ đồ giải tìm mọi nghiệm :

```

Khởi trị mảng chứa kết quả V thoả mãn điều kiện P
Repeat
  If gặp Đích then begin Hiện nghiệm ; Lui ; end;
  If Thất bại then begin Thông báo vô nghiệm ; exit ; end;
  If Có đ- ờng then Tiến
    Else Lui
Until false;

```

Bài mã đi tuần (Cách 3) Duyệt quay lui (backtracking) tìm mọi nghiệm , chỉ chạy đ- ợc với n khoảng 6,7

```

uses crt;const max = 7; fo = 'ma3.out';

```

```

dd      : array[1..8] of -2..2 = (-2,-2,-1,1,2,2,1,-1);
dc      : array[1..8] of -2..2 = (-1,1,2,2,1,-1,-2,-2);
type    ma      = array[-1..max+2,-1..max+2] of integer;
        mb      = array[1..max,1..max,1..8] of boolean;
        mt      = array[1..max,1..max] of integer;
var      a       : ma;
        b       : mb;
        tx,ty    : mt;
        f       : text;
        m,n,x,y,lx,ly,sb,sn,k,lk : integer;
procedure nhap;
begin
    write('nhap m,n = ');
    readln(m,n);
    write('nhap x,y = ');
    readln(x,y);
end;
procedure hangrao;
var i,j : integer;
begin
    for i:=-1 to m+2 do
        for j:=-1 to n+2 do a[i,j] := -1;
        for i:=1 to m do
            for j:=1 to n do a[i,j] := 0;
        end;
end;
procedure khoitri2;
var i,j,h,k : integer;
begin
    for i:=1 to m do
        for j:=1 to n do
            for k:=1 to 8 do b[i,j,k] := false;

            for i:=1 to m do
                for j:=1 to n do
                    begin
                        tx[i,j] := 0;
                        ty[i,j] := 0;
                    end;
                end;
            end;
        end;
end;
procedure hien;
var i,j : integer;
begin
    inc(sn);
    writeln(f,sn);
    for i:=1 to m do
        begin
            for j:=1 to n do write(f,a[i,j]:6);
            writeln(f);
        end;
    end;
end;
function tien_duoc(var x,y,sb : integer) : integer;
var u,v : integer;
begin
    tien_duoc := 9;

```

```

    for k:=1 to 8 do
    begin
        u := x+dd[k];
        v := y+dc[k];
        if a[u,v]=0 then
        if not b[x,y,k] then
        begin
            tx[u,v]:= x;
            ty[u,v]:= y;
            tien_duoc := k;
            b[x,y,k] := true;
            inc(sb);
            x := u;
            y := v;
            a[x,y] := sb;
            exit;
        end;
    end;
end;
procedure tongket;
begin
    if sn=0 then write(f,'vo nghiem ')
    else write(f,'tong so nghiem la : ',sn);
    close(f);
end;
procedure backtracking;
var lx : integer;
begin
    sb := 1;
    a[x,y] := 1;
    khoitri2;
    repeat
        if sb = m*n then hien;
        if sb < 1 then break;
        k := tien_duoc(x,y,sb);
        if not (k<9) then
        begin
            a[x,y] := 0;
            for k:=1 to 8 do b[x,y,k] := false;
            dec(sb);
            lx := x;
            x := tx[x,y];
            y := ty[lx,y];
        end;
    until false;
end;
BEGIN
    clrscr;
    nhap;
    hangrao;
    assign(f,fo);
    rewrite(f);
    backtracking;
    tongket;

```


Bài N hâu : Hãy xếp N quân hậu trên bàn cờ $N \times N$ sao cho chúng không khống chế nhau Thuật toán Backtracking.

```
uses crt;const      max    = 20;
      fo      = 'hau.out';
type  tv      = array[1..max] of byte;
var    v      : tv;
      d      : longint;
      f      : text;
      n      : byte;
procedure hien;
  var i : longint;
  begin
    writeln(f,'nghiem ',d);
    for i:=1 to n do write(f,v[i]:3);
    writeln(f);
  end;

procedure hienvn;
  begin
    writeln(f,'vo nghiem');
    close(f);
    halt;
  end;

function duoc(i : byte) : boolean;
  var j : byte;
  begin
    duoc := false;
    for j:=1 to i-1 do
      if (v[i]=v[j]) or (abs(v[i]-v[j])=i-j) then exit;
    duoc := true;
  end;

function tien(i : byte) : boolean;
  begin
    tien := true;
    while v[i]<n do
      begin
        inc(v[i]);
        if duoc(i) then exit;
      end;
    tien := false;
  end;

procedure backtracking;
  var i : byte;
  begin
    for i:=1 to n do v[i] := 0;
    i := 1;
    repeat
      if i>n then
        begin
```

```

        inc(d);
        hien;
    end;
    if i<1 then break;
    if tien(i) then inc(i)
    else
    begin
        v[i] := 0;
        dec(i);
    end;
    until false;
end;
BEGIN
    clrscr;
    write('nhap n = ');readln(n);
    if (n<1) or (n>max) then exit;

    assign(f,fo);
    rewrite(f);
    d := 0;

    backtracking;

    if d=0 then hienvn;
    close(f);
END.

```

Bài 6 : Tìm từ chân chính (chỉ gồm các kí tự thuộc tập $A=[\text{'1'..'9'}]$, không có 2 xâu con liên nhau bằng nhau) sao cho độ dài của từ bằng số nguyên N ($N \leq 40000$) và ký tự C thuộc tập A chỉ xuất hiện không quá K lần .

```

uses crt;
const  maxn  = 40000;
        fo    = 'pureword.out';
var    w      : array[1..maxn] of byte;
        n,k,dem : longint;
        len    : byte;
        sok    : longint;
        kituc  : Byte;
procedure init;
    var i : longint;
    begin
        for i:=1 to n do w[i] := 0;
        k := 1; { mới đầu từ chỉ có 1 ký tự }
        len := 3; { nghĩa là tập A =['1'..'3'] }
        dem := 0;
    end;
function equal(i,k : longint): boolean;
    var j : longint;
    begin
        equal := false;
        for j:= k downto k-i+1 do
            if w[j]<>w[j-i] then exit;
        equal := true;
    end;
function pure(k: longint): boolean;

```

```

var i : longint;
begin
  pure := false;
  for i:=1 to k div 2 do    { i : do dai 2 xau con lien nhau }
    if equal(i,k) then exit;
  pure := true;
end;
function k_tu_c(k : longint) : boolean;
var i,p : longint;
begin
  p := 0;
  k_tu_c := false;
  for i:=1 to k do
    begin
      if w[i]=kituc then inc(p);
      if p>sok then exit;
    end;
  k_tu_c := true;
end;
function coduong: boolean;
var i : longint;
begin
  coduong := true;
  for i:= w[k]+1 to len do
    begin
      w[k] := i;
      if pure(k) and k_tu_c(k) then exit;
    end;
  coduong := false;
end;
procedure pw;
var f : text;
procedure result;
var i : longint;
begin
  inc(dem);
  for i:=1 to n do
    begin
      write(f,w[i]);
      if i mod 80 =0 then writeln(f);
    end;
  writeln(f);
end;
procedure sum;
var i : longint;
begin
  if dem>0 then write(f,'tong so nghiem la : ',dem)
  else write(f,'vo nghiem');
end;
{ tim tat ca cac nghiem }
begin
  assign(f,fo);
  rewrite(f);
  repeat
    if k>n {dich} then result;

```

```

    if k<1 {thatbai} then break;
    if coduong and (k<=n) then inc(k) {tien}
    else {lui}
    begin
        w[k] := 0;
        dec(k);
    end;
until false;
sum;
close(f);
end;
{ Tim mot nghiem
begin
    assign(f,fo);
    rewrite(f);
    repeat
        if k>n (*dich*) then begin result;close(f);exit;end;
        if k<1 (*that bai*) then
            begin writeln(f,'vo nghiem ');close(f);exit;end;
        if coduong and (k<=n) then inc(k) (*tien*)
        else (*lui*)
            begin
                w[k] := 0;
                dec(k);
            end;
    until false;
    close(f);
end; }
BEGIN
    clrscr;
    write('do dai cua tu chan chinh la N = ');
    readln(N);
    write('ki tu lap la : ');readln(kituc);
    write('so lan lap la : ');readln(sok);
    init;
    PW;
END.

```

V Thuật toán khác :

Bài 4 : Cho N số nguyên d-ơng thuộc tập P , Hãy tìm tập con S của P sao cho với mọi số x trong P đều có thể biểu diễn d-ới dạng tích chỉ gồm các số thuộc tập con S .

Thuật toán tìm tập cơ sở (dùng dữ liệu kiểu queue)

```

program sinh;uses crt;const max = 10000; fi = 'input.inp'; fo =
'output.txt';type mang = array[1..max] of integer;
mang2 = array[1..max] of byte;
var a,q : mang;
dx : mang2;
n,m : integer;
f : text;
procedure docf;
var i : integer;
begin
assign(f,fi); reset(f);
readln(f,n);
for i:=1 to n do read(f,a[i]);
close(f);
end;
procedure qs(dau,cuoi : integer);
var i,j,g,coc :integer;
begin
i:=dau; j:=cuoi;
g:=a[(dau+cuoi) div 2];
repeat
while a[i]<g do inc(i);
while a[j]>g do dec(j);
if i<=j then
begin
coc:=a[i]; a[i]:=a[j]; a[j]:=coc;
inc(i); dec(j);
end;
until i>j;
if i<cuoi then qs(i,cuoi);
if j>dau then qs(dau,j);
end;
function duoc(k : integer) : boolean;
var dau,cuoi : integer;
i,p : integer;
begin
duoc:=true;
fillchar(dx,sizeof(dx),0);

```

```

dau:=0; cuoi:=1;
q[cuoi]:=k; dx[k]:=1;
while dau<cuoi do
begin
  inc(dau); k:=q[dau];
  for i:=1 to m do
    if k mod a[i]=0 then
      begin
        p:=k div a[i];
        if dx[p]=0 then
          begin
            inc(cuoi);
            q[cuoi]:=p;
            dx[p]:=1;
          end;
        if p=1 then exit;
      end;
  end;
  end;
  duoc:=false;
end;
procedure write_out;
var i : integer;
begin
  assign(f,fo); rewrite(F);
  writeln(F,m);
  for i:=1 to m do
    begin
      write(f,a[i] : 5);
      if i mod 16 =0 then writeln(F);
    end;
  close(f);
end;
procedure thuchien;
var i : integer;
begin
  qs(1,n);
  m:=1;
  for i:=2 to n do
    if not duoc(a[i]) then
      Begin
        Inc(m);
        a[m]:=a[i];
      end;
  write_out;
end;
BEGIN
  Clrscr;
  docf;
  thuchien;
END.

```

Bài 5 : Cho n số nguyên d-ơng đôi một khác nhau là tập S . Hãy chọn từ S một tập con P có ít phần tử nhất mà với mọi $(x,y) \mid x \in S, y \in P$ thì $\text{UCLN}(x,y) \neq 1$.
Thuật toán tìm tập ổn định ngoài.

```

uses crt;const max = 30;
      mang = array[0..max] of integer;
mang;
      g      : mang2;
      n,k    : integer;
      f      : text;
      dem    : longint;
procedure test;
  var f      : text;
      i,p    : integer;
  begin
    assign(f,fi);
    rewrite(f);
    n := 10;
    writeln(f,n);
    randomize;
    for i:=1 to n do
      begin
        p := random(100)+1;
        write(f,p:5);
        if i mod 20 = 0 then writeln(f);
      end;
    close(f);
  end;
procedure docf;
  var i,j : integer;
      f   : text;
  begin
    fillchar(a,sizeof(b),0);
    assign(f,fi);
    reset(f);
    readln(f,n);
    for i:=1 to n do read(f,b[i]);
    close(f);
  end;
function ucln(a,b : integer) : integer;
  var d : integer;
  begin
    if (a=0) and (b=0) then exit;
    while b>0 do
      begin
        d := a mod b;
        a := b;
        b := d;
      end;
    ucln := a;
  end;
procedure taodothi;
  var i,j : integer;
  begin
    for i:=1 to n-1 do
      for j:=i+1 to n do
        if ucln(b[i],b[j])<>1 then
          begin
            g[i,j] := 1;

```

```

fi      = 'ondinh2.inp';      fo      = 'ondinh2.out';type
mang2 = array[0..max,0..max] of 0..1;var  a,b      :

```

```

    g[j,i] := 1;
  end;
end;
Procedure tao_on_dinh_ngoai(i : integer);
  Var j : integer;
  procedure hien;
  var i : Byte;
  begin
    inc(dem);
    for i:=1 to k do
      write(f,b[a[i]]:4);
    writeln(f);
  end;
function od_ngoai (a : mang): Boolean;
  var x : integer;
  function khong_thuoc : boolean;
    var j : integer;
    begin
      for j:= 1 to k do
        if x = a[j] then
          begin khong_thuoc := false; exit; end;
        khong_thuoc := true;
      end;
  function noi : boolean;
    var j : integer;
    begin
      for j:=1 to k do
        if g[x,a[j]]=1 then
          begin noi := true; exit; end;
        noi := False;
      end;
  begin
    for x:=1 to N do
      if khong_thuoc then
        if not noi then
          begin od_ngoai := False; exit; end;
        od_ngoai := True;
      end;
  begin { Tao_on_dinh_ngoai(i) }
    if i>k then
      if od_ngoai(A) then hien;
    else { i<=k }
      for j:=A[i-1]+1 to N-k+i do
        begin
          A[i] := j;
          tao_on_dinh_ngoai(i+1);
        end;
    end;
  end;
procedure lam;
begin
  for k:=1 to n div 2 +1 do { xet bo on dinh ngoai k phan tu }
    begin
      dem := 0;
      fillchar(a,Sizeof(a),0);
      a[0] := 0;
    end;
  end;
end;

```



```

        tao_on_dinh_ngoai(1);
        if dem>0 then { ton tai bo on dinh ngoai kphan tu}
        begin
            writeln(f,dem,' nghiem ');
            { Writeln(F2,'So od ngoai la : ',k);}
            break; {chi tim bo on dinh ngoai nho nhat }
        end;
    end;
end;
BEGIN
    test;
    docf;
    taodothi;
    assign(f,fo);
    rewrite(f);
    lam;
    close(f);
END.

```

Bài 7 : Bài toán sắp ba lô : Cho n đồ vật , đồ vật thứ i có trọng lượng là w_i , giá trị là v_i .Ng-ời ta xếp các đồ vật vào 1 chiếc va ly có sức chứa tối đa là $limw$. Hãy chọn những đồ vật nào xếp vào va ly để giá trị va ly là lớn nhất .

Đây là bài toán tìm véc tơ $x = (x_1, x_2, \dots, x_n)$ với x_i chỉ nhận giá trị 0,1 , sao cho

$\sum x_i \cdot w_i \leq limw$ và $\sum x_i \cdot v_i$ đạt max .

{ xếp cac do vat vao va ly, moi loai chi chon toi da la 1 vat }uses crt;const $mn = 100;$
 $mw = 300;$

```

fi      = 'knapsack.inp';
fo      = 'knapsack.out';
type    tf      = array[0..mn,0..mw] of integer;
        twv     = array[1..mn] of integer;
        tkq     = array[1..mn] of byte;
var      f      : tf;          g      : text;          w,v      : twv;          tong      : integer;
        mt,luumt,n,limw : integer;
procedure docf;
  var i,j : integer;
      f : text;
  begin
    assign(f,fi);      reset(f);
    read(f,n,limw);
    for i:=1 to n do read(f,w[i]);
    for i:=1 to n do read(f,v[i]);
    close(f);
  end;
procedure hienf;
  var i,j : integer;
  begin
    write(n,' ',limw);writeln;
    for i:=1 to n do write(w[i]:4);writeln;
    for i:=1 to n do write(v[i]:4);writeln;
  end;
procedure taobang;
  var i,j : integer;
  function max2(x,y : integer) : integer;
  begin
    if x<y then max2 := y else max2 := x;
  end;
  begin
    for i:=0 to n do
      for j:=0 to limw do f[i,j] := -1;
    for i:=0 to n do
      for j:=0 to limw do f[i,j] := -1;
    for j:=0 to limw do f[0,j] := 0;

    for i:=0 to n do f[i,0] := 0;
    for i:=1 to n do
      for j:=1 to limw do
        begin
          if f[i,j]=-1 then
            if (j-w[i]>=0) then
              f[i,j] := max2(f[i-1,j],f[i-1,j-w[i]]+v[i])
            else f[i,j] := f[i-1,j];
        end;
      end;
  end;
procedure timkq(i,j : Integer);
  begin
    if (i<>0) and (j<>0) then
      begin
        if f[i,j]=f[i-1,j] then timkq(i-1,j)
        else
          begin
            writeln(g,'vat thu ',i:4,' : w =' :8,w[i]:4,'v =' :8,v[i]:4);

```

```

        timkq(i-1,j-w[i]);
        tong := tong+w[i];
    end;
end;
end;
BEGIN
    clrscr;
    docf;
    hienf;
    taobang;
    tong := 0;
    assign(g,fo);
    rewrite(g);
    timkq(n,limw);
    Writeln(g,'tong gia tri va ly : ',f[n,limw]);
    Writeln(g,'tong trong luong : ',tong);
    writeln('da chay xong chuong trinh ');
    close(g);
    readln;
END.

```

Trò chơi úp bài



Cho M quân bài mang các số từ 1 đến M ($M \leq 12$), các quân bài đang lật ngửa. Cho một số nguyên dương N ($N \leq 200$). Trò chơi như sau: Hai người lần lượt thay nhau úp quân bài theo qui tắc:

- + Cộng giá trị quân bài vào tổng điểm, nếu tổng điểm bằng N thì người đó thắng
- + Khi úp một quân bài (ngửa) thì đồng thời lật ngửa lại quân bài đang bị úp trước đó.

Hãy lập trình theo yêu cầu:

- 1) Nhập từ bàn phím số N, M.
- 2) Bốc thăm ai đi trước
- 3) Thể hiện trò chơi trên màn hình trò chơi giữa người và máy sao cho khả năng thắng của máy có thuận lợi hơn

Thuật toán:

Giả sử $N=10$, $M=3$. Trước hết lập bảng ph-ong án sau:

	1	2	3	4	5	6	7	8	9	10
1	1	0	0	1	1	0	0	1	1	0
2	0	0	0	1	0	0	0	1	0	0
3	0	0	1	1	0	0	1	0	0	0

Nếu máy đi trước:

Chọn quân số 1 (vì $A[1,1] = 1$), dồn người chơi phải chọn quân 2 hoặc 3, do đó cột điểm tiếp theo là $1+2=3$ hoặc $1+3=4$. Trong các cột điểm 3 và 4, đến lượt máy đi lại có số 1, nên máy lại đi chọn quân ở hàng nào đó có số 1... Quá trình cứ như thế, cho đến khi sẽ dẫn tới tình trạng: sau khi người đi quân số 2 hoặc 3 thì tổng điểm là 9 đến lượt máy đi, máy úp quân số 1, đi-ợc tổng điểm là 10. Máy thắng.

Nếu máy đi sau:

Rất có thể máy bị dồn vào tình trạng : nhận cột điểm không có số 1 . Khi đó máy phải úp quân nào đó để cột điểm mới có ít số 1 nhất , nghĩa là tạo ra tình thế bất lợi nhất cho ng- ời (Máy hy vọng ng- ời chơi này không biết qui luật , úp phải quân bài ở hàng 0 của cột điểm mới này)

Vấn đề còn lại các em sẽ thắc mắc là : *Làm thế nào có bảng ph- ơng án nh- vậy ?*

Lý do đơn giản là chúng ta lần ng- ọc từ trạng thái kết thúc chắc thắng về trạng thái đầu . Cụ thể

+ Gán $A[1,N-1] = 1$

+ Sau đó xây dựng dần các số 1 ở các cột điểm $d = N-2, N-3, \dots, 1$ theo qui tắc :

Chọn số quân lần l- ợt là $S_q = 1 \dots M$. Gọi số l- ợng số 1 ở cột $d+S_q$ là x (với điều kiện $x \leq N$) .
Nếu $x=0$ hoặc ($x=1$ và $A[S_q,x]=1$) thì $A[S_q,d]=1$; còn lại $A[S_q,d]=0$

CHƯƠNG TRÌNH

```
Uses Crt;
Type pt = 0..1;
Var Diem,sq,m,n,Luu : Byte;
    S : String;
    A : Array[1..12,0..200] of 0..1;
    Ch : Char;
    Ok : Boolean;
Procedure Ve(i,j : Byte;Ch : Char);
Var k,h : Byte;
Begin
    Textcolor(7);
    If j<>0 then
        For k:=i to i+4 do
            For h := j to j+4 do
                Begin
                    Gotoxy(h,k);
                    Write(ch);
                End;
            Textcolor(14);
        End;
End;
Procedure Nhap;
Begin
    Repeat
        Clrscr;
        Write('So diem toi da ( N<= 200), N = ');
        {$I-} Readln(N); {$I+}
    Until (Ioresult=0) and (N in [1..200]);
    Repeat
        Gotoxy(1,2);
        Write('So quan bai ( M<=12 ) , M = ');
        {$I-} Readln(M); {$I+}
    Until (Ioresult=0) and (M in [1..12]);
End;
Function Sl_dau(diem : Byte) : Byte;
Var d,j : Byte;
Begin
    d := 0;
    For j:=1 to M do
        If A[j,diem]=1 then Inc(d);
    SL_dau := d;
End;
Function Thang(i,t : Byte) : pt;
Var j,p : Byte;
```

```

Begin
  p := SL_dau(i+t);
  If p>1 then Thang := 0
  Else
    If p=0 then Thang := 1
    Else
      If p=1 then
        Begin
          If A[t,i+t]=1 then Thang := 1
          Else Thang := 0;
        End;
      End;
  End;
End;
Procedure Taobang;
Var i,j : Byte;
Begin

  For sq:=1 to M do
    Begin
      Ve(5,sq*6,char(219));
      Gotoxy(sq*6+2,10);
      Write(sq);
    End;

    FillChar(A,Sizeof(A),0);
    A[1,N-1] := 1;

    For j:=N-2 downto 0 do
      For i:=1 to M do
        If (i+j<=N) and (Thang(j,i)=1) then A[i,j] := 1;

        { A[1,1] := 0; }
        Diem := 0;
        Luu := 0;
      End;
    End;
  End;
Procedure Boctham;
Begin
  Gotoxy(20,16);
  Write('Ban chon di truoc hay di sau (T/S) ? ');
  Repeat
    Ch := Upcase(Readkey);
  Until Ch in ['T','S'];
  Gotoxy(20,16);
  Clreol;
End;
Procedure GhiMaydi(sqm,diem : Byte);
Begin
  Gotoxy(50,16); Write('May up quan bai so ',sqm:3);
  Gotoxy(20,18); Textcolor(12);
  Write('Tong so diem ',diem:6);
  Textcolor(14);
End;
Procedure May_choi;
Var k,x : Byte;
Begin
  { Tinh huong tot }

```

```

For k:=1 to M do
  If (k<>Luu) and (A[k,diem]=1) then
    Begin
      Ve(5,luu*6,char(219));{ Lat bai cua nguoi }
      Luu := k;
      Ve(5,luu*6,char(176));{ May up quan moi }
      Inc(diem,k);
      Ghimaydi(k,diem);
      Exit;
    End;
  { Tinh huong xau : chon cot co it hang co dau }
  { de hy vong nguoi kia boc dung hang khong dau }
  x := M;
  For k:=1 to M do
    If k<>Luu then
      If (SL_dau(k+diem)<x) then x := k;
      Ve(5,luu*6,char(219)); { Lat bai cua nguoi }
      Luu := x;
      Ve(5,luu*6,char(176));{ May up quan bai moi }
      Inc(diem,x);
      Ghimaydi(x,diem);
  End;
Procedure Nguoidi;
Var Ch : Char;
Begin
  Gotoxy(1,24);
  Write('Ban chon quan bai bang cach chuyen mui ten ',char(24));
  Repeat
    Gotoxy(sq*6+2,11);
    Writeln(char(24));{ Viet mui ten len }
    Ch := Upcase(Readkey);
    Gotoxy(sq*6+2,11);
    Write(chr(32)); { Xoa mui ten len }

    Case ch of
      'K' : If sq>1 then Dec(sq) Else sq := m;
      'M' : If sq<m then Inc(sq) Else sq := 1;
    End;
  Until (sq<>Luu) and (Ch=#13);
  Gotoxy(1,16);Write(' ');
  Gotoxy(1,16);
  Write('Ban vua up quan = ',sq);
  Inc(diem,sq);
  If Luu>0 then Ve(5,luu*6,char(219));
  Luu := sq;
  Ve(5,luu*6,char(176));
  Delay(1000);
End;
BEGIN
  Textcolor(14);
  TextBackGround(1);
  Repeat
    Nhap;
    Taobang;
    Boctham;

```

```

If ch='T' then nguoidi;
Ok := False;
If diem<=N then
Repeat
    May_choi;
    If Diem<=N then Nguoidi Else Ok := True;
Until Diem>N;
Clrscr;
Gotoxy(20,20);
If Ok then Writeln('Ban thang ! ') Else Writeln('May thang ! ');
Gotoxy(40,20); Write('ESC to quit ...');
Until Readkey=#27;
END.

```

MỘT SỐ BÀI TOÁN VUXỚP LỊCH

Bài 1 : Cho N công việc , mỗi công việc i phải làm tr-ớc một số công việc jk1.. j k2 ..j ks nào đó trong N công việc này . Hãy xếp lịch thực hiện các công việc này .

Bài 2 : Cho N công việc . Mỗi công việc i phải làm sau một số công việc jk1.. j k2 ..j ks nào đó trong N công việc này và biết thời gian thực hiện công việc là ti . Xếp lịch thực hiện nhiều công việc nhất .

Bài 3 : Cho N công việc . Mỗi công việc i cho biết thời gian thực hiện công việc là ti

- Tính thời gian min thực hiện đủ N công việc
- Cho thời điểm cuối phải hoàn thành mỗi công việc i này là Ci . Có thể xếp lịch thực hiện N công việc hay không (Thông báo "có" hay "không")
- Nếu kết quả câu b) là "không" thì xếp đ-ợc nhiều công việc nhất là bao nhiêu ?

Thuật toán tham lam

(Bài làm của Lê Sỹ Vinh 12 CT Lê Quý Đôn- Giải nhất Tin học Quốc tế 1998)

```

{$A+,B-,D+,E+,F-,G-,I+,L+,N-,O-,P-,Q-,R+,S+,T-,V+,X+}
{$M 16384,0,655360}

```

```

Uses Crt;
Const max =1000;
    Input ='Input.txt';
    Output ='Output.txt';

```

```

Type Mang =array[1..max] of Integer;
Var C, T , Tt , Kq, Lkq, Tt2 : Mang;
    N, Sl : Integer;

```

```

Procedure Read_Input;
Var f : text;
    i : Integer;
Begin
    Assign(f, Input); Reset(F);
    Readln(F, N);
    For i:=1 to N Do Read(f, T[i]); Readln(F);
    For i:=1 to N Do Read(f, C[i]); Readln(F);

```

```
CLOSE(f);
End;
```

Procedure Solution1;

```
Var i, Tmin : Longint;
F : text;
begin
  Tmin:=0;
  For i:=1 to N Do Tmin:=Tmin+ T[i];

  Assign(F, Output); Rewrite(f);
  Writeln(F, Tmin);
  Close(F);
End;
```

Function Kiemtra(k : Integer) : boolean;

```
{ Tap Hop Co K cong Viec Co Thoa Man Hay Khong }
Var i, Now, Sh : Longint;
Begin
  Kiemtra:=False;
  Now:=0;
  For i:=1 to K Do
    Begin
      Sh:=Tt[i];
      Now:=Now+ T[Sh];
      If Now>C[Sh] THEN Exit;
    End;
  Kiemtra:=True;
end;
```

Procedure Solution2;

```
Var i,j, Coc : Integer;
F : text;
Begin
  { Sap Sep Theo C[i] }
  For i:=1 to N Do Tt[i]:=i;
  For i:=1 to N Do
    for j:=i+1 to N Do
      If C[ Tt[i] ]> C[ Tt[j]] Then
        Begin
          Coc:=Tt[i]; Tt[i]:=Tt[j]; Tt[j]:=Coc;
        End;

  Assign(f, Output); Append(f);
  If Kiemtra(N) Then WRitEln(F,'CO')
  Else WRitEln(F,'KHONG');
  CLose(F);
End;
```

function ThoaMan : Boolean;

```
Var i, j, Coc : Integer;
begin
  For i:=1 to Sl Do Tt[i]:= Kq[i];
  { Sap Sep Theo C[i] }
```



```

For i:=1 to Sl Do
  for j:=i+1 to Sl Do
    If C[ Tt[i] ]> C[ Tt[j]] Then
      Begin
        Coc:=Tt[i]; Tt[i]:=Tt[j]; Tt[j]:=Coc;
      End;
  ThoaMan:=Kiemtra(Sl);
End;

```

```

Procedure Solution3;
Var i,j , Coc : Integer;
    F : text;
Begin
  { Sap Sep Theo T[i] }
  For i:=1 to N Do Tt2[ i]:=i;
  For i:=1 to N Do
    for j:=i+1 to N Do
      If T[ Tt2[i] ]> T[ Tt2[j]] Then
        Begin
          Coc:=Tt2[i]; Tt2[i]:=Tt2[j]; Tt2[j]:=Coc;
        End;
    Sl:=0; { Kq Bang Rong }
    For i:=1 to N Do
      Begin
        Lkq:=Kq;
        Inc(Sl); Kq[Sl]:= Tt2[i];
        If ThoaMan=false THEN
          Begin
            Kq:=Lkq; Sl:=Sl-1;
          End;
      End;
    End;

    Assign(f, Output); Append(F);
    WRiteln(F, Sl);
    CClose(f);
  End;

```

```

BEGIN
  Clrscr;
  Read_Input;
  Solution1;
  Solution2;
  Solution3;
END.

```

```

Input.txt
Output.txt
4
1 3 11 1
3 4 15 8

```

Bài 4 : Cho N công việc ,với mỗi công việc cho thời điểm bắt đầu có thể thực hiện , thời gian thực hiện , thời điểm tối đa phải kết thúc . Xếp lịch để thực hiện đ- ợc nhiều công việc nhất .

```
{ $A+,B-,D+,E+,F-,G-,I+,L+,N-,O-,P-,Q-,R+,S+,T-,V+,X+ }
```

```
{ $M 16384,0,655360 }
```

```
Uses crt;
```

```
Const Input ='Viec.Inp';
```

```
Output ='viec.out';
```

```
max :=51;
```

```
Type Kieu =Record
```

```
dau,tg,cuoi : Integer;
```

```
Tt : Byte;
```

```
End;
```

```
Mang =Array[0..max] of Kieu;
```

```
Ta :=Array[1..max] of Byte;
```

```
Var a , kq, lkq: mang;
```

```
Cx : Ta;
```

```
N , maxviec, viec , conlai, time: Integer;
```

```
Procedure Nhap;
```

```
Var f : text;
```

```
i : Byte;
```

```
Begin
```

```
Assign(f,Input); Reset(F);
```

```
Readln(f,N);
```

```
For i:=1 to N Do Readln(f,A[i].dau,a[i].tg,a[i].cuoi);
```

```
Close(F);
```

```
End;
```

```
Procedure Sapsep; { Sap xep theo thoi diem bat dau , tang dan }
```

```
var i,j : Byte;
```

```
Begin
```

```
For i:=1 to N Do A[i].tt:=i;
```

```
For i:=1 to N Do
```

```
For j:=i+1 to N Do
```

```
If A[i].dau>A[j].dau Then
```

```
Begin
```

```
A[max]:=A[i]; A[i]:=A[j]; A[j]:=A[max];
```

```
End;
```

```
End;
```

```
Function Ln(k,t : integer) : Integer;
```

```
Begin
```

```
if k>t Then ln:=k
```

```
Else ln:=t;
```

```
End;
```

```
Procedure Lay(k : Byte);
```

```
Var i : Byte;
```

```
Begin
```

```
Dec(conlai);
```

```
Cx[k] := k;
```

```
Inc(viec);
```

```
Kq[viec].tt := k;
```

```
Kq[viec].dau := Ln(a[k].dau,time);
```

```
Kq[viec].cuoi:= kq[viec].dau+A[k].tg;
time:=kq[viec].cuoi;
```

```
For i:=1 to N Do
  If (Cx[i]=0) And (Time+A[i].Tg>A[i].Cuoi) Then
    Begin
      Cx[i]:=k; Dec(Conlai);
    End;
```

```
End;
```

```
Procedure Bo(k : Byte);
```

```
Var i : Byte;
```

```
Begin
```

```
  Inc(Conlai);
```

```
  Dec(Viec); Cx[k]:=0;
```

```
  For i:=1 to N Do
```

```
    If (Cx[i]=k) Then
```

```
      Begin
```

```
        Cx[i]:=0; Inc(Conlai);
```

```
      End;
```

```
End;
```

```
Procedure Perfect;
```

```
Begin
```

```
  maxviec:=Viec; Lkq:=Kq;
```

```
End;
```

```
Function Dao : boolean;
```

```
Var Tg1,x1,x2 : Integer;
```

```
Begin
```

```
  Tg1:=kq[viec-2].Cuoi;
```

```
  x1:=kq[viec-1].tt; x2:=kq[viec].tt;
```

```
  Tg1:=ln(Tg1,A[x2].dau)+a[x2].tg;
```

```
  Tg1:=Ln(Tg1,A[x1].dau)+A[x1].Tg;
```

```
  Dao:=true;
```

```
  If (Tg1<=A[x1].Cuoi) And (Tg1<=Kq[Viec].Cuoi) Then
```

```
    Begin
```

```
      If (Tg1<Kq[Viec].Cuoi) then Exit;
```

```
      If (x2<x1) then Exit;
```

```
    End;
```

```
    Dao:=False;
```

```
End;
```

```
Function Ktcan : Boolean;
```

```
Var i,tg1,tg2 : Integer;
```

```
Begin
```

```
  ktcan:=False;
```

```
  If Conlai+Viec<=maxviec Then Exit;
```

```
  If (viec>=2) Then
```

```
    If Dao Then Exit;
```

```
  Tg1:=Kq[Viec-1].Cuoi; Tg2:=Kq[Viec].Dau;
```

```
  For i:=1 to N Do
```

```
    If Cx[i]=0 Then
```

```

    If ln(tg1,A[i].Dau)+A[i].Tg<Tg2 Then Exit;
    Ktcan:=True;
End;

```

```

Procedure Vet;
Var i,tg : Integer;
Begin
    Tg:=Time;
    For i:=1 to N Do
        If (cx[i]=0) and (Time+A[i].tg<=A[i].Cuoi) Then
            Begin
                Lay(i);
                If Viec>maxviec Then Perfect;
                IF ktcan Then Vet;
                time:=Tg;
                bo(i);
            End;
    End;
End;

```

```

Procedure Bailam;
Begin
    Fillchar(Cx,Sizeof(Cx),0);
    maxviec:=0; viec:=0;
    Time:=0; Conlai:=N; Kq[0].Cuoi:=0;
    Vet;
End;

```

```

Procedure Hienkq;
Var f : text;
    i : Byte;
Begin
    Assign(F,Output); ReWrite(f);
    Writeln(F,maxviec);
    For i:=1 to maxviec Do
        Writeln(F,A[Lkq[i].tt].tt,' ',Lkq[i].dau,' ',Lkq[i].Cuoi);
    Close(F);
End;

```

```

Procedure Taofile;
Var f :text;
    i,tg,dau,Cuoi : Integer;
Begin
    Write('NHAP N = '); Readln(N);
    Randomize;
    Assign(F,Input); ReWrite(F);
    Writeln(f,N);
    For i:=1 to N Do
        Begin
            Dau:=Random(10); Cuoi:=Dau+Random(100);
            Tg:=Random(Cuoi-dau)+1;
            Writeln(F,Dau,' ',tg,' ',Cuoi);
        End;
    Close(f);
End;

```

```

begin
  Clrscr;
  {  Taofile; }
  Nhap;
  Sapsep;
  bailam;
  Hienkq;
End.

```

Bài 5 : Cho N công việc ,với mỗi công việc cho giá trị của công việc (tính bằng đơn vị tiền) , thời gian thực hiện , thời điểm cuối cùng phải kết thúc . Xếp lịch để thực hiện đ- ọc nhiều tiền công nhất .

```

Uses Crt;
Const   Max   = 60;
        Fi    = 'v2.INP';
        Fo    = 'CV3.OUT';
Type    PT    = Record
          Thoigian,Tien,Ketthuc,Ten:Byte; { Thoi gian,Tien,ten,ketthuc }
        End;
Var     A,Q,LQ : Array[1..Max]of PT;
        D      : Array[1..Max]of Byte;
        N,top,Ltop : Byte;
        Tien,Thoidiem,TongTien : Integer;
        Conlai   : LongInt;
Procedure Input;
  Var    F    : Text;
         k    : Byte;
  Begin
    FiLLChar(A,Sizeof(A),0);
    FiLLChar(D,Sizeof(D),0);
    Assign(F,Fi);
    Reset(F);
    ReadLn(F,N);
    Conlai:=0;
    For k:=1 to N do
      Begin
        ReadLn(F,A[k].Thoigian,A[k].Ketthuc,A[k].Tien);
        A[k].Ten:=k;
        Conlai:=Conlai+A[k].Tien;
      End;
    Close(F);
  End;
Procedure Trao(Var u,v:PT);
  Var Coc : PT;
  Begin
    Coc := u;
    u   := v;
    v   := Coc;
  End;
Procedure Xap_xep;
  Var i,j : Byte;
  Begin
    For i:=1 to N-1 do
      For j:=i+1 to N do
        If A[i].Ketthuc>A[j].Ketthuc then Trao(A[i],A[j]);

```

```

End;
Procedure Lay(k:Byte);
  Var j : Byte;
  Begin
    Tien := Tien+A[k].Tien;
    D[k] := k;
    Conlai := Conlai-A[k].Tien;
    Inc(top);
    Q[top].Thoigian := Thoidiem; { Thoi gian truoc khi lam k }
    Thoidiem := Thoidiem+A[k].Thoigian;

    Q[top].Ten := k;
    Q[top].Ketthuc := Thoidiem; { Thoi gian sau khi lam k }
    For j:=1 to N do
      If (D[j]=0)And(A[j].Ketthuc<Thoidiem)then
        Begin
          D[j] := k;
          Conlai := Conlai-A[j].Tien;
        End;
    End;
End;
Procedure Thao(k:Byte);
  Var j : Byte;
  Begin
    For j:=1 to N do
      If D[j]=k then
        Begin
          D[j] := 0;
          Conlai := Conlai+A[j].Tien;
        End;
      Thoidiem := Thoidiem-A[k].Thoigian;
      Tien := Tien-A[k].Tien;
      Dec(top);
    End;
Function Can:Boolean;
  Begin
    Can := True;
    If Conlai+Tien<=Tongtien then Exit;
    Can := False;
  End;
Procedure Luu_KQ;
  Begin
    LQ:=Q;
    Tongtien := Tien;
    Ltop := Top;
  End;
Procedure Try;
  Var k : Byte;
  Begin
    For k:=1 to N do
      If (D[k]=0)And(Thoidiem+A[k].Thoigian<=A[k].Ketthuc) then
        Begin
          Lay(k);
          If Tien>Tongtien then Luu_KQ;
          If Can then Exit;
        End;
    Try;
  End;

```

```

        Thao(k);
        End;
    End;
Procedure Output;
    Var    F    : Text;
           k    : Byte;
    Begin
        Assign(F,Fo);
        ReWrite(F);
        WriteLn(F,Tongtien);
        For k:=1 to Ltop do
            Begin
                Write(F,A[LQ[k].Ten].Ten:4,A[LQ[k].Ten].Thoigian:4,
                    A[LQ[k].Ten].ketthuc:4);
                WriteLn(F,' ',LQ[k].Thoigian:6,LQ[k].Ketthuc:4,
                    A[LQ[k].Ten].Tien:6);
            End;
        Close(F);
    End;
BEGIN
    Thoidiem := 0;
    { Test; }
    Input;
    Try;
    Output
END.

```

Bài 6 : (Đề thi chọn đội tuyển quốc gia năm 1995 . Bài 2 ngày 25-4-1995)

Trong một trường đại học có M thầy giáo đánh số từ 1 đến M và N lớp học đánh số từ 1 đến N . Với $1 \leq i \leq M$, $1 \leq j \leq N$, thầy i phải dạy cho lớp j P[i,j] ngày , P[i,j] là số nguyên trong khoảng từ 0 đến 10 . Trong mỗi ngày mỗi thầy không dạy hơn 1 lớp và mỗi lớp không học hơn một thầy .Hãy thu xếp lịch cho các thầy giáo sao cho toàn bộ yêu cầu giảng dạy trên đ- ợc hoàn thành trong số ngày ít nhất .Các ngày trong lịch dạy đánh số lần l- ợt là 1,2,3,...

Đọc thông tin từ một File văn bản tên là INP.B2 ,trong đó dòng đầu ghi lần l- ợt giá trị M và giá trị N ($M \leq 20, N \leq 20$) , dòng thứ i+1 ($1 \leq i \leq M$) ghi lần l- ợt N giá trị P[i,1],P[i,2],...,P[i,n] là các số nguyên trong khoảng 0 đến 10 .Hai giá trị liên nhau trên một dòng cách nhau ít nhất một dấu trắng .

Lời giải ghi ra File văn bản có tên là OUT.B2 , trong đó dòng thứ nhất ghi số ngày hoàn thành toàn bộ khối l- ượng giảng dạy , trong các dòng tiếp theo lần l- ợt từ ngày 1 , ghi theo quy cách theo thí dụ d- ưới đây , mỗi dòng lịch dạy trong ngày đó của các thầy , lần l- ợt từ thầy 1 , nếu thầy nào không dạy không ghi ra Ví dụ với File dữ liệu

```

4
2      0      0      0
0      1      1      0
1      0      1      0
1      1      1      1
0      0      0      1

```

File kết quả có thể có nội dung nh- sau :

Số ngày : 4

Ngày 1 : Thầy 2 dạy lớp 2 , Thầy 3 dạy lớp 3, Thầy 4 dạy lớp 1,

Ngày 1 : Thầy 1 dạy lớp 1, Thầy 2 dạy lớp 3, Thầy 4 dạy lớp 2,

Ngày 1 : Thầy 3 dạy lớp 1, Thầy 4 dạy lớp 3, Thầy 5 dạy lớp 4,

Ngày 1 : Thầy 1 dạy lớp 1, Thầy 4 dạy lớp 4,

{ \$A+,B-,D+,E+,F-,G-,I+,L+,N-,O-,P-,Q-,R-,S+,T-,V+,X+ }

{ \$M 56384,0,655360 }

Program Thay_giao;

Uses crt;

const max=20;

Max1=200;

Fi='Thaygiao.inp';

Fo='Thaygiao.out';

Type mang=array[1..max,1..max] of integer;

mang2=array[1..max1,1..max] of byte;

mang3=array[1..max] of integer;

Mang4=array[1..max1] of integer;

Var A : mang;

Lop,kq : mang2;

dong,cot : mang3;

TT : mang4;

M,n,snc,sn : integer;

Time : longint;

F : text;

Procedure read_inp;

var i,j : integer;

begin

Assign(f,fi);

reset(F);

readln(f,m,n);

for i:=1 to m do

Begin

for j:=1 to n do

read(f,A[i,j]);

readln(F);

end;

Close(f);

end;

Function max_arr(var A:mang3; n : integer) : integer;

var i,ma : integer;

Begin

ma:=0;

for i:=1 to n do

If A[i]>ma then

Ma:=A[i];

Max_arr:=ma;

end;

Function Songay : integer;

var d,c : integer;

Begin

d:=max_arr(dong,m);

C:=max_arr(cot,n);

If d>c then songay:=d

else songay:=c;

end;

function Ok : boolean;

var i,j : integer;

Begin

Ok:=false;

for i:=1 to m do

for j:=1 to n do

If a[i,j]<>0 then exit;

Ok:=true;

end;

Procedure Write_out;

var i,j : integer;

Begin

Assign(f,fo);

rewrite(F);

Writeln(f,snc);

for i:=1 to snc do

Begin

Write(f,'Ngày ',i,' ');

for j:=1 to m do

If Kq[i,j]<>0 then

Write(f,j,'/',Kq[i,j],' ');

Writeln(f);

end;

Close(F);

Writeln((meml[0:\$46C]-time) /18.2 : 8: 2);

halt;

end;

Procedure try(sngay,sthay : integer);

var i,j : integer;

Begin

if sngay>snc then

Begin

If Ok then Write_out;

Exit;

end;

If sthay>m then

Begin

If (sngay+Sngay<>Snc) then exit;

try(sngay+1,1);

exit;

end;

for i:=1 to n do

If (A[Sthay,i]>0) and (Lop[sngay,i]=0) then

Begin

Dec(A[Sthay,i]);

Lop[sngay,i]:=1;

dec(dong[sthay]);

dec(Cot[i]);

kq[sngay,sthay]:=i;

try(sngay,sthay+1);

kq[sngay,sthay]:=0;

```

        inc(dong[sthay]);
        inc(Cot[i]);
        Lop[sngay,i]:=0;
        inc(A[Sthay,i]);
    end;
    try(sngay,sthay+1);
end;

```

```

Procedure Init_data;
var i,j : integer;
begin
    Fillchar(Lop,sizeof(lop),0);
    for i:=1 to m do
        Begin
            dong[i]:=0;
            For j:=1 to n do
                Dong[i]:=Dong[i]+A[i,j];
            end;
        for j:=1 to n do
            begin
                cot[j]:=0;
                for i:=1 to n do
                    Cot[j]:=Cot[j]+A[i,j];
                end;
            Snc:=songay;
            Fillchar(tt,sizeof(tt),0);
        end;
    end;
end;

```

```

Procedure Solution;
begin
    init_data;
    try(1,1);
end;

```

```

BEGIN
    Clrscr;
    Time:=meml[0:$46C];
    Read_inp;
    Solution;

```

END.

```

5 4
2 0 0 0
0 1 1 0
1 0 1 0
1 1 1 1
0 0 0 1

```

Bài 7 : (Bài 1 - thi quốc tế 1996 — Tại Hunggari)

Một nhà máy chạy một dây chuyền sản xuất . Có 2 nguyên công cần phải thực hiện đối với mỗi sản phẩm theo trình tự sau : đầu tiên là nguyên công A , sau đó tới nguyên công B . Có một số máy

```

Program Jobs;
Const
    MaxM=30;           { max number of machines }
Type
    Operation='A'..'B';
    ProcTime=Array[Operation,1..MaxM] Of Word;
Var
    N:Longint;          { number of jobs }
    M:Array[Operation] Of Word; { M[op] is the number of machines of type op }

```

PTime: ProcTime; { PTime[op,m] is the processing time for machine
m of type op }
TA, { the time needed to perform single operation A on all N jobs }
TB: Longint; { the time needed to perform single operation B on all N jobs }
d :Longint;

Procedure ReadInput;
{ Global output variables: N, M, PTime }
Var InFile: Text; i: Word;
Begin
Assign(InFile, 'input.txt'); Reset(InFile);
ReadLn(InFile,N);
ReadLn(InFile,M['A']);
For i:=1 To M['A'] Do
Read(InFile, PTime['A',i]);
ReadLn(InFile);
ReadLn(InFile,M['B']);
For i:=1 To M['B'] Do
Read(InFile, PTime['B',i]);

Close(InFile);
End { ReadInput};

Function Compute_Time(Op:Operation):Longint;
{ Computes the minimal time that is needed to perform operation Op on N jobs }
{ Global input variables: M, PTime }
Var t,Processed:Longint;
i:Word;
Begin
t:=0;
Repeat
Inc(t);
Processed:=0;
For i:=1 To M[Op] Do
Processed:=Processed+(t Div PTime[Op,i]);
Until Processed>=N;
Compute_Time:=t;
End;{ Compute_Time }

Function Finish(Op:Operation; t: Longint): Longint;
{ Finish(Op,t) is the number of jobs that are finished at time t
according to the optimal schedule for single operation Op for N jobs. }
{ Global input variables: N, M, PTime }
Var Res,UpTo: Longint;
i: Word;
Begin
Res:=0;
For i:=1 To M[Op] Do
If (t Mod PTime[Op,i])=0 Then Inc(Res);
{ If the number of jobs that can be completed up to time t
is more then N then decrease Res to the proper value. }
UpTo:=0;
For i:=1 To M[Op] Do UpTo:= UpTo+ (t-1) Div PTime[Op,i];
If UpTo >= N Then
Res:= 0

```

Else If Upto+Res>N Then
  Res:= N-UpTo;
  Finish:=Res;
End {Finish};

```

Procedure Adjust;

```

{ Computes the delay time d when the first type B machine starts to work }
{ Global input variables: TA, TB }
{ Global output variables: d }
Var Inter:Word;{ number of jobs in the intermediate container }
  t: Longint;
  JB:Word;
Begin
  d:=1; t:=0; Inter:=0;
  While d+t<TA Do Begin
    Inter:=Inter+Finish('A',d+t);
    JB:=Finish('B',TB-t); { # jobs starting at time d+t }
    While Inter<JB Do Begin { while not enough jobs available }
      Inc(d);
      Inter:=Inter+Finish('A',d+t);
    End;
    Inter:=Inter-JB;
    Inc(t);
  End;
End;{ Adjust }

```

Procedure WriteOut(AnswerA,AnswerB:Longint);

```

Var OutFile: Text;
Begin
  Assign(OutFile, 'output.txt'); Rewrite(OutFile);
  WriteLn(OutFile, AnswerA);
  WriteLn(OutFile, AnswerB);
  Close(OutFile);
End;{ WriteOut }

```

Begin {Main}

```

  ReadInput;
  TA:= Compute_Time('A');
  TB:= Compute_Time('B');
  Adjust;
  WriteOut(TA, d+TB);
End.

```

Solution 2 :

Uses Crt;

Const Mn = 1000;

Fi = 'input-4.txt';

Fo = ";

Type Ta = Array[1..mn] of Byte; { Thời gian xử lý từng máy }

```
Var N : Integer; { So san pham <=1000 }
```

```
M1,M2 : Byte; { Soluong may tung loai A,B <=30 }
```

```
T1,T2 : Ta;
```

```
F : Text;
```

```
tg : Integer;
```

```
Procedure DocF;
```

```
Var F : Text;
```

```
i : Integer;
```

```
Begin
```

```
Assign(F,Fi);
```

```
{ $i- } Reset(F); { $I+ }
```

```
If IoResult<>0 then
```

```
Begin
```

```
Writeln('Loi Ffile ');
```

```
Readln;
```

```
Halt;
```

```
End;
```

```
Readln(F,N);
```

```
Readln(F,M1);
```

```
For i:=1 to M1 do Read(F,T1[i]);
```

```
Readln(F);
```

```
Readln(F,M2);
```

```
For i:=1 to M2 do Read(F,T2[i]);
```

```
Close(F);
```

```
End;
```

```
Function spht(X : Ta;m,tg : Integer):Integer;
```

```
Var sp,i : Integer;
```

```
Begin
```

```
sp := 0;
```

```
For i:= 1 to m do sp:=sp+tg div X[i];
```

```
spht := sp;
```

```
End;
```

```
Function Thoigian(X : Ta;m: Integer): Integer;
```

```
Var tg,sp : Integer;
```

```
Begin
```

```
tg := 0;
```

```
sp := 0;
```

```
While sp<N do
```

```
Begin
```

```
Inc(tg);
```

```
sp := spht(X,m,tg);
```

```
End;
```

```
Thoigian := tg;
```

```
End;
```

```
Procedure Tinh;
```

```
Var i,x,tg : Integer;
```

```
Function Conthieu(tgthieu : Integer): Integer;
```

```
Var lam,i : Integer;
```

```
Begin
```

```
conthieu := N - spht(T2,m2,tg-tgthieu-1);
```

```
End;
```

```
Begin
```

```
tg := Thoigian(T2,m2);
```

```

x := 0;
For i:=0 to tgb-1 do
    While spht(T1,M1,i+x)<conthieu(i) do Inc(x);
Tgb := Tgb+x;
Writeln(F,Tgb);
End;
Procedure Lam;
Var ds_caua : Integer;
Begin
    Assign(F,Fo);
    Rewrite(F);
    Ds_caua := Thoigian(T1,m1);
    Writeln(F,Ds_caua);
    Tinh;
    Close(F);
End;
BEGIN
    Clrscr;
    DocF;
    Lam;
END.

```

Bài toán 8 : (Ph- ơng pháp đệ quy , vét cạn tìm nghiệm tối - u)

Cho N công việc (mã số từ 1 đến N) và M nhóm thợ (mã số từ 1 đến M) ($0 < N, M < 100$). Thuê thợ theo nguyên tắc phải thuê toàn nhóm và sao cho n công việc đều đ- ợc thực hiện với 2 tr- ờng hợp sau :

Câu a : Số nhóm thợ phải thuê là ít nhất

Câu b : Số thợ thuê là ít nhất

Dữ liệu vào từ File 'nhomtho.inp'

Dòng đầu là 2 số n, m

Trong m dòng tiếp theo : số đầu tiên của dòng i trong m dòng này là số thợ của nhóm i , các số tiếp theo của dòng là các mã số của các công việc mà nhóm này có thể làm .

Dữ liệu ra trên màn hình :

Câu a : các mã số là tên các nhóm thợ đ- ợc thuê trong tr- ờng hợp A

Câu b : các mã số là tên các nhóm thợ đ- ợc thuê trong tr- ờng hợp B

Thí dụ :

File 'nhomtho.inp'

5 5

6 1 3

5 5 1 2

9 4 1 5

9 4 5 2 3

6 2 5 1 4

Kết quả trên màn hình là :

Câu A : 1 4 (hoặc 1 5)

Câu B : 1 5

Chú ý : Nếu mỗi nhóm thợ không đặc tr- ng bởi số ng- ời , thay bằng giá trị công việc nhóm đó đạt đ- ợc . Đồng thời mỗi nhóm có thể gọi là 1 " ng- ời " thì

Bài toán trên có thể thay hình thức phát biểu : Cho M thợ , N công việc , giá công thuê thợ i là $B[i]$. Nếu $A[i,j]=1$ thể hiện thợ i làm đ- ợc công việc j . Hãy thuê thợ để hoàn thành tất cả N công việc trong 2 tr- ờng hợp

Câu a : Thuê sao tốn ít tiền nhất ,

Câu b : Thuê sao ít thợ nhất .

File dữ liệu vào cho nh- cũ

Bài toán 8 : (M nhóm thợ , hoàn thành N công việc)

```

Uses Crt;
Const Max = 50;
      Fi = 'nhomtho1.INP';
Type Ta = Array[1..max,1..max] of Byte;
      Tb = Array[1..max] of Byte;
Var N,M,LN,LT,Sn,St : Byte;
    A : Ta;
    B,KqA,KqB,Kq,phu : Tb;
    Thcv : Set of Byte;
Procedure TaoF;
  Var f : Text;
      k,p,i,j : Byte;
      TH : Set of Byte;
  Begin
    Assign(f,fi);
    Rewrite(f);
    Write('So cong viec n = ');Readln(n);
    Write('So nhom tho m = ');Readln(m);
    Writeln(f,n,' ',m);
    Randomize;
    For i:=1 to m do
      Begin
        Write(f,Random(10)+1,' ');
        TH := [];
        For j:=1 to n do
          Begin
            k := Random(n)+1;
            If Not (k in TH) then
              Begin
                TH := TH+[k];
                Write(f,k,' ');
              End;
          End;
        Writeln(f);
      End;
    Close(f);
  End;
Procedure Nhap;
  Var f : Text;
      i,j : Byte;
  Begin
    Assign(f,Fi); {$i-} Reset(f); {$i+}
    If (ioresult<>0) then
      Begin
        Write('Error file data ',fi,'.Enter to quit');
        Readln; halt;
      End;
    Readln(f,n,m);
    For i:=1 to m do
      Begin
        Read(f,B[i]);
        While not SeekEoln(f) do

```



```

Begin
  Read(f,j);
  A[i,j] := 1;
End;
Readln(f);
End;
Close(f);
End;

```

```

Function Dk_Can:Boolean;{= False : Có công việc không thể thuê nhóm nào làm đ- ợc}
Var i,j : Byte;
Function Cot_0(j:Byte):Boolean;{ True: c/v j không nhóm nào làm đ- ợc (cột j là cột 0)}
  Var i : Byte;
  Begin
    Cot_0 := False;
    For i:=1 to m do
      If a[i,j]<>0 then Exit;
    Cot_0 := True;
  End;
Begin
  Dk_Can := False;
  For j:=1 to n do
    If Cot_0(j) then Exit;
  Dk_Can := True;
End;

```

```

Procedure Toiuu;
Begin
  If (sn<Ln) then
    Begin
      Ln:=sn;
      KqA:=Kq;
    End;
  If (st<Lt) then
    Begin
      Lt:=st;
      KqB:=Kq;
    End;
End;

```

```

Procedure Them_nhom(i:Byte);
Var j : Byte;
Begin
  For j:=1 to n do
    If a[i,j]=1 then
      Begin
        Inc(Phu[j]); { So tho lam cong viec j }
        Thcv:=thcv+[j];
      End;
  Inc(sn);
  Inc(st,b[i]);
End;

```

```

Procedure Loai_nhom(i:Byte);
Var j : Byte;

```

```

Begin
  For j:=1 to n do
    If (A[i,j]=1) then
      Begin
        Dec(Phu[j]); {Phu[j] : số tho biết cv j của các nhóm đã thuê }
        { Thcv : tập hợp các công việc thuê }
        If (Phu[j]=0) then Thcv:=Thcv-[j];
      End;
    Dec(sn);
    Dec(st,b[i]);
  End;

```

```

Function Chapnhan(i:Byte):Boolean; { True : Nhóm i có khả năng làm cv chưa có ai làm }
Var j : Byte;
Begin
  Chapnhan := True;
  For j:=1 to n do
    If (A[i,j]=1) and Not (j in Thcv) then Exit;
  Chapnhan := False;
End;

```

```

Procedure Vet(i:Byte);
Begin
  If (Thcv=[1..n]) then
    Begin
      Toiuu;
      Exit;
    End;
  If ((Sn>=Ln) and (St>=Lt)) or (i=m+1) then Exit;
  If Chapnhan(i) then
    { Nhóm i làm được công việc mà nhóm tho đã tuyển không thể làm được }
    Begin
      Them_nhom(i);
      Kq[i]:=1;
      Vet(i+1);
      Loai_nhom(i);
      Kq[i]:=0;
    End;
  Vet(i+1);
End;

```

```

Procedure Khoitri;
Var i : Byte;
Begin
  Ln:=Max+1;
  Lt:=Max+1;
  St:=0;
  sn:=0;
  Thcv:=[];
  For i:=1 to n do Phu[i]:=0;
End;

```

```

Procedure Hienkq;
Var i : Byte;
Begin
  Writeln('Đang chạy chương trình ... ');
  Write('Phương án thuê ít nhóm nhất là : ');

```

```

For i:=1 to n do
  If KqA[i]=1 then Write(i:4);

  Write(#10#13,'Phuong an thue it tho nhat la : ');
  For i:=1 to n do
    If KqB[i]=1 then Write(i:4);
  Writeln(#10#13,'Chuong trinh da chay xong ! ');
End;
Procedure Xuly;
Begin
  If Not Dk_Can then
  Begin
    Writeln('Khong ton tai phuong an thue .Enter de thoat');
    Readln;
    Halt;
  End;
  Khoitri;
  Vet(1);
End;
BEGIN
  Clrscr;
  {TaoF;}
  Nhap;
  Xuly;
  Hienkq;
  Readln;
END.

```

Bài 9 : (Bài thi Tin học quốc gia 1995) Kết quả thi đấu quốc gia của n vận động viên (đánh số từ 1 đến N) trên m môn (đánh số từ 1 đến m) đ- ọc đánh giá bằng điểm (giá trị nguyên không âm) . Với mỗi vận động viên ta biết điểm đánh giá trên từng môn của vận động viên ấy . Các điểm này đ- ọc ghi trên một File văn bản có cấu trúc :

- + Dòng đầu ghi số vận động viên và số môn
- + Các dòng tiếp theo , mỗi dòng ghi các điểm đánh giá trên tất cả m môn của một vận động viên theo thứ tự môn thi 1,2,...,m . Các dòng này đ- ọc ghi theo thứ tự vận động viên 1,2,...,n
- + Các số ghi trên một dòng cách nhau ít nhất 1 dấu cách

Cần chọn ra k vận động viên và k môn để thành lập đội tuyển thi đấu Olympic quốc tế , trong đó mỗi vận động viên chỉ đ- ọc thi đấu đúng 1 môn ($1 \leq k \leq M, N$) , sao cho tổng số điểm của các vận động viên trên các môn đã chọn là lớn nhất .

Yêu cầu :

Đọc bảng điểm từ 1 File văn bản (Tên file cho từ bàn phím) ,sau đó cứ mỗi lần nhận một giá trị k nguyên d- ong từ bàn phím, ch- ong trình đ- a lên màn hình kết quả tuyển chọn d- ối dạng k cặp (i,j) với ý nghĩa vận động viên i đ- ọc chọn thi đấu môn j và tổng số điểm t- ong ứng với cách chọn . Ch- ong trình kết thúc khi nhận đ- ọc giá trị k=0 Các giá trị giới hạn : $1 \leq M, N \leq 20$, điểm đánh giá từ 0 đến 100

Thí dụ : File dữ liệu

```

3      3
1      5      0
5      7      4
3      6      3

```

mỗi khi nạp một giá trị k ta nhận đ- ọc :

k=1 , máy trả lời

(2,2)

Tổng số điểm = 7

k=2 , máy trả lời

(2,1) (3,2)

Tổng số điểm = 11

k=3 , máy trả lời

(1,2) (2,1) (3,3)

Tổng số điểm = 13

K=0 Kết thúc

{ \$A+,B-,D+,E+,F-,I+,L+,N-,O-,R-,S+,V- }

{ \$M 16384,0,655360 }

Program BL3;

Uses Crt;

Const Max = 20;

Type Ta = Array[1..max,1..max] of Integer;

Tb = Array[1..max] of Byte;

Tl = Array[1..max] of Integer;

Var N,M,k : Byte;

a : Ta;

b,lb : Tb;

G,Lg : Integer;

Ok : Set of Byte;

Procedure Input;

Var Tf : String;

f : Text;

Ok : Boolean;

i,j: Byte;

Begin

Repeat

Write(#10#13,'Cho biet ten file du lieu : ');

Readln(tf);

{ \$i- } Assign(f,tf); Reset(f); { \$i+ }

Ok:=Ioresult=0;

If Not Ok then

Begin

Writeln('File loi hoac khong co file ten la :',tf);

End;

Until Ok and (tf<>'');

Readln(f,n,m);

For i:=1 to n do

Begin

For j:=1 to m do Read(f,a[i,j]);

Readln(f);

End;

Close(f);

End;

Procedure NhapK;

Begin

Repeat

Write(#10#13,'Cho biet so mon can chon K:=');

{ \$i- } Readln(k); { \$i+ }

Until (Ioresult=0) and (k>=0) and (k<=m) and (k<=n);

End;

Procedure Hien;

Var i,j : Byte;

Begin

For i:=1 to n do

```

Begin
  For j:=1 to m do Write(a[i,j]:4);
  Writeln;
End;
End;

```

```

Procedure HienNghiem;
Var i : Byte;
Begin
  For i:=1 to n do
    If (Lb[i]>0) then Write('(',i,',',Lb[i],')');
  Writeln(#10#13,'Tong so diem = ',lg);
End;

```

```

Procedure VETCAN(i,somon:Byte);
Var j : Byte;
Begin
  If (somon>k) then
    Begin
      If (lg<g) then
        Begin
          Lb:=b;
          Lg:=g;
        End;
      Exit;
    End;
  If (i>n) then Exit;
  For j:=1 to m do
    If Not (j in ok) then
      Begin
        g:=g+a[i,j];
        b[i]:=j;
        Ok:=Ok+[j];
        Vetcan(i+1,somon+1);
        g:=g-a[i,j];
        b[i]:=0;
        Ok:=Ok-[j];
      End;
    Vetcan(i+1,somon);
  End;
End;

```

```

Procedure Vet;
Var i : Byte;
Begin
  For i:=1 to m do B[i]:=0;
  Lg:=-maxint div 2;
  G:=0;
  Ok:=[];
  Vetcan(1,1);
  Hiennghiem;
End;

```

```

BEGIN
  Clrscr;
  Repeat
    Input;
    Hien;
  Repeat
    NhapK;
    If (k>0) Then VET;
  Until (k=0);
  Write(#10#13,'ESC de thoat hoac phim bat ki de thu ');
  Write('lai voi file khac');
  Until (readkey=#27);
END.

```

Bài 9 : Cho M vận động viên , N môn thể thao . Vận động viên i đấu môn j đ- ọc số điểm là Di j . Cần chọn K vận động viên thi đấu k môn (mỗi vận động viên chỉ thi đúng 1 môn) . Nêu rõ cần chọn K vận động viên nào và những vận động viên ấy mỗi ng- ời thi đấu môn nào ?

```

Uses Crt;
Const Max = 100;
      Fi = 'Tongk.txt';
      Fo = "";
Type Pt = Record d,c,gt : Byte; End;
      M1 = Array[1..Max*Max+1] of Pt;
      M2 = Array[1..Max] of Record d,c :Byte;End;
Var B,LB : M1;
     M,N,k : Byte;
     Dx,Kq,Lkq : M2;
     Tong,LTong,csMax : LongInt;

```

```

Procedure DocF;
  Var i,j : Byte;
      F : Text;
  Begin
    Assign(F,Fi);
    {$I-} Reset(F); {$I+}
    If IoResult<>0 then
      Begin
        Writeln('Loi File ');
        Readln;
        Halt;
      End;
    Readln(F,M,N,k);
    For i:=1 to M do
      Begin
        For j:=1 to N do
          Begin
            Read(F,B[(i-1)*N+j].gt);
            B[(i-1)*N+j].d := i;
            B[(i-1)*N+j].c := j;
          End;
        Readln(F);
      End;
    Close(F);
    LB := B;
    CsMax := M*N;

```

```

End;
Procedure Sapxep_dl; { Sap giam dan }
  Procedure Quick(dau, cuoi : LongInt);
    Var    i, j, L : LongInt;
           phu   : Pt;
    Begin
      i := dau;
      j := cuoi;
      L := (i+j) div 2;
      Repeat
        While B[i].gt>B[L].gt do Inc(i);
        While B[j].gt<B[L].gt do Dec(j);
        If i<=j then
          Begin
            phu := B[i];
            B[i] := B[j];
            B[j] := phu;
            Inc(i);
            Dec(j);
          End;
        Until i>j;
        If dau<j then Quick(dau, j);
        If i<cuoi then Quick(i, cuoi);
      End;
    End;

    Begin
      Quick(1, M*N);
    End;
  Procedure Khoitri;
    Begin
      FillChar(B, Sizeof(B), 0);
      FillChar(Dx, Sizeof(Dx), False);
      FillChar(Kq, Sizeof(Kq), 0);
      Tong := 0;
      Ltong := 0;
    End;
  Procedure GhiToiuu;
    Begin
      Lkq := kq;
      Ltong := Tong;
    End;
  Procedure Chon(i, j : Byte); { xet toi o thu i trong Kq, tu o j trong B }
    Var    d1, c1 : Byte;
           delta, L, p, cL, Luu : LongInt;
    Begin
      cL := k-i; { cl : con lai }
      Delta := Tong-Ltong;
      If cL<0 then
        Begin
          If Delta>=0 then GhiToiuu;
        End
      Else
        Begin
          L := j-1;
          Repeat

```

```

        Inc(L);
        d1 := B[L].d;
        c1 := B[L].c;
        Until (L> Csmax) or ((Dx[d1].d=0) and (Dx[c1].c=0));
        If L<= csMax then
        If B[L].gt+B[L+1].gt*cL+Delta>0 then
        For p := L to csMax-1 do
        Begin
            d1 := B[p].d;
            c1 := B[p].c;
            If (B[p].gt+B[p+1].gt*cL+Delta>0) and
                (Dx[d1].d=0) and (Dx[c1].c=0) then
            Begin
                Dx[d1].d := 1;
                Dx[c1].c := 1;
                Luu := Tong;
                Tong := Tong+B[p].gt;
                Kq[i].d := d1;
                Kq[i].c := c1;

                Chon(i+1,p+1);

                Dx[d1].d := 0;
                Dx[c1].c := 0;
                Tong := Luu;
                Kq[i].d := 0;
                Kq[i].c := 0;
            End;
        End;
    End;
End;
End;
Procedure Inkq;
    Var i      : Byte;
        F      : Text;
    Begin
        Assign(F,Fo);
        ReWrite(F);
        Writeln(F,'k= ',k,' Tong = ',LTong);
        For i:=1 to k do
            Writeln(F,Lkq[i].d:2,' ',Lkq[i].c:2,' = ',
                LB[(Lkq[i].d-1)*N+Lkq[i].c].gt);
        Close(F);
    End;
BEGIN
    Clrscr;
    Khoitri;
    DocF;
    Sapxep_dl;
    Chon(1,1);
    Inkq;
END.

```