Chương III : DUYỆT VÀ TÌM KIẾM

I. Tìm kiếm :

A. Lý Thuyết :

1. Tìm kiếm theo chiều sâu :

```
Procedure          DFS ( i : integer ) ;
var
     j    :    integer ;
     Begin
          thăm đỉnh ( i ) ;
          xet [i]:=false ;
          for j := 1 to n do if xet[j] then if
a[i,j]=1 then DFS(j);
     End ;
```

2. Tìm kiếm theo chiều rộng :

```
Procedure          BFS (i);
Var
      dau , cuoi  , j :  integer ;
      Begin
            q [1 ]:=i ;
            dau :=1 ;
            cuoi := 1 ;
            While dau<= cuoi do
            Begin
                  i:= q [ dau ] ;
                  inc ( dau ) ;
                  Thăm đỉnh ( i ) ;
                  For j := 1 to n do
                  If xet [j] and ( a[i,j]=1) then
                  begin
                        inc ( cuoi ) ; q [cuoi ] := j ;
                        xet [j]:=false;
                  end ;
            End ;
      End ;
```

Chú ý :

Thuật toán tìm kiếm theo chiều rộng này sẽ đi từ s đến t bằng con đường qua ít nhất các đỉnh . Chính vì thế , nếu trong đồ thị có các cung chỉ là 0 và 1 thì đường đi trong thuật toán này là đường đi ngắn nhất . Chính vì thế nó có ứng dụng rất quan trọng trong các bài toán sau này .

Thuật toán này chạy dữ liệu trên mảng, còn tìm kiếm theo chiều sâu là trên stack. Chính vì thế tìm kiếm chiều sâu dễ gây ra tràn stack. Khi chúng ta dùng nó thì phải thêm dòng báo tăng bộ nhớ stack {$M 65000,6500 }

3. Tìm kiếm nhị phân trên một mảng đã sắp xếp :

```
Function binary ( x : integer ; a : array[1..n] of integer ):word ;
var
   low,high,mid      :      integer ;
   begin
        low := 1 ; high := n ;
        while low<=high do
        begin
             mid := (high+low)/2 ;
             if a[mid]<x then low:=mid+1
             else if x<a[mid] then high:=mid-1
             else high:=-1;
        end ;
        if high=-1 then tim:=mid else tim:=0 ;
   end ;
```

B. Bài toán :

Bài toán 1 : Đường đi đến số 0

Đề bài :

Mỗi một số nguyên dương đều có thể biểu diễn dưới dạng tích của 2 số nguyên dương X,Y sao cho X<=Y . Nếu như trong phân tích này ta thay X bởi X-1 còn Y bởi Y+1 thì sau khi tính tích của chúng ta thu được hoặc là một số nguyên dương mới hoặc là số 0 .

Ví Dụ : Số 12 có 3 cách phân tích 1*12 ,3*4 , 2*6 . Cách phân tích thứ nhất cho ta tích mới là 0 : (1-1)*(12+1) = 0 , cách phân tích thứ hai cho ta tích mới 10 : (3-1)*(4+1) = 10 , còn cách phân tích thứ ba cho ta 7 : (2-1)*(6+1)=7 . Nếu như kết quả là khác không ta lại lặp lại thủ tục này đối với số thu được. Rõ ràng áp dụng liên tiếp thủ tục trên, cuối cùng ta sẽ đến được số 0, không phụ thuộc vào việc ta chọn cách phân tích nào để tiếp tục

Yêu cầu : Cho trước số nguyên dương N (1<=N<=10000), hãy đưa ra tất cả các số nguyên dương khác nhau có thể gặp trong việc áp dụng thủ tục đã mô tả đối với N .

Dữ liệu : Vào từ file Zeropath.Inp chứa số nguyên dương N .

Kết quả : Ghi ra file văn bản Zeropath.Out :

Dòng đầu tiên ghi K là số lượng số tìm được

Dòng tiếp theo chứa K số tìm đọc theo thứ tự tăng dần bắt đầu từ số 0 .

Lu ý : Có thể có số xuất hiện trên nhiều đường biến đổi khác nhau, nhưng nó chỉ được tính một lần trong kết quả .
Ví Dụ :

ZEROPATH.INP	ZEROPATH.OUT
12	6 0 3 4 6 7 10

Hướng Dẫn :

Đơn giản là sau mỗi lần phân tích thì chắc chắn kết quả mới luôn nhỏ hơn số đó. Vì vậy ta chỉ cần lưu trữ dưới mảng A: [0..10000] of boolean ; trong đó A[i]=true nếu nó xuất hiện trên đường đi đó , ngược lại thì A[i]=false. Bằng cách loang theo chiều sâu, chúng ta sẽ đánh dấu các số nếu nó được dùng đến , cho đến khi không thể nào loang được nữa thì dừng .

Bài toán 2 : Nhà gương cười

Đề Bài :

Ban quản lí nhà gương cười muốn thay đổi lại toàn bộ các tấm gương phủ tường của nhà gương để phục vụ tốt hơn khách tham quan. Bạn cần giúp ban quản lí tính diện tích gương cần mua để thực hiện công việc đó .

Nhà gương được mô tả bởi bảng ký tự kích thước NxN (3<=N<=33. Bạn thấy đấy con số ‘3’ quả là thần bí). Một số ô của bảng chứa dấu chấm (‘.’) để kí hiệu ô trống . Một số ô khác chứa dấu thăng (‘#’) để ký hiệu ô vuông được bao bọc bởi các bức tường .
Tất cả các ô vuông đều có kích thước 3*3 mét .

Hình 1 : Mô tả nhà gương cười .

Ngời ta xây dựng các bức tường bao bọc chung quanh nhà gương, ngoại trừ ô ở góc trên trái và ô ở góc dưới phải tương ứng với lối vào ra của nhà gương. Các bức tường cũng được xây dựng chung quanh các ô đánh dấu thăng. Ngoài ra không có bức tờng nào khác. Giả thiết rằng ô ở góc trên trái và dưới phải của bảng luôn chứa dấu chấm .
Yêu Cầu : Bạn cần tính diện tích của các bức tường ở phía trong của nhà gương, tức là phần nhìn thấy được bởi du khách vào chơi trong nhà gương và cũng chính là diện tích gương cần mua để đổi mới nhà gương. Lưu ý là không có kẽ hở giữa các bức tường để du khách có thể đi hoặc nhìn sang ô có bức tường liền kề. Xem hình vẽ minh hoạ cho phần bức tường nhìn thấy được từ bên trong nhà gong. Chiều cao của mỗi bức tường đều là 3 mét .

Dữ liệu : Vào từ file Mirror.inp :

Dòng đầu tiên chứa số N

Dòng thứ i trong số N dòng tiếp theo chứa N ký tự mô tả các ô ở dòng thứ i của bảng ký tự mô tả nhà gương. Các ký tự trên một dòng chỉ là dấu chấm hoặc dấu thăng được ghi liền nhau .

Kết quả : Ghi ra file Mirror.Out : diện tích của các bức tường mà bạn tìm được .

Ví Dụ :

MIRROR.INP	MIRROR.OUT
5 ....## ..#.. ..###	198

Hướng dẫn :

Chúng ta sẽ loang theo chiều sâu, tìm phần gương có trong nhà. Nếu đến bức tường nào đã được đánh dấu thì không xét đến. Thực chất, một ô như vậy có đến bốn hướng có thể được ghép gương. Nhưng khi ghép nó rồi thì các ô bên cạnh nó có chung cạnh sẽ được đánh dấu luôn. Cứ như vậy cuối cùng ta lấy số mặt trong (nhà) nhân với 9 là ra kết quả.

Bài toán 3 : Con ngựa

Đề bài :

Một bàn cờ hình chữ nhật kích thước MxN , M,N nguyên dương không lớn hơn 100. Bàn cờ chia thành các ô vuông đơn vị bằng các đường song song với các cạnh. Các dòng ô vuông đánh số từ 1 đến M từ trên xuống dưới, các cột đánh số từ 1 đến N từ trái sang phải . Cho trước một số nguyên dương K<=1000. Một con ngựa đứng ở ô [u,v] và nhảy không quá k bước .

Yêu cầu : Hãy cho biết con ngựa có thể nhảy đến bao nhiêu ô khác ô[u,v] trên bàn cờ và đó là những ô nào (khi đứng tại một ô , con ngựa có thể nhảy tới ô đối đỉnh của hình chữ nhật kích thước 2x3).

Dữ liệu : Vào từ file MA.INP trong đó :

Dòng đầu tiên ghi hai số M,N

Dòng thứ hai ghi số K

Dòng thứ ba ghi hai số U ,V

Kết quả : Ghi ra file MA.OUT :

Dòng đầu tiên ghi S là số ô con ngựa có thể nhảy đến

Tiếp theo là S dòng, mỗi dòng ghi chỉ số dòng và chỉ số cột của một ô mà con ngựa có thể nhảy đến .

Ví Dụ :

MA.INP	MA.OUT
5 5 1 2 3 4	6 1 1 1 5 3 1 3 5 4 2 4

Hướng dẫn :

Chúng ta sẽ loang theo chiều sâu , tìm kiếm xem những ô nào con mã có thể đặt chân đến trong vòng K bước nhảy .

Bài toán 4 : Tiền thưởng

Đề bài :

Chúng ta đã biết truyền thuyết về giải thưởng 264-1 đồng tiền vàng mà ông vua phải trao cho người sáng chế ra trò chơi cờ vua. Sau đây là một truyền thuyết khác về giải thưởng này. ông vua cho phép nhà sáng chế lựa chọn giải thưởng theo quy tắc sau :

" Hãy đặt con ngựa lên một ô nào đó của bàn cờ. Trên ô mà nhà ngươi đặt ta sẽ đặt voà đó 2K đồng tiền vàng. Tiếp theo trên những ô mà con ngựa có thể đến được sau một nước đi ta sẽ đặt vào đó 2K-1 đồng tiền vàng. Nói chung, trên những ô mà con ngựa có thể đến sau ít nhất p<=k nước ta sẽ đặt vào đó 2K-P đồng tiền vàng. Nếu như nhà ngươi tỏ ra quá tham lam, không đủ sức đem hết được số lượng đồng tiền vàng của giải thởng thì ta sẽ chém đầu ngươi "

Yêu Cầu : Bạn hãy giúp nhà sáng chế xác định vị trí đặt ngựa để có thể nhận được giải thưởng lớn nhất mà vẫn giữ đợc cái đầu trên cổ của mình.

Dữ liệu : Vào từ file Prize.Inp chứa hai số nguyên K (0<=K<=25) và M (0<=M<=109)

Kết quả : Ghi ra file văn bản Prize.out :

Dòng đầu tiên chứa N là số lượng đồng tiền vàng theo giải thưởng mà bạn tìm được (N= 0 nếu như không có cách lĩnh thưởng) .Nếu N > 0 thì ghi tiếp

Dòng thứ hai ghi vị trí các ô của bàn cờ mà tại đó có thể đặt ngựa để đạt giải thưởng trên.

Biết rằng bàn cờ vua được đánh số các hàng theo thứ tự chữ cái : a,b,c,d,e,f,g,h . Và các cột được đánh số theo thứ tự số : 1,2,3,4,5,6,7,8 .

Ví Dụ :

PRIZE.INP	PRIZE.OUT
1 5	5 a2 a7 b1 b8 g1 g8 h2 h7
2 21	17 a1 a8 h1 h8
3 92	91 b3 b6 c2 c7 f2 f7 g3
g6	
2 4	0

Hớng Dẫn :

Chúng ta sẽ loang theo chiều sâu, đánh dấu các ô mà con mã nhảy tới. Và đánh dấu số bước mà con mã đến đó. Với mỗi ô đặt mã có thể có thì ta tìm xem tổng số tiền có thể được. Ta thấy rằng nếu tồn tại cách để thì sẽ có bốn ô giống nhau, và chúng đối xứng nhau qua các đường chéo. Chính vì thế ta chỉ cần loang 1 / 4 các ô bàn cờ là được .

Bài toán 5 : Các con hậu chung sống hoà bình

Đề bài :

Trên bàn cờ quốc tế kích thước N x N (N 50) người ta đặt N con hậu. Ta nói các con hậu ở trạng thái chung sống hoà bình nếu như không tìm được hai con mà con này có thể tấn công con kia. Bạn cần tìm số lợng trạng thái chung sống hoà bình, có thể thu được từ một trạng thái cho trước bằng cách xếp lại vị trí của ba trong số N con hậu đã xếp.

Dữ liệu: Vào từ File văn bản QUEEN. INP:

+ Dòng đầu tiên chứa số N

+ N dòng tiếp theo cho biết vị trí N con hậu được xếp trên bàn cờ ở trạng thái chung sống hoà bình. Mỗi dòng chứa 2 số nguyên X, Y ghi cách nhau bởi dấu cách. Các số này cho biết toạ độ theo chiều đứng và theo chiều ngang và nằm trong khoảng từ 1 đến N.

Kết quả: Ghi ra File QUEEN.OUT số lượng trạng thái tìm được.

Chú ý: Các con hậu không được đánh số. Vì vậy nếu bạn chỉ đổi chỗ vòng quanh ba con hậu bất kỳ thì không dẫn đến trạng thái mới.

Ví dụ:

QUEEN.INP	QUEEN.OUT
4	0
2 1	
1 3	
3 4	
4 2	

Hướng dẫn :

Ta sẽ tìm lần ba con hậu trong N con hậu đã cho; Rồi tìm cách đổi chỗ chúng. Với mỗi lần đổi chỗ chúng mà khác với cách nguyên thuỷ thì sẽ tăng số cách lên 1. Cứ như thế mà chương trình chỉ cần chạy với tổ hợp chập 3 của N : = N&*(N-1)*(N-2) .

Bài toán 6 : ĐƯỜNG ĐI TRÊN LƯỚI Ô VUÔNG

Đề bài :

Cho một lưới ô vuông kích thước N x N. Các dòng của lưới được đánh số từ 1 đến N từ trên xuống dưới, các cột của lưới được đánh số từ 1 đến N từ trái qua phải. Ô nằm trên giao của dòng i, cột j sẽ được gọi là ô (i, j) của lưới. Trên mỗi ô (i, j) của lưới người ta ghi một số nguyên dương ai, i, j = 1,2, ..., N. Từ một ô bất kỳ của lưới được phép di chuyển sang ô có chung cạnh với nó. Thời gian để di chuyển từ một ô này sang một ô khác là 1 phút. Cho trước thời gian thực hiện di chuyển là K (phút), hãy xác định cách di chuyển bắt đầu từ ô (1, 1) sao cho tổng các số trên các ô di chuyển qua là lớn nhất (Mỗi ô của lưới có thể di chuyển qua bao nhiêu lần cũng được).

Dữ liệu: Vào từ file văn bản NETSUM.INP:

Dòng đầu tiên chứa các số nguyên dương N, K (2 N 100), 1 K 10000).

Dòng thứ i trong số N dòng tiếp theo chứa các số nguyên ai1, ai2... , aiN, 0 < ai 10000.

(Các số trên cùng một dòng được ghi cách nhau bởi ít nhất một dấu cách).

Kết quả: Ghi ra file văn bản NETSUM.OUT:

Dòng đầu tiên ghi tổng số các số trên đường di chuyển tìm được.

K dòng tiếp theo mỗi dòng ghi toạ độ của một ô trên đường di chuyển (bắt đầu t ô (1, 1)).

Ví dụ:

NETSUM.INP	NETSUM.OUT
5 7	2 1
1 1 1 1 1	1 1
1 1 3 1 9	1 2
1 1 6 1 1	1 3
1 1 3 1 1	2 3
1 1 1 1 1	2 4
	2 3
	2 4

Hướng dẫn :

Loang các ô có thể đến của các đường đi trên lưới. Tìm cách đi nào có đường đi mà tổng lớn nhất thì sẽ lấy .

Bài toán 7 : Di chuyển ghế trong vườn

Đề bài :

Trong một khu vườn có dạng một hình vuông độ dài cạnh n được chia ra thành lưới n x n ô vuông có một số cây và một cái ghế băng. Mỗi cây chiếm một ô vuông, còn ghế băng chiếm ba ô vuông liên tiếp. Người

ta muốn di chuyển ghế băng từ vị trí ban đầu đến một vị trí mới. Hình 1 dưới đây mô tả khu vườn: Các số 0 mô tả ô rỗng, các số 1 cho biết ô có cây. Vị trí đặt ghế được đánh dấu bởi các chữ "b", còn vị trí kết thúc được đánh dấu bởi các chữ "e" (Xem hình 2). Giả thiết là luôn có cách di chuyển từ vị trí đầu đến vị trí cuối.

00011	bbb11
00000	00000
00000	00000
11000	11000
00000	eee00
Hình 1	Hình 2

Ta có thể di chuyển ghế sang phải, sang trái, lên trên, xuống dưới, hoặc quay một góc 90^o nếu như không có cây nằm trên đường di chuyển nó. Các hình 3 - 8 mô tả các phép di chuyển vừa nêu. Lưu ý là phép quay 90^o quanh tâm của ghế chỉ có thể thực hiện khi các ô lân cận là rỗng. Trong hình 9 các ô đánh dấu "x" phải là rỗng mới có thể quay ghế hình 10 là kết quả của phép quay.

00000	00000	00000	00000	00000	00000	00000	00000
00000	0bbb0	00000	00000	00000	00b00	0xxx0	00b00
0bbb0	00000	00000	00bbb	bbb00	00b00	0bbb0	00b00
00000	00000	0bbb0	00000	00000	00b00	0xxx0	00b00
00000	00000	00000	00000	00000	00000	00000	00000
Hình 3 Ban đầu		Hình 4 Trên		Hình 5 Dưới		Hình 6 Phải	Hình 7 Trái H

Cần tìm cách di chuyển với số bước di chuyển là nhỏ nhất.

Dữ liệu: Vào từ File văn bản TMOVE.INP:

+ Dòng đầu tiên chứa số n là kích thước của vườn (3 n 40)

+ n dòng tiếp theo mỗi dòng chứa n ký tự mô tả khu vườn. Trong đó có 3 chữ "b" mô tả vị trí của ghế và 3 chữ "e" mô tả vị trí cần chuyển ghế đến.

Kết quả: Ghi ra File TMOVE.OUT:

+ Dòng đầu tiên ghi số bước di chuyển ít nhất cần thực hiện K

+ Dòng thứ 2 ghi K ký tự mô tả các bóc di chuyển cần thực hiện. Sử dụng các ký tự L, R, D, U, T theo thứ tự tương ứng để mô tả các phép di chuyển trái, phải, dưới, trên, quay.

Ví dụ:

TMOVE.INP	TMOVE.OUT	TMOVE.INP	
	TMOVE.OUT		
5	8	6	7
bbb11	DDRRDDLL	bbblel	DTRDRRU
00000		0000e0	

00000	0000e0
11000	110001
eee00	111001
	001101

Hướng dẫn :

Chúng ta quy định vị trí của một chiếc ghế là gồm : vị trí ô trung tâm (ô chính giữa của ghế) mà nó đứng; và hướng của ghế : Hướng lên (dọc) hoặc hướng ngang (ngang) . Mỗi lần xoay ghế, hay chuyển ghế thì sẽ đưa vị trí của ghế đến vị trí mới. Chúng ta sẽ loang các trạng thái theo chiều rộng để tìm đường đi ngắn nhất giữa vị trí cũ và vị trí mới .

Bài toán 8 : Chess

Đề bài :

Một bàn cờ có dạng bảng chữ nhật kích thước mxn, trên đó có một số quân cờ. Một quân cờ chỉ có thể di chuyển sang một ô kề cạnh còn trống, mỗi di chuyển như vậy được gọi là một bước di chuyển. Cho hai trạng thái của bàn cờ, hãy chỉ ra một dãy các bước di chuyển để đưa bảng từ trạng thái ban đầu đến trạng thái đích với số phép di chuyển là ít nhất. Mỗi trạng thái được mô tả là một ma trận mxn trong đó số ở hàng i cột j là 1 nếu tại vị trí i,j tương ứng có quân cờ đang đứng hoặc bằng 0 nếu không có.

Dữ liệu : Vào từ file : Chess.inp ;

Dòng đầu ghi hai số nguyên dương m , n (1 < m , n <= 10)

Tiếp theo là 2m dòng thể hiện ma trận mô tả trạng thái xuất phát và trạng thái đích. m dòng đầu tiên thể hiện ma trận xuất phát, m dòng tiếp theo là ma trận đích.

Input cho đảm bảo luôn có nghiệm.

Kết quả : Ghi Ra file Chess.Out

Dòng đầu ghi K là số ít nhất các phép biến đổi tìm được.

K dòng tiếp theo, mỗi dòng mô tả một phép biến đổi, theo đúng thứ tự biến đổi, gồm 4 số nguyên dương u, v , x, y thể hiện di chuyển quân cờ ở vị trí (u,v) sang vị trí (x,y).

Hướng dẫn :

ở bài toán thứ 36 ở chương II , các bạn thấy được một thuật toán, mà ứng dụng của nó nhờ vào bài toán ghép cặp. Tuy nhiên chương trình bài toán sẽ rất dài. Chúng ta sẽ giải quyết nhờ một phương pháp hết sức đơn giản : Tìm kiếm theo chiều rộng. Ta sẽ coi một trạng thái của bảng là một đỉnh của đồ thị mới. Mỗi lần di chuyển một quân cờ trên bàn thì nó sẽ tạo ra một trạng thái mới của bảng, tức là sẽ đến một đỉnh mới. Trong

bài toán này chúng ta sẽ chỉ xét với (m=n=4). Tức là ở file input, không có dòng đầu tiên .

Mỗi trạng thái của bảng là một loạt các ô có giá trị 0 và 1. Chúng ta sẽ trải nó ra thành một hàng thì sẽ tạo ra một bảng một chiều chỉ toàn các số 1 và 0 . Vì có 16 ô , nên mỗi bảng như vậy sẽ tương ứng với hệ nhị phân của một số nào đó nằm trong word (16 bit) . Tức là số đỉnh của đồ thị có thể có sẽ là 216 .

a1	a2	a3	a4
a5	a6	a7	a8
a9	a10	a11	a12
a13	a14	a15	a16

Bảng 1

Bảng mới sau khi trải như sau :

a1	a2	a3	a4	a5	a6	a7	a8	a9	a10	a11	a12	a13	a14	a15	a16

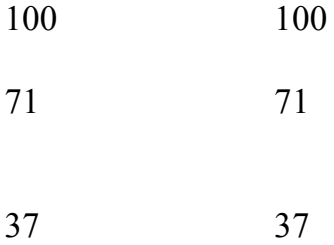
Bảng 2

Khi di chuyển các quân thì các quân (ở bảng 1) có thể đi tới 4 ô bên cạnh nếu có thể (trừ trường hợp đi ra ngoài bảng). Tức là tương ứng ở bảng 2, giả sử tại vị trí Ai thì nó có thể có đi đến những ô : Ai-1 , Ai+1 , Ai-4, Ai+4 (Phải trừ những trường hợp nó đi ra ngoài bảng). Quá trình di chuyển quân 1 như vậy tức là bít thứ i sẽ được tắt, còn các bít được đến sẽ được bật. Loang theo chiều rộng của quá trình chuyển bit cho đến khi chuyển đến được trạng thái mong muốn. Đường di chuyển đó chính là cách di chuyển cờ với số bước ít nhất để đến trạng thái mong muốn .

Bài toán 9 : Rót nước

Đề bài :

Có ba bình thuỷ tinh hình trụ tròn có kích thước giống nhau và có dung tích 100 đơn vị (Cm3). Trên hai bình có khắc một số vạch giống nhau ghi dung tích phần từ đáy bình đến thiết diện ngang ứng với vạch này và một bình không có vạch (xem hình vẽ).



13 13

Bình 1 Bình 2 Bình 3

Ban đầu bình 1 chứa 100 đơn vị nước còn hai bình kia rỗng. Hãy xét xem liệu có thể đổ nước qua lại giữa các bình để có thể có được một đơn vị nước ở bình 3 không? Sau mỗi lần đổ nước từ bình này sang bình khác, ít nhất một trong hai bình phải hoặc là rỗng hoặc mức nước còn lại phải trùng với một vạch của nó, lượng nước bám vào thành bình xem nh không đáng kể. Nếu có thể, hãy chọn cách với số lần đổ nước từ bình này sang bình khác là ít nhất.

Dữ liệu: Vào được cho bởi file INP.TXT trong đó dòng thứ nhất ghi số nguyên dương N, 1<=N<=20, là số vạch trên các bình 1 và 2. Dòng thứ hai ghi N số nguyên dương tăng dần là các mức thể tích ứng với N vạch, mức cao nhất luôn là 100.

Kết quả: Ghi ra file OUT.TXT. Dòng thứ nhất ghi chữ CO hay KHONG tuỳ theo kết quả xem xét. Nếu là có, ghi trong dòng thứ hai số lần đổ nước giữa hai bình, từ dòng thứ ba, mỗi dòng ghi một lần đổ nước thể hiện bởi 5 số: ví dụ nếu đổ nước từ bình 1 sang bình 3 kết quả cho bình 1 chứa 71 đơn vị, bình 2 - 0 đơn vị, bình 3 - 29 đơn vị thì ghi 5 số:

1 3 71 0 29

Thứ tự các dòng từ trên xuống dưới là thứ tự của các lần đổ nước.

Ví dụ :

INP.TXT	OUT.TXT
4	CO
13 37 71 100	8

Hướng dẫn :

Chúng ta sẽ coi ba cọc là ba đỉnh. Thì mỗi đỉnh có thể có 100 mực nước. Nhưng tổng mức nước của ba bình luôn là 100. Vậy thì nếu biết mực nước ở hai bình đầu là i, j thì bình còn lại là 100-i-j . Vậy Chúng ta chỉ cần lưu giá trị của hai bình đầu là được, còn lại thì nó chỉ xác định duy nhất một bình còn lại. Giả sử Rot[i,j] là số hiện trạng của ba bình có bình 1, bình hai có mức nước là: i , j và bình 3 là 100-i-j . từ hiện trạng bây giờ , ta loang theo chiều rộng các cách đổ nước qua lại giữa các bình, thì nó sẽ xác định thêm một trạng thái mới. Giả sử chúng ta rót từ bình 1 sang bình 2 với điều kiện bình một về mực k nào đó (k<i) thì hiện trạng mới là: Rot[k,j+i-k]. Tương tự cho các hiện tượng có thể, các bạn có thể tìm được số lần rót ít nhất để có thể có hiện trạng Rot[99-k1,k1] thì lúc đó bình 3 sẽ có 1 lít nước (k1 là các giá trị có thể có từ 0 đên 99).

Chúng ta có thể giả tương tự cho bài toán *ba con kì nhông* ở phần chương V (mà ở bài toán đó chúng ta chỉ kiểm tra điều kiện có nghiệm

chứ chưa tìm cách chuyển màu), từ bài toán trên ta có thể giải như bài toán vừa ra.

Bài toán 10 : bắn bi

Đề bài :

Cho một bảng vuông gồm NxN ô vuông đơn vị, các dòng ô vuông được đánh số từ 1 đến N từ trên xuống dưới, các cột ô vuông được đánh số từ 1 đến N từ trái sang phải. Tại một số ô vuông có đặt vật cản. Mỗi vật cản ở một trong hai trạng thái 1/2. Từ phía trái của bảng khi bắn một viên bi theo một dòng nào đó, nếu trên đường đi viên bi gặp vật cản 1/2 thì viên bi sẽ rẽ sang trái/phải và vật cản sẽ chuyển sang trạng thái 2/1.

Bài toán đặt ra như sau: có một đích đặt phía bên phải của bảng ngang với dòng D của bảng, hãy tìm cách lần lượt bắn một số ít nhất viên bi từ bên trái bảng theo các dòng thích hợp sao cho lượt bắn cuối cùng trúng đích, khi bắn một viên bi, chỉ sau khi viên bi đó ra khỏi bảng ta mới bắn tiếp viên bi khác.

Dữ liệu : Vào được cho bởi file BI.INP trong đó dòng thứ nhất ghi số nguyên dương N<=30, trong N dòng tiếp theo, dòng thứ I ghi N số B[I,1], . . ., B[I,N] mà B[I,J]=0 nếu ô [I,J] trống, B[I,J]=1/2 nếu tại ô [I,J] có vật cản ở trạng thái 1/2. Dòng cuối cùng ghi số D. Biết rằng số vật cản không quá 15.

Kết quả: Ghi ra file BI.OUT như sau: dòng thứ nhất ghi số 0/1 tuỳ theo có thể hay không có thể bắn trúng đích, nếu dòng thứ nhất ghi số 1, dòng thứ hai ghi số S là số lần bắn bi, trong S dòng tiếp theo, mỗi dòng ghi một số hiệu dòng mà theo trình tự đó ta lần lượt bắn bi vào dòng tương ứng.

Ví dụ

BI.INP	BI.OUT
3 0 2 0 0 1 2 0 0 1 1	1 1 2

Hướng dẫn :

Dễ thấy rằng, tất cả bảng chỉ có tối đa 15 vật cản. Mà các vật cản này chỉ có thể chuyển từ 1 sang trạng thái 2. Chúng ta sẽ dùng một bảng 1 chiều lưu dữ trạng thái của bảng bằng cách lưu giữ trạng thái của các vật cản: A[i]=0 hoặc 1 nếu như vật cản tính từ trên xuống , trái sang phải có vị trí thứ i là trạng thái 1 hoặc 2. Hoàn toàn tương tự như bài toán thứ 2, mỗi lần bắn một viên bi thì trạng thái bit của bảng sẽ thay đổi . Tức là số trạng thái có thể có của bảng là 215. Nó tương ứng với các số từ 0 đến

215, Ta sẽ dùng mảng đánh dấu : DD:Array[0..215] Loang theo chiều rộng sẽ ra kết quả cần tìm .

Bài toán 11 : Kết quả cuộc thi

Đề bài :

Trong đợt tổng kết phiếu dự thi đoăn kết quả của giải vô địch bóng đá. Ban tổ chức cuộc thi cần chọn ra k (k<=2000) người có điểm số cao nhất để trao giải thưởng. Danh sách tên và điểm của n (n<=106) người dự thi được ghi trên một file văn bản giả thiết rằng không có hai ngời nào có cùng số điểm.

Yêu cầu : Đưa ra danh sách K người có điểm cao nhất của cuộc thi .

Dữ liệu : Vào từ file văn bản Loto.Inp :

Dòng đầu tiên ghi 2 số nguyên dương N,K

Tiếp đến là N nhóm dòng, mỗi nhóm dòng gồm 2 dòng, trong đó dòng đầu ghi họ tên (là dãy gồm không quá 10 ký tự) của người dự thi, dòng thứ hai ghi điểm số mà anh ta đạt được .

Kết Quả : Ghi ra file văn bản Loto.Out K nhóm dòng tương ứng với k người có điểm cao nhất của cuộc thi theo thứ tự điểm số giảm dần .

Dòng đầu tiên trong nhóm ghi họ tên

Dòng thứ hai ghi điểm số

Ví dụ :

loto.inp	loto.out
6 3 A 9 B 7 C 12 D 8 E 5 F 11	C 12 F 11 A 9

Hướng dẫn :

Thuật toán : Lấy K số lớn nhất, vì số số hạng rất lớn, nên chúng ta vừa đọc file vừa xử lí .

Bước 1 : Chúng ta nhận K số đầu tiên của fike vào bảng Top

Bước 2 : Sắp xếp bảng Top này tăng dần

Bước 3 : Với mỗi lần đọc thêm một số thì chúng dùng phương pháp tìm kiếm nhị phân để nhận số đó không, nếu nhận thì sẽ loại số cuối cùng của bảng Top

Cứ như bước 3 cho đến khi đọc hết file thì bảng Top là bảng cần dùng

Sử dụng phép Move để chuyển bảng với tốc độ nhanh .

Bài toán 12 : Xén Cỏ

Đề bài :

Một mảnh vườn hình tam giác đều đỉnh hướng lên trên, cạnh đáy nằm ngang. Độ dài cạnh vườn bằng N là một số nguyên dương không lớn

hơn 30. Vườn được chia thành các ô tam giác đều cạnh 1 đơn vị bằng các đường thẳng song song cách đều với các cạnh vườn. Mỗi ô hoặc trồng cỏ hoặc trồng hoa, ta gọi tắt tương ứng là ô cỏ và ô hoa. Mỗi ô được định vị bởi hai số I, J mà I là số thứ tự dòng chứa ô tính từ trên xuống dưới, J là số thứ tự của ô trên dòng I tính từ trái sang phải, ta gọi I, J là *toạ độ* của ô.

Để xén cỏ, người ta dùng một rô bốt đáy hình tam giác đều có cạnh 2 đơn vị. Ban đầu rô bốt luôn ở 4 ô trên cùng của vườn và các ô này không trồng hoa. Rô bốt có thể di chuyển tịnh tiến theo 6 hướng với các ký hiệu sau:

A: Sang trái; B: Sang phải; C: Theo hướng Tây-Nam lập với đường nằm ngang góc 60^0 , D: Theo hướng Đông-Bắc lập với đường nằm ngang góc 60o; E: Theo hướng Đông-Nam lập với đường nằm ngang góc 60o; F: Theo hướng Tây-Bắc lập với đường nằm ngang góc 60o.

Mỗi bước di chuyển, rô bốt tịnh tiến một đoạn 1 đơn vị. Khi rô bốt di chuyển qua ô nào thì cỏ/hoa trong ô đó đều bị xén.

Rô bốt không được di chuyển qua các ô trồng hoa (Điều kiện (*)).

Yêu Cầu :

1. Hãy tìm cho rô bốt một hành trình tuân theo điều kiện (*) sao cho số ô cỏ được xén là nhiều nhất.

2. Cho trước một ô cỏ (U,V), liệu rô bốt có thể di chuyển đến cắt cỏ tại ô đó mà hành trình của rô bốt từ vị trí ban đầu đến ô (U,V) tuân theo điều kiện (*) không? Nếu có, hãy tìm một hành trình với số bước di chuyển ít nhất.

Dữ liệu: Vào được cho bởi file văn bản VUON.INP trong đó dòng thứ nhất ghi hai số N, M, M là số ô hoa. Trong M dòng tiếp theo, mỗi dòng ghi hai số là toạ độ của một ô hoa. Các ô hoa được đánh số thứ tự từ 1 đến M theo trình tự dòng từ trên xuống dưới, dòng cuối cùng ghi hai số U và V.

Kết quả: Ghi ra file văn bản VUON.OUT như sau: dòng thứ nhất ghi số C là số ô cỏ mà rô bốt có thể cắt được nhiều nhất (kể cả bốn ô tại vị trí ban đầu của rô bốt; dòng thứ hai ghi từ đầu dòng một xâu ký tự chỉ gồm các ký tự A..F thể hiện hành trình của rô bốt mà thứ tự các ký tự theo trình tự di chuyển trong hành trình của rô bốt, xâu này có thể rỗng, dòng thứ ba ghi số 1/0 tuỳ theo kết quả câu 2 là có/không, nếu là số 1, dòng thứ ba ghi số B là số bước di chuyển của rô bốt để đi đến ô (U,V), nếu B>0, dòng thứ t ghi từ đầu dòng một xâu chỉ gồm các ký tự A..F thể hiện hành trình của rô bốt mà thứ tự các ký tự theo trình tự di chuyển trong hành trình.

Ví dụ

VUON.INP	VUON.OUT

6	3	26
4	1	EAEEACA
4	6	1
6	7	1
3	1	C

Hướng dẫn :

Chúng ta lưu giữ vị trí của con robot như bài toán một. Sau đó tìm cách đi trên vườn mà sao cho không đụng đến hoa, để thoả mãn bài toán. Tức là loang theo chiều sâu , hoặc chiều rộng. Đối với ý 2 thì nên loang theo chiều rộng để tìm ra con đường ngắn nhất đó .

Bài toán 13 : Match

Đề bài :

Một nhà máy sản xuất động cơ xe máy sản xuất được N pittông và N xilanh. Các pittông được đánh số từ 1 đến N , các xilanh cũng được đánh số từ 1 đến N. Mỗi pittông và mỗi xilanh đều được gán với một chỉ số chất lượng là một số nguyên. Gọi chỉ số chất lợng của pittông i là ai , i=1,2,..N, còn chỉ số chất lợng của xilanh j là bj , j=1,2..N. Nếu lắp ráp pittông i với xilanh j thì ta được một bộ pittông-xilanh hoàn chỉnh có đánh giá chất lượng là ai x bj.

Yêu cầu :

Hãy giúp nhà máy lắp rắp K bộ pittông-xilanh hoàn chỉnh với giá chất lượng là lớn nhất .

Dữ liệu : Vào từ file văn bản Match.Inp :

Dòng đầu tiên ghi hai số nguyên dương N và K (N<=107 , K<=N và K<=103)

Dòng thứ i trong số n dòng tiếp theo chứa hai số nguyên ai , bj được ghi cách nhau bởi dấu cách , ai<=32767 , bj<=32767 , i =1,2,..N

Kết quả : Ghi ra file văn bản Match.Out :

Dòng đầu tiên chứa tổng đánh giá chất lượng của K bộ pittông-xilanh hoàn chỉnh tìm được

Mỗi dòng trong số K dòng tiếp theo chứa hai chỉ số p , q trong đó p là chỉ số của pittông còn q là chỉ s của xilanh của một bộ pittông-xilanh hoàn chỉnh trong số K bộ pittông-xilanh cần lắp rắp

Ví dụ :

MATCH.INP	MATCH.OUT
6 2 1 5 3 3 4 6 1 9 -2 4 2 -3	54 3 4 2 3

Hướng dẫn :

Chúng ta cần tìm K cặp sao cho chúng có tích lớn nhất. Nhưng chúng ta phải tính đến trường hợp một số âm nhân với một số âm thì cho kết quả là một số dương .

Thuật toán :

Bước 1 : Sắp xếp dãy Ai tăng dần , gọi là dãy A1i

Bước 2 : Sắp xếp dãy Ai giảm dần , gọi là dãy A2i

Bước 3 : Sắp xếp dãy Bi tăng dần , gọi là dãy B1i

Bước 4 : Sắp xếp dãy Bi giảm dần, gọi là dãy B2i

Bước 5: Với mỗi cặp trên cùng của tích Ai1[top1]*Bi1[top2] với Ai2[last1]*Bi2[last2] thì cặp nào lớn hơn thì lấy nó ra tạo thành một cặp, và loại các số này ra khỏi các bảng chứa chúng .

Cứ tiếp tục thực hiện bước 5 cho đến khi lấy được K cặp .

Bài toán 14 : Bro

Đề bài :

Có n quán bia (n <= 10000) nằm trên một đường tròn, đánh số 1,2 ..n theo chiều kim đồng hồ. Thông tin về một quán bia bao gồm A(i) là lượng bia cần cung cấp theo lít, B(i) là khoảng cách đến quán bia tiếp theo theo chiều kim đồng hồ. Chú ý khi đó B(n) là khoảng cách từ quán n đến quán 1. Hàng ngày, nhà máy bia phải cung cấp bia cho các quán, giá chuyên chở 1 lít trên 1 đơn vị khoảng cách. Vì tất cả đều nằm trên đường tròn nên có hai con đường từ nhà máy đến quán, do đó chi phí tính theo đường ngắn hơn trong hai đường.

Cần đặt nhà máy bia tại một vị trí trên đường tròn sao cho tổng chi phí vận chuyển bia là nhỏ nhất .

Dữ liệu : Vào từ file BRO.INP như sau :

Dòng đầu tiên là số N

N dòng, dòng thứ i ghi 2 số A(i), B(i) theo như đã mô tả.

Các A(i), B(i) nguyên dương <= 32000

Kết Quả : Ghi ra file BRO.OUT như sau :

Một dòng ghi tổng chi phí tối thiểu.

Hướng dẫn :

Dễ thấy luôn có nghiệm tối ưu là nhà máy phải đặt tại một quán nào đó. Do đó ta thử đặt nhà máy tại các quán xem tại vị trí nào ít nhất, nếu chỉ đơn thuần theo cách đó độ phức tạp sẽ là n2. Ở đây ta sẽ lưu một cách khéo léo để lưu lại các kết quả . Cụ thể

Khi thử mỗi vị trí i ta có vị trí j đối xứng quán j qua tâm đường tròn. Khi đó vị trí j sẽ là điểm phân chia các quán thành 2 lớp: Mỗi lớp có đường đi ngắn nhất tới nhà máy theo đường riêng. Tuy nhiên, khi duyệt các quán theo thứ tự trên đường tròn thì vị trí j dịch chuyển đúng 1 vòng,

do đó số quán bị chuyển từ lớp này sang lớp khác là tuyến tính với n . Do đó chi phí tính toán tuyến tính theo n,

Bài toán 15 : magic

Đề bài :

Kế tục thành công của trò chơi với khối lập phương thần bí, Ngài Rubik sáng tạo ra dạng phẳng của trò chơi này gọi là trò chơi các ô vuông thần bí. Đây là một bảng gồm 8 ô vuông bằng nhau (xem hình 1).

Trong bài này chúng ta xét bảng trong đó mỗi ô vuông có một mầu khác nhau. Các mầu được ký hiệu bởi 8 số nguyên dương đầu tiên (xem hình 1). Trạng thái của bảng được cho bởi dãy ký hiệu mầu của các ô được viết lần lượt theo chiều kim đồng hồ bắt đầu từ ô ở góc trái trên và kết thúc tại ô ở góc trái dưới. Ví dụ, trạng thái của bảng trong hình 1 được cho bởi dãy (1,2,3,4,5,6,7,8). Trạng thái này được gọi là trạng thái khởi đầu.

Có thể dùng 3 phép biến đổi cơ bản đối với bảng có tên là 'A', 'B' và 'C':

'A': đổi chỗ dòng trên và dòng dưới,
'B': thực hiện một phép hoán vị vòng quanh sang phải,
'C': quay theo chiều kim đồng hồ bốn ô giữa.

Biết rằng từ trạng thái khởi đầu luôn có thể chuyển về một trạng thái bất kỳ bằng cách dùng các phép biến đổi cơ bản nói trên.

Tác động của ba phép biến đổi cơ bản được mô tả trong hình 2, trong đó các số viết bên cạnh bảng dùng để chỉ vị trí các ô vuông của bảng và ô vuông ở vị trí p chứa số i có nghĩa là, sau khi áp dụng phép biến đổi tương ứng, ô vuông mà vị trí trước khi biến đổi của nó là i được chuyển đến vị trí p.

Bạn phải viết chương trình tìm dãy các phép biến đổi cơ bản để chuyển bảng từ trạng thái khởi đầu cho trong hình 1 về một trạng thái đích cho trước (Câu A). Bạn sẽ được thêm hai điểm nếu số lượng phép biến đổi tìm được không vượt quá 300 (Câu B).

Dữ liệu : Vào từ File INPUT.TXT chứa 8 số nguyên dương trong dòng đầu tiên mô tả trạng thái đích.

Kết liệu: Trong dòng đầu tiên của file OUTPUT.TXT chương trình của bạn cần ghi L là số lượng phép biến đổi của dãy các phép biến đổi tìm được. Trong L dòng tiếp theo phải ghi dãy tên các phép biến đổi cơ bản theo trình tự thực hiện, mỗi tên ghi ở vị trí đầu tiên của mỗi dòng.

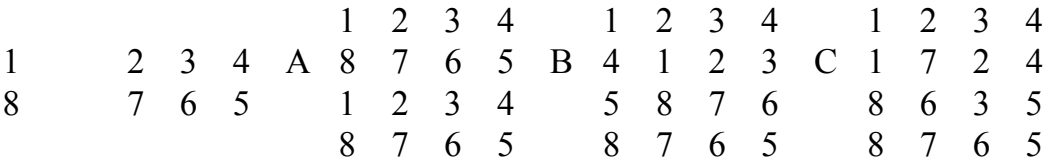
Công cụ trợ giúp

Chương trình MTOOL.EXE ghi trong thư mục bài giúp bạn luyện tập với trò chơi các ô vuông thần bí. Bằng cách thực hiện lệnh "mtool input.txt output.txt" bạn có thể xem tác động của dãy các phép biến đổi cho trong output.txt đối với bảng ở trạng thái đích cho trong input.txt.

Ví dụ về dữ liệu vào và ra

INPUT.TXT
2 6 8 4 5 7 3 1

OUTPUT.TXT
7
B
C
A
B
C
C
B



Hình 1 Hình 2

Hướng dẫn :

Chúng ta trải tấm đó thành một mảng một chiều. Tức là sẽ có 8 phần tử, và một trạng thái của trò chơi là một tập hợp số là một hoán vị nào đó của tập hợp (1,2,3,..8). Tức là mỗi trạng thái được đặc trưng bởi một dãy số đó. Để thuận lợi trong việc lưu giá trị, thì giá trị của nó tức chính là giá trị vị trí của nó trong hoán vị. Tức là có thể có 8! trạng thái của bảng. Mỗi trạng thái tương ứng với số x nằm trong khoảng 1,..8!. Để giải bài toán này thì chúng ta phải giải bài toán tìm vị trí của một hoán vị nào đó, và bài toán : khi biết vị trí, hãy tìm hoán vị đó. (có thể xem ở chong V) . Sau đó thực hiện loang theo chiều rộng tương tự như các bài toán trên để tìm ra cách xoay ô ít nhất để đưa bảng số về trạng thái cần .

Bài toán 16 : TÌM KIẾM MẪU

Đề bài :

Cho một hình chữ nhật kích thước m x n (m dòng và n cột) được chia thành các ô vuông như mô tả trong hình 1a dưới đây. Mỗi ô vuông được đánh số liên tiếp sao cho ô ở góc trên bên phải được đánh số 1, ô bên phải nó được đánh số 2 và ô ở góc dưới phải được đánh số m x n.

1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 16
17 18 19 20

Hình 1a

Hình 1b cho thấy lưới ô vuông với một số ô được sơn đen. Xét một hình chữ nhật con với ô góc trên trái chính là ô góc trên trái của hình chữ nhật đã cho, còn ô ở góc dưới phải có toạ độ (p, q). Chẳng hạn hình chữ nhật con xác định bởi ô dưới phải (2, 1) của hình chữ nhật cho trong hình 1b có dạng cho trong hình 1c. Hình chữ nhật con này xuất hiện ở 4 vị trí khác nhau trong hình chữ nhật đã cho trong hình 1b (xem các hình dưới đây).

(Hình)

Bạn cần viết chương trình nhập thông tin về hình chữ nhật đã cho và một hình chữ nhật con của nó, sau đó đưa ra số lần xuất hiện của hình chữ nhật con trong hình chữ nhật đã cho. Bạn không cần tính các phép quay của hình chữ nhật con. Lưu ý rằng hình chữ nhật con luôn được chọn ở góc trên trái của hình chữ nhật đã cho.

Dữ liệu: Vào từ file văn bản RPART.INP:

Dòng đầu tiên chứa bốn số nguyên m (1 m, n 20), p (1 p m) và q (1 q n).

Dòng thứ hai cho biết số lượng ô được bôi đen x, (0 x m, n). Mỗi một trong số x dòng tiếp theo chứa vị trí của một ô được sơn đen (cách đánh số vị trí đọc mô tả trong hình 1a).

Kết quả: Ghi ra file RPART.OUT số lần xuất hiện của hình chữ nhật con trong hình chữ nhật đã cho (tính cả vị trí ban đầu ở góc trên trái của nó).

Ví dụ:

RPART.INP	RPART.OUT	RPART.INP	RPART.OUT
5 4 2 1	4	4 5 2 2	2
7		2	
3		2	
5		5	
10 11 14 18 19			

Hướng dẫn :

Tìm kiếm từng hình chữ nhật con có thể có của hình ban đầu , rồi tìm các hình có thể có . (Tìm kiếm thông thường) .

Bài toán 17 : NHÀ CAO TẦNG

Đề bài :

Một quần thể nhà cao tầng được xây dựng trên một nền hình chữ nhật, trên đó được chia thành M x N ô vuông (M dòng, N cột). Các dòng được đánh số từ 1 đến M và các cột được đánh số từ 1 đến N. Người ta xem khu nhà được tạo bởi các khối có đáy là một ô vuông với những chiều cao nào đó mà người ta gọi là những đơn nguyên. Một đơn nguyên được định nghĩa là một tập hợp các đơn nguyên có đáy tạo thành một miền gồm những ô vuông kề cạnh. Thí dụ, hình vẽ dưới đây mô tả một quần thể gồm 3 khối nhà:

hình

Người ta đánh số các khối nhà bằng những số nguyên liên tục bắt đầu từ 1 theo trình tự duyệt các ô đáy theo từng dòng từ 1 đến M, và trên mỗi dòng, duyệt các ô đáy theo từng cốt từ 1 đến N. Thí dụ, các khối nhà cho trong hình vẽ trên được đánh số theo thứ tự các ô đáy như sau (có mầu xám)

1 1 2
2
3
4 3

Người ta muốn quét vôi các bức tường xung quanh tất cả các khối nhà. Hãy xác định:

+ Số lợng các khối nhà

+ Tổng số diện tích quét vôi

+ Khối nhà có diện tích quét vôi lớn nhất và giá trị của diện tích này.

Dữ liệu vào cho trên file văn bản có dạng:

M N

N [1, 1] H[1, 2] . . . H [1, N]

H[2, 1] H[2, 2] . . . H[2, N]

. . . . . .

H[M, 1] H[M, 2] . . . H[M, N]

trong đó H[i, j] là chiều cao của đơn nguyên [i, j] với quy ước bằng 0 khi đơn nguyên này không có. Giả thiết rằng các giá trị này đều là những số nguyên và tính theo đơn vị một cạnh ô vuông. Các số trên cùng một dòng ghi cách nhau ít nhất một dấu trắng.

Thí dụ, hình vẽ trên mô tả 3 khối nhà, tơng ứng với file dữ liệu:

```
4 6
1 2 3 0 2 1
1 0 1 0 0 1
2 1 1 0 0 1
0 0 0 1 1 0
```

Kết quả : Ghi ra file văn bản gồm:

+ Dòng đầu ghi số lượng các khối nhà

+ Dòng thứ hai ghi tổng số diện tích quét vôi

+ Dòng thứ ba ghi số nhà của khối có diện tích quét vôi lớn nhất và giá trị của diện tích này (ghi cách nhau ít nhất một dấu trắng).

Thí dụ, với file dữ liệu đã nêu, file kết quả là:

```
3
50
1 30
```

Ghi chú

Các giá trị diện tích được tính theo đơn vị diện tích một ô vuông

Giới hạn kích thước M, N 100, H[i, j] 2000

Hướng dẫn :

Chúng ta sẽ tính lượng vôi có thể có tại mỗi nhà cao tầng. Đơn giản bằng cách chúng ta tìm kiếm lần lượt theo tất các cạnh biên của các hình (biên trên , dưới , trái , phải) rồi đến mặt trên. Lúc đó thì ta sẽ loại trừ được trường hợp các nhà trùng nhau. Hoặc là tìm kiếm theo loang tại mỗi vị trí. Hoặc dùng hàm tính các giá trị có thể có tại mỗi nhà .

Bài toán 18 : Lưới tam giác

Đề bài :

Cho một lưới các ô vuông kích thước MxN , M,N nguyên dương không lớn hơn 50. Các dòng của lưới đánh số từ 1 đến M từ trên xuống dưới, các cột đánh số từ 1 đến N từ trái sang phải . Mỗi ô vuông được chia thành bốn tam giác bằng hai đường chéo, các tam giác đặt tên là P,Q,R,S như hình vẽ :

Mỗi tam giác con có mầu trắng hay đen . Để ghi nhận tình trạng mầu của các tam giác trong một ô vuông , ta quy ước như sau: tam giác P mầu trắng /đen sẽ được gán số 0/1 , tam giác Q mầu trắng /đen sẽ được gán số 0/2, tam giác R mầu trắng/đen sẽ được gán số 0/4, tam giác S mầu trắng/đen sẽ được gán số 0/8. Tình trạng mầu của mỗi ô vuông sẽ được thể hiện bởi một số trong phạm vi từ 0 đến 15 bằng tổng các số gán cho các tam giác thuộc ô đó. Như vậy lới có thể đợc cho bởi một mảng hai chiều kích thước MxN .

Ta gọi một tập các tam giác đen là một hình nếu nó thoả mãn các điều kiện :

Từ một điểm bất kỳ thuộc hình ta có thể đi đến một điểm bất kỳ khác thuộc hình theo một đường hoàn toàn nằm trong hình

Nếu thêm vào hình một tam giác đen khác thì điều kiện 1 bị vi phạm

Ta có thể ghi nhận một hình bằng cách dùng một mảng kích thước MxN mà phần tử [I,J] của mảng bằng tổng các số ứng với các tam giác của ô [I,J] thuộc hình

Yêu cầu : Hãy tìm mọi hình của lưới

Dữ liệu : được cho từ file Luoi.Inp trong đó :

Dòng thứ nhất ghi hai số M , N

Tiếp theo là M dòng , dòng thứ i ghi N số M[i,1],M[i,2]....M[i,n]

Yêu cầu : Hãy tìm mọi hình của lưới

Dữ liệu : Được cho từ file Luoi.Inp trong đó :

Dòng thứ nhất ghi hai số M , N

Tiếp theo là M dòng , dòng thứ i ghi N số M[i,1],M[i,2]....M[i,n] mà M[i,j] là số ghi nhận tình trạng mầu của các tam giác thuộc ô [I,J]

Kết quả : Ghi vào file Luoi.Out :

Dòng thứ nhất ghi số H

Tiếp theo là H nhóm dòng , mỗi nhóm gồm M+1 dòng mà dòng đầu ghi số D là số tam giác của hình , M dòng còn lại ghi các thông tin về một hình theo cách nói trên .

Ví Dụ :

LUOI.INP LUOI.OUT

4 4 1 4 10 0 4 0 12 0 4 0 0 0 1 0 7 0 5 1 1 0 0 0

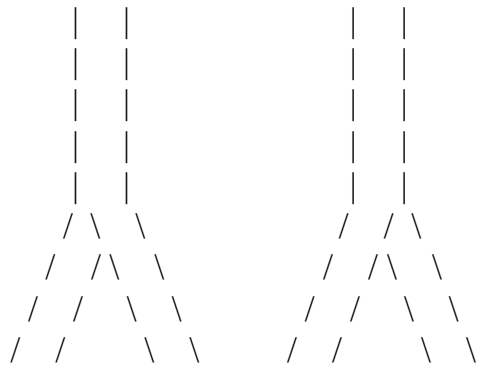
Hướng dẫn :

Loang các thành phần liên thông của các miền . Rồi đánh số thứ tự của các miền .

Bài toán 19 : ôtômát

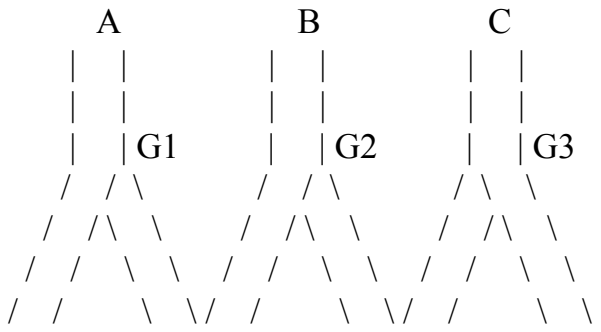
Đề bài :

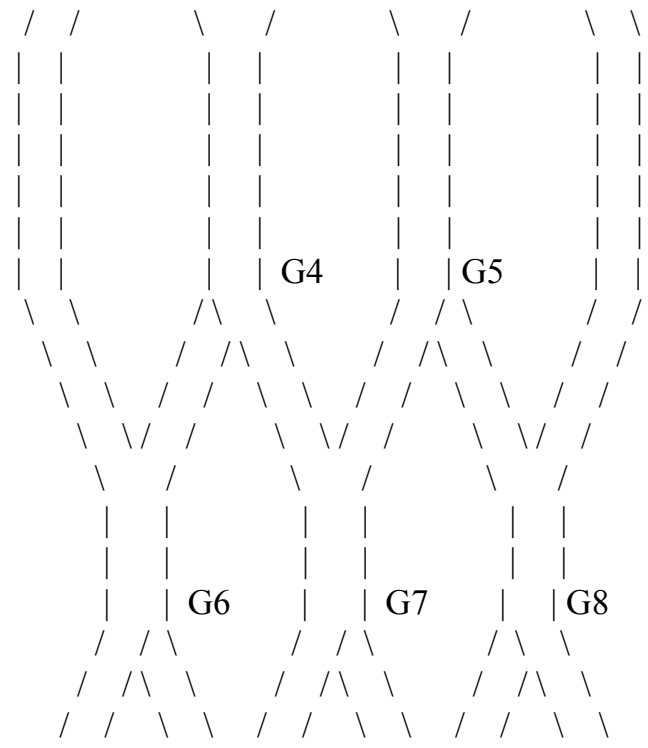
Có một ô tô mát được ghép từ các chi tiết có một trong hai trạng thái 0 hay 1 như hình 1. Ô tô mát có cấu trúc như hình 2 gồm 8 chi tiết G1, . ., G8 với ba lối vào A, B, C. Trạng thái của ô tô mát được thể hiện bởi một xâu nhị phân độ dài 8 là các trạng thái tương ứng của G1, . ., G8.



Hình 1 Trạng thái 1 Trạng thái 0

Ô tô mát hoạt động như sau: khi thả một quả cầu vào một lối vào nào đó, sau khi quả cầu đi qua một chi tiết nào đó, chi tiết đó thay đổi trạng thái từ 0 thành 1 hoặc từ 1 thành 0. Hoạt động của ô tô mát đợc thể hiện bởi một xâu ký tự S chỉ gồm các chữ cái hoa A, B, C mà mỗi ký tự trong xâu S thể hiện việc ta thả quả cầu vào lối vào với tên ký tự đó. Ví dụ S = AABC có nghĩa là ta lần lợt thả quả cầu vào các lối A, A, B, C.





Hình 2

Bài toán đặt ra như sau: Cho hai trạng thái bất kỳ T1 và T2 của ô tô mát. Hãy tìm một xâu ký tự S ngắn nhất có thể được thể hiện hoạt động của ô tô mát chuyển từ trạng thái T1 đến trạng thái T2.
Các xâu T1 và T2 nhập từ bàn phím và viết xâu S ra màn hình.

Hướng dẫn :

Với mỗi cách thả bi thì sẽ có một trạng thái mới của 8 ôtomat : G1,G2,..G8 . Mỗi trạng thái Gi có hai giá trị 1 ,0. Cho nên chúng hợp thành một xâu nhị phân của một số có 8 bit. Ta sẽ coi một trạng thái là một số nằm trong khoảng [0,28] trong đó mỗi số viết dưới dạng nhị phân là trạng thái của các otomat. Sau đó thực hiện các phép toán loang theo chiều rộng. Ta sẽ tìm được cách thả bi vào Otomat .

Bài toán này so với các bài toán : Chess , Bi , ..Thì nó hoàn toàn dễ hơn rất nhiều. Nên các bạn hoàn toán tự viết lấy được chơng trình của mình.

II. Duyệt và nhánh cận :

A. Lý Thuyết :

1. Thuật toán đệ quy nhánh cận :

Sau đây là thủ tục đệ quy cơ bản :

```
Procedure try ( i ) ;
begin
        Nếu i đã vượt hạn thì xuất kết quả
```

```
        ngược lại thì
        Nếu i chưa vượt hạn thì
        begin
                đánh dấu xâm nhập tới i ;
                try ( giá trị tiếp theo của i )
                loại i ra khỏi danh sách ;
        end ;
end ;
```

Tuy thuật toán đơn giản, thế nhưng khi chúng ta ứng dụng nó sẽ rất quan trọng trong các bài toán tìm kiếm quan trọng. Vì số vòng for (hoặc repeat) là có hạn khi chúng ta viết chơng trình. Còn nhờ vào thủ tục đệ quy thì ta hoàn toàn thay thế số vòng for một cách đơn giản.

Vì thủ tục trên sẽ đi hết tất cả các giá trị của tìm kiếm. Nên khi bài toán có dữ liệu lớn thì thuật toán dường như không thể chạy nổi. Thông thương có hai loại bài toán duyệt : đó là tính theo cấp độ chương trình phải chạy thường là N! hoặc tăng theo hàm mũ. Chính vì thế công việc hạn chế các bước duyệt là một điều vô cùng quan trọng. Một bài toán mà chúng ta phải duyệt thì các bước duyệt được chặn càng sớm thì nó càng đỡ vất vả trong chơng trình. Nên khâu phát hiện cách duyệt là một điều vô cùng quan trọng .

1	2	3
4	5	6
7	8	9
	0	

Chẳng hạn chúng ta nhìn phím điện thoại sau :

Khi chúng ta cho con mã ở vị trí 0 rồi sau đó nhảy đi theo quy tắc nhảy của nó. Thì sau khi nó đi đợc 6 bước nào đó thì sẽ được một số nào đó trong cuộc gọi. Bài toán đặt ra là có bao nhiêu số có thể có .

Tất hẳn các bạn sẽ cho các số có thể có. Rồi thử vào quá trình đi. Nhưng chúng ta thấy rằng ô số 5 sẽ không bao giờ được con mã đặt chân tới. Đấy là ô không phải duyệt. Tuy ví dụ không ảnh hưởng đến tốc độ và chương trình bài toán. Nhưng đây là ví dụ làm các bạn phải nhận định rõ ràng các phần mà ta không duyệt và duyệt là vô cùng quan trọng. Nó sẽ có ích khi bộ dữ liệu là rất lớn . Mà chương trình đệ quy (tìm kiếm) lại tăng theo cấp số nhân nên một phát minh nào cũng có thể tăng tốc độ chương trình rất nhiều .

Ngoài ra một công cụ không thể thiếu trong dạng này đó là các cận. Các bạn hãy định nghĩa rằng cận chỉ là một hàm nào đó mà làm cho giới hạn chơng trình của bài toán không chạy đến. Chúng ta sẽ dùng một hàm kiểm tra nào đó trong phần duyệt để xem giá trị tiếp theo đó có thoả mãn không. Sau đây là thủ tục đệ quy có cận :

```
procedure try (i ) ;
begin
    nếu i vượt hạn thì ghi kết quả ;
    nếu i chưa vợt hạn thì
    nếu cho i vào danh sách thì thoả mãn bài toán thì (* hàm cận *)
    begin
            đánh dấu sự xâm nhập của i ;
            try ( giá trị tiếp của i ) ;
            loại i ra khỏi danh sách ;
    end ;
```

Tuy có những bài toán hàm cận rất đơn giản, thế nhưng tốc độ chương trình bài toán thì cải thiện lên hàng trăm, hàng vạn lần. Các bạn sẽ đươc tiếp xúc với các bài toán đó sau này .

B. Bài toán :

Bài toán 20 : Xổ số điện toán

Đề bài :

Có N người (đánh số từ 1 đến N) tham gia một đợt xổ số điện toán. Mỗi người nhận đợc một thẻ gồm M ô (đánh số từ 1 đến M). Người chơi được chọn K ô trong số các ô đã cho bằng cách đánh dấu các ô được chọn. Sau đó các thẻ này được đưa vào máy tính để xử lý.

Máy tính chọn ra K ô ngẫu nhiên (gọi là các ô kết quả) và chấm điểm từng thẻ dựa vào kết quả đã sinh. Cứ mỗi ô chọn đúng với ô kết quả thì thẻ chơi được tính 1 điểm. Giả thiết biết các ô chọn cũng như các điểm tương ứng của từng thẻ chơi, hãy xác định tất cả các kết quả có thể có mà máy sinh ra.

Vào: Đọc dữ liệu từ file văn bản XOSO.INP, gồm:

- Dòng đầu ghi các số N, M, K
- Dòng thứ i trong N dòng tiếp ghi thẻ chơi của người i gồm K+1 số: K số đầu là các số hiệu của các ô chọn, số cuối là điểm tương ứng.

Ra: Ghi kết quả ra file văn bản XOSO.OUT, mỗi dòng là một kết quả gồm K số ghi số hiệu các ô mà máy đã sinh.

Ghi chú:

- các số trên cùng một dòng trong các file vào / ra, được ghi cách nhau ít nhất một dấu trắng.
- giới hạn kích thước: N 100, M 50, K 10.
- dữ liệu vào trong các test là hợp lệ và đảm bảo có ít nhất một đáp án.

Ví dụ:

XOSO.INP	XOSO.OUT

5 9 4	1 2 3 4
2 4 6 8 2	2 3 4 7
5 6 8 9 0	
2 4 5 6 2	
1 2 3 7 3	
3 5 6 9 1	

Hướng dẫn :

Chúng ta sẽ duyệt chỉnh hợp các xổ số có thể. Sau đó trong mỗi cái chúng ta sẽ kiểm tra có đúng với bảng đó không .

Cách tìm cận :

Chúng ta xây dựng bảng K[i] kiểm tra số kết quả đúng của người thứ i. Sau mỗi lần thêm một số nào đó vào xổ số, thì ta kiểm tra xem nếu việc thêm này làm tăng số kết quả đúng của người chơi nào đó lên thì sẽ loại. Tuy cận đơn giản, nhưng chương trình đã rút gọn đi rất nhiều (chạy rất nhanh)

Bài toán 21 : Phiếu đục lỗ

Đề bài :

Cho một bảng hình vuông NxN. Trên đó ta có N2 lỗ đợc cho sẵn. Nhưng lỗ đó có thể chưa chắc đã bị đục. Cho một tập K bảng đó. Biết rằng lúc đầu các bảng này được chồng lên nhau tạo thành một khối sao cho lỗ (i,j) của tầng trên cũng là lỗ (i,j) của tầng dưới. Một phép xoay bảng là xoay theo chiều kim đồng hồ một góc 90 , 180 , 270 .

Yêu cầu : Hãy tìm cách xoay ít nhất sao cho từng vị trí (i,j) của khối thì đều có ô bị đục .

Dữ liệu : Vào từ file BL4.Inp :

Dòng đầu tiên hai số M,N, (0<=N<=10 , 0<=M<=20)

M dòng tiếp, mỗi dòng ghi N*N số biểu diễn các ô của bảng thứ i có bị đục không (nếu thì ghi 1, ngược lại ghi 0)

Kết quả : Ghi ra file Bl4.Out :

Dòng đầu tiên là số phép xoay cần thực hiện

Dòng tiếp theo ghi M số biểu diễn phép quay của bảng thứ i (nếu ghi 0 thì không xoay , nếu ghi 1 thì xoay 90 , nếu 2 thì xoay180 , nếu ghi 3 thì xoay 270)

Ví dụ :

Hướng dẫn :

Chúng ta sẽ duyệt các bảng có thể xoay hoặc không xoay, nếu xoay thì chúng có ba trạng thái. Như vậy mức độ phức tạp là 410. Nhưng chúng ta sẽ có một phương pháp loại bớt những trạng thái không thể bằng cách :

Khi chúng ta xoay một khối (có bỗn hướng 0 : không xoay , 1 : xoay 90o , 2: xoay 180o và 3 xoay 270o). Nhưng khi đó mỗi lần thêm thì chúng ta tăng thêm số các số có thể bị đục, tức là giảm đi các ô không bị đục lỗ. Nếu số ô còn chưa được đục mà lớn hơn số ô có thể có (đục) thì chúng ra sẽ loại nó. Mặt khác chúng ta dùng cận trên để chặn số lần duyệt .

Bài toán 22 : Turn Around

Đề bài :

Để sắp xếp đống củi vừa thu được , cu tí muốn sắp xếp nó một cách hiệu quả. Bằng cách cậu xếp các đống củi (mẩu gỗ tròn) theo trật tự giảm dần từ đáy và tuân theo quy tắc vật lý .

Ví Dụ : cậu có 9 mẩu gỗ thì cu cậu sẽ sắp xếp nó thành một khối như sau :

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

Mỗi mẩu gỗ có một hướng trong 4 hướng (đông , tây , nam , bắc). Sau khi cu tí đã xếp xong thì cu tí không nghĩ với cách xếp của nó. Tuy nhiên, khi xoay một mẩu gỗ thì các mẩu gỗ kề nó sẽ quay theo. Ví dụ : nếu cu tí quay mẩu gỗ thứ 6 thì các mẫu gồ 2 ,3,7,9,8,5 sẽ quay theo chiều ngược lại .

Yêu cầu: Hãy tìm cách xoay mẫu gỗ để từ một trạng thái ban đầu, đưa về trạng thái mong muốn (là tất cả cùng quay về hướng bắc)

Dữ liệu : Vào từ file Turn.Inp :

Dòng đầu tiên ghi số N (N<50)

Dòng kế tiếp ghi N số thể hiện cho hớng của từng mẩu gỗ

Kết quả : Ghi ra file Turn.Out :

Dòng đầu tiên ghi số bước cần quay

Dòng kế tiếp, mỗi dòng là cặp số (x,t) thể hiện mẩu x quay đi 90*t

.

Ví dụ :

Hướng dẫn :

Cách 1 :

Thuật toán : Duyệt (hạn chế những phần không duyệt)

Đầu tiên chúng ta điền các vị trí vào mảng như sau :

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

Hoàn toàn tương tự cho các ví dụ khác về cách điền. Ta thấy một nhận xét rằng :

Khi chúng ta biết được số lần xoay của hàng dưới cùng. Thì chắc chắn hàng trên nó cũng biết được.Thật vậy Khi chúng ta biết được số lần biến đổi của ô 1, ô 2 thì ô 5 cũng biết: vì Số lần biến đổi ô 1 bằng số lần biến đổi của ô 5 và ô 2 và cả ô 1. Nhưng chúng ta xác định được ô 1, 2 thì 5 xác định được. Hoàn toàn tương tự : khi biết ô 1 , 5 , 2 , 3 thì ô 6 xác định được. Cứ như vậy ta biết hết hàng 2 tương tự cho hàng trên nữa. Cứ như vậy cho đến hết bảng .

Vì vậy chúng ta chỉ cần duyệt hàng dưới (về cách biến đổi) còn các hàng trên chỉ phụ thuộc theo mà thôi . Nhưng số hạng nhiều nhất ở hàng dưới là 10. (vì số khúc gỗ không quá 50). Tức là Chúng ta chỉ cần duyệt 410 là cùng .

Cách 2 :

Thuật toán : Giải phương trình Gaus .

thực chất thuật toán 1 là ứng dụng của nghiệm pt Gaus. Thật vậy số lần biến đổi ô 1 thì chỉ có các ô tác động đến nó: ô 2 và ô 5. Biết được giá trị ô 1 và ô 2 thì xác định được ô 5.

Bài toán 23 : Xoay ô

Đề bài :

Một diện tích được tạo bởi các ô vuông, gồm N hàng và N cột (các hàng được đánh số từ 1 đến N theo chiều từ trên xuống dưới và các cột được đánh số từ 1 đến N theo chiều từ trái sang phải). Mỗi ô vuông được chia đôi theo đường chéo: một nửa mầu đen, một nửa mầu trắng. Trạng thái của mỗi ô vuông được mã hoá từ 0 đến 3 theo hình vẽ sau:

0 1 2 3

Một diện tích được xác định bởi bảng trạng thái các ô vuông của nó. Có thể thay đổi các trạng thái này bằng cách xoay các ô theo chiều kim đồng hồ một trong những góc 90, 180, 270.

Diện tích được gọi là hợp lệ nếu các phần cùng mầu của các ô sát nhau không được có chung cạnh, chẳng hạn diện tích trong hình 1 là không hợp lệ, hình 2 là hợp lệ:

Hình 1 Hình 2

Hãy xác định xem diện tích đã cho đã hợp lệ chưa? nếu chưa, cần tìm cách xoay lại một số ô cần thiết để diện tích là hợp lệ, sao cho số ô cần xoay là ít nhất.

Dữ liệu: Vào cho trong file văn bản ROT.INP, dòng đầu là số N, tiếp theo là bảng N dòng, N cột ghi trạng thái của các ô tơng ứng. Các giá trị trên cùng một dòng ghi cách nhau ít nhất một dấu trắng.

Lời giải đưa ra file văn bản ROT.OUT, trong đó nếu diện tích đã cho đã là hợp lệ thì chỉ cần ghi một số 0, trái lại văn bản gồm dòng đầu là số ô cần xoay, các dòng tiếp, mỗi dòng mô tả một ô cần xoay bao gồm: toạ độ hàng, toạ độ cột và góc cần xoay theo chiều kim đồng hồ của ô (tính theo đơn vị độ - là một trong các giá trị 90, 180, 270). Các số này ghi cách nhau ít nhất một dấu trắng.

Thí dụ, ROT.INP (hình 1):

ROT.OUT

4

6

2 0 1 0

1 1 270

2 0 3 2

2 1 270

2 0 2 3

2 3 180

1 2 1 0

2 4 180

3 2 270

4 2 180

diện tích nhận được cho bởi hình 2.

Giới hạn kích thước N 10.

Hướng dẫn :

Thuật toán Tìm Kiếm (Duyệt)

Chúng ta thấy rằng khi một hình đã xác định được hai hình có chung cạnh (hai hình này phải chung đỉnh) thì chúng ta chỉ xác định được duy nhất một hình tại vị trí đó. Gọi mảng A[i,j] có gía trị : Nếu một ô nào đó có hai ô chung kề cạnh (hai ô đó chung một đỉnh) có giá trị lần lượt là i, j thì giá trị tại ô đó là A[i,j] .

Thật vậy chúng ta thấy hình trên , khi chúng ta xác địn được hai ô A1 và A3 thì chỉ tồn tại duy nhất một giá trị (và hoàn toàn xác định) của ô A2 . Hoàn toàn tương tự cho ô A4 . Chính vì thế chúng ta có hai trờng hợp :

Nếu một ô khi đã biết được hai ô có chung cạnh với nó: ở phía trên và bên phải

Nếu một ô khi đã biết được hai ô có chung cạnh với nó : ở phía dưới và bên trái .

Chúng ta gọi hai mảng cố định đó là : Trên và Dưới . Thì ta có :

```
type
  arrtd = array[0..3, 0..3] of byte;
const
  tren: arrtd = ((0, 1, 1, 0),
                 (0, 1, 1, 0),
                    (3, 2, 2, 3),
              (3, 2, 2, 3));
  duoi: arrtd = ((0, 0, 3, 3),
                    (1, 1, 2, 2),
                    (1, 1, 2, 2),
                    (0, 0, 3, 3));
```

Chính vì thế chúng ta chỉ cần xác định được đường chéo của bảng thay đổi như thế nào thì tạo ra một bảng mới. Với một bảng mới, chúng ta sẽ lấy trường hợp có số lần xoay từ bảng cũ đến nó là ít nhất .

Bài toán 24: Boing

Đề bài :

Hẳn các bạn đã biết trò chơi boing . Tuy nhiên, khi thế kỉ của máy tính xuất hiện thì trò boing không còn thú vị lắm vì có sự hỗ trợ nên hầu như mọi lần ném đều đạt điểm tối đa. Chính vì thế trung tâm vui chơi giải trí đã đưa ra một trò chơi mới cải tiến từ trò boing .

Trò chơi mới được xuất phát từ một bảng với một số quân đặt ở trên bảng đó có 3 biên kín và một biên hở là đường bi vào . Ví Dụ :

Để đánh đổ được các quân ta phải ném thẳng ít nhất 3 quả bóng theo đường thẳng. Tuy nhiên trong trò chơi mới này, bạn lại có thể ném quả bóng vào bảng theo không chỉ hướng vuông góc với cạnh mà còn được phép ném quả bóng với một góc 45. như ví dụ trên, ta ném một quả bóng tại ô (4,3) với một góc 45 thì :

1. Quả bóng sẽ đi đến ô (3,4) và làm đổ quân tại ô (3,4)
2. Đập vào tường, đổi hướng khác đi đến (2,3) (1,2) và làm đổ quân tại ô (1,2)

3. Đập vào tường, đổi hướng đi đến ô(2,1) và làm đổ quân tại ô (2,1)
4. đập vào tường đi đến ô(3,2),(4,3) và đi ra ngoài bảng

Hãy tìm cách ném bóng sao cho số quả bóng cần ném là ít nhất để có thể đánh đổ được toàn bộ các quân đặt trong bảng .

Dữ liệu : Vào từ file văn bản Boing.Inp :

Dòng đầu tiên ghi hai số M, N thể hiện cho kích thước của bảng

M dòng kế tiếp, mỗi dòng gồm N số thể hiện cho bảng với số 1 thể hiện ô có quân và số 0 thể hiện cho ô không có quân

Kết quả : Ghi ra file văn bản Boing.Out :

Dòng đầu tiên là số quả bóng cần ném

Các dòng kế tiếp , mỗi dòng gồm 2 số thể hiện (ô xuất phát của quả bóng , hướng đi theo hình sau :

Ví Dụ :

4 4
0 1 0 0
1 0 0 0
0 0 0 1
0 0 0 0

Boing.Out :

1
3 1

Hướng dẫn :

Chúng ta vẫn duyệt các cách thả bóng tại mỗi vị trí có ba hướng. Nhưng chúng ta thấy rằng khi chúng ta thả hóng 1 ở một ô nào đó ở đáy thì nó sẽ đi ra ngoài bảng bảng hướng -1 ở một ô nào đó trên đáy (tức là tại vị trí này nếu bắn hướng -1. thì cũng cho kết quả hoàn toàn như bắn hướng 1 của từ ô kia). Chính vì thế chúng ta chỉ cần duyệt các cách có thể có của một hướng 1 và 0. Tức là tối đa trình duyệt sẽ chạy là 220 , nhưng thêm vào đó chúng ta thêm cận trên.

Bài toán 25 : ABC

Đề bài :

Cho một xâu S . Xâu này chỉ gồm các ký tự A , B , C và ngoài ra nó phải thoả mãn tính chất sau :

Không có hai xâu con nào liên tiếp trong S là giống nhau

Số kí tự B trong S là ít nhất có thể có

Yêu cầu : Hãy tìm một xâu S thoả mãn tính chất trên mà có độ dài cho trớc N (N<=250) .

Dữ liệu : Vào từ file ABC.INP :

Chỉ một số N duy nhất - là độ dài của xâu S cần tìm

Kết quả : Ghi ra file ABC.OUT :
Dòng đầu tiên là ghi số kí tự B ít nhất cần dùng
Dòng thứ hai ghi xâu S
Ví Dụ :

ABC.INP	ABC.OUT
10	2 ACABACBCAC

Hướng Dẫn :

Tại mỗi vị trí sẽ có 3 giá trị A,B,C . Nhưng thấy rằng , trong bốn kí tự liên tiếp của xâu S thì phải có ít nhất 1 kí tự B . Đây chính là công cụ chính, nó đã rút gọn chơng trình từ duyệt toàn bộ thành một bài toán duyệt mà tốc độ không tồi . Các bạn có thể cải tiến bài toán thành nhiều cách. Sau đây là một cách giải quyết rất hay cho những số N lớn hơn (250) nhưng độ dài xâu đó phải chia hết cho 4 :

Xâu S có các xâu con có 4 kí tự là :

ABCB : (1) ; ABCA : (2) ; ACBA : (3) ; ACBC : (4) ;
BACB : (5) ; BABC : (6) ; BCBA : (7) ; BCAB : (8) ;
CABA : (9) ; CABC : (10) ; CBAC : (11) ; CBAB : (12) ;

Ta sẽ thành lập một đồ thị 12 đỉnh như trên, trong đó đỉnh i sẽ nối với đỉnh j khi xâu j có thể được ghép sau xâu i. Ma trận kề của đồ thị đó nh sau :

(1) : 2 , 3 , 4 , 9 , 10 .
.....
(12) : 3 , 4 , 9 ,10 .

Trong đó khi chúng ta duyệt, tức là chúng ta sẽ lấy các xâu con này vào, như thế thì giảm đi rất nhiều bước duyệt . Mặt khác nhờ vào đồ thị, cũng như việc u tiên đặt vào của các xâu có ít kí tự B vào trước .

Bài toán 26 : Chuyên đề Tháp Hà Nội

Đề bài :

Có 3 cọc trên vòng tròn , được đánh số theo thứ tự chiều kim đồng hồ là 1 ,2 ,3. Có N đĩa kích thước từ lớn đến nhỏ, được xếp chồng đĩa nhỏ nằm trên đĩa lớn tại cọc 1

Yêu cầu : Cần chuyển N đĩa từ cọc 1 sang cọc 2 theo các quy tắc sau :

Mỗi bước chỉ chuyển một đĩa sang cọc sát cạnh nó theo chiều kim đồng hồ (1 sang 2 , 2 sang 3 , 3 sang 1)

Trong quá trình chuyển đĩa, chỉ cho phép đĩa nhỏ đặt lên trên đĩa lớn

Dữ liệu : file Disk.inp chứa số N (N <=15)

Kết quả : file Disk.Out : mỗi dòng mô tả một bước di chuyển đĩa gồm bắt đầu là s hiệu cọc xuất phát , tiếp theo là số hiệu cọc chuyển tới .

Ví Dụ :

DISK.INP
3
1 2 3 1 1 2 3 1 2 3 1 2 3 1 1 2

DISK.OUT
1 2 2 3 1 2 3 1 2 3 1 2 2 3

Hướng dẫn :

Gọi ba cọc đó là : A:[1..N], B[] , C[] . Ta cần chuyển ba cọc đó về trạng thái A[] ,B[1..n],C[] . Gọi Try(N,a,b) là thủ tục giải quyết bài toán tháp hà nội .

Trường hợp: vị trí b đứng sát vị trí a theo chiều kim đồng hồ .

Trường hợp này ta có các cặp (a,b) sau đây : (1,2) , (2,3) ,(3,1) . Đặc tả cho điều kiện này là b=a mod 3 + 1 .

(A[1..N],B[],C[]) - Cấu hình ban đầu . Để chuyển N tầng từ a sang b theo chiều kim đồng hồ ta phải :

(A[N],B[],C[1..N-1])-Chuyển tạm n-1 tầng từ a qua c

(A[],B[N],C[1..N-1])-Sau đó chuyển tầng còn lại của a qua b

(A[],B[1..N],C[])- Cuối cùng chuyển N-1 tầng từ c sang b hoàn tất .

Ta có thể mô tả bằng chương trình :

```
Try(N-1,i,k,j);
 Ghi(i,j);
 Try(N-1,k,j,i);
```

Trường hợp : vị trí b không đứng sát a theo chiều kim đồng hồ .

Các cặp (a,b) có thể có là : (1,3),(2,1) ,(3,2) . Đặc điểm chung của các cặp này là : Nếu đi từ a đến b theo chiều kim đồng hồ chúng ta phải vợt qua c là vị trí nằm giữa a và b. Cấu hình buộc phải có :

(A[1..N],C[],B[])- Để chuyển N tầng từ a sang b cách a qua vị trí c ta phải :

(A[N],C[],B[1..N-1])-Chuyển tạm N-1 tầng từ a qua b

(A[],C[N],B[1..N-1])-Sau đó chuyển tạm tầng còn lại ở a qua c

(A[1..N-1],C[N],B[])- Rồi lại chuyển tạm N-1 tầng từ b qua a nhằm giải phóng vị trí đích b

(A[1,,N-1],C[],B[N])- Chuyển tầng lớn nhất N từ c sang b

(A[],C[],B[1..N]) - Cuối cùng chuyển N-1 tầng từ a qua b là hoàn tất .

Ta có thể mô tả bằng chương trình :

```
Try(N-1,i,j,k);
 Ghi(i,k);
 Try(N-1,j,i,k);
 Ghi(C,B);
 Try(N-1,i,j,k);
```

Mở rộng :

Chúng ta có thể mở rộng bài toán trên một cách tổng quát và khó hơn nhiều :

Vẫn với điều kiện như bài toán trên, chúng ta lại có thêm điều kiện ;

Các đĩa có một màu nhất định nào đó trong số ba màu : xanh(X),tím (T),Hồng (H) , và Nâu (N). Nhưng với mỗi một màu có một cách chuyển như sau :

+ Với màu xanh : thì chỉ được chuyển vòng xuôi (cọc 1 sang cọc 2 , cọc 2 sang cọc 3 , cọc 3 sang cọc 1)

+ Với màu Tím :thì chỉ được chuyển sang các cọc bên cạnh nó (1 sang 2 , 2 sang 3 và ngọc lại)

+ Với màu Hồng thì được chuyển sang các cọc bất kỳ

+ Với màu Nâu thì chỉ được chuyển vòng ngược (2 sang 1 , 3 sang 2 , 1 sang 3) .

Chúng ta có thể tổng quát thủ tục try(N,i,j) nh sau :

```
procedure try(i,a,b:Integer);
begin
   if i>0 then
   begin
      case st[i] of
       'x':
          begin
             if b= a mod 3+1 then
             begin
                try(i-1,a,6-a-b); ghi(a,b);try(i-1,6-a-b,b);
             end
             else
             begin
                try(i-1,a,b);ghi(a,6-a-b);try(i-1,b,a);
                ghi(6-a-b,b); try(i-1,a,b);
             end;
          end;
       't':
          begin
             if abs(a-b)=1 then
             begin
                try(i-1,a,6-a-b); ghi(a,b); try(i-1,6-a-b,b);
             end
             else
             begin
                try(i-1,a,b);ghi(a,6-a-b);try(i-1,b,a);ghi(6-a-b,b);
```

```
                try(i-1,a,b);
            end;
        end;
    'h':
        begin
            try(i-1,a,6-a-b);ghi(a,b);try(i-1,6-a-b,b);
        end;
    'n':
        begin
            if a=b mod 3 +1 then
            begin
                try(i-1,a,6-a-b); ghi(a,b); try(i-1,6-a-b,b);
            end
            else
            begin
                try(i-1,a,b);ghi(a,6-a-b);try(i-1,b,a);ghi(6-a-b,b);
                try(i-1,a,b);
            end;
        end;
    end;
end;
end;
```

Bài toán 27 : Nổ Mìn {nguyễn xuân my}

Đề bài :

Cho một sân hình chữ nhật có kích thước MxN được chia thành MxN ô vuông bằng nhau bởi các đường song song với các cạnh. Trên mỗi ô vuông có thể chôn hoặc không chôn một quả mìn. Tình trạng của bãi mìn có thể được mô tả bởi một trong hai cách sau:

Cách thứ nhất: Dùng một mảng MIN1[1..M,1..N] trong đó phần tử MIN1[i,j] = 1 hay 0 tuỳ theo ô [i,j] có mìn hay không.

Cách thứ hai: Mỗi ô của sân có nhiều nhất 8 ô khác nó kề cạnh với nó như trong hình 1.

```
1     2     4          1     0     1          16   184  96
128   ô i,j 8          0     1     1          58   109  194
64    32    16         1     1     0          12   134  131
```

Hình 1 *Hình 2 (bên trái: MIN1, bên phải: MIN2)*

Ta ứng các ô này với các số 1, 2, 4, 8, 16, 32, 64, 128 như hình 1. Khi đó ta lập một mảng MIN2[1..M,1..N] nh sau: MIN2[i,j] bằng tổng các số hạng có dạng X.Y trong đó X là số ứng với một ô kề cạnh với nó, Y bằng 1 hay 0 tuỳ theo ô đó có hay không có mìn, *ô [i,j] có bao nhiêu ô kề cạnh*

thì tổng có bấy nhiêu số hạng. Trong hình 2 cho ví dụ về cách lập mảng MIN2 ứng với tình trạng mìn cho bởi mảng MIN1.
Viết chương trình làm các việc sau:

1. Đọc từ file MIN1.TXT mảng MIN1 cho biết tình trạng mìn theo cách thứ nhất, trong đó dòng thứ nhất ghi hai số nguyên dương M, N <=100, trong M dòng sau, dòng thứ i ghi N số lần lượt là MIN1[i,1], . ., MIN1[i,N]. Ghi ra file MIN1-2.TXT mảng MIN2 thể hiện tình trạng mìn theo cách thứ hai, mỗi dòng của mảng MIN2 ghi thành một dòng của file.
2. Đọc từ file MIN2.TXT mảng MIN2 cho biết tình trạng mìn theo cách thứ nhất, trong đó dòng thứ nhất ghi hai số nguyên dương M, N <=100, trong M dòng sau, dòng thứ i ghi N số lần lượt là MIN2[i,1], . ., MIN2[i,N]. Ghi ra file MIN2-1.TXT mảng MIN1 thể hiện tình trạng mìn theo cách thứ hai, mỗi dòng của mảng MIN1 ghi thành một dòng của file.

Hướng dẫn :

Đối với câu 2 , thì Chúng ta sẽ chỉ cần duyệt một hàng (cột) ở biên là đợc . Bằng cách thành lập phương trình Gaus (hoàn toàn giống bài toán Turn Around). Thì chúng ta chỉ cần tìm số mìn ở hàng biên (giả sử là hàng 1). Thì từ hàng1 ta sẽ xác định được hàng thứ hai, mà từ hàng thứ hai ta có thể xác định được hàng thứ 3,....Cứ như thế ta sẽ tìm được cả bảng. Ta sẽ chọn hàng để duyệt khi M <N Hoặc cột để duyệt khi N<M. Ngoài ra sử dụng cận trên để chặn các bước duyệt .

Bài toán 28 : Robot quét vôi

Đề bài :

Có 9 căn phòng (đánh số từ 1 đến 9) đã được quét vôi với mầu trắng, xanh hoặc vàng. Có 9 rôbôt (đánh số từ 1 đến 9) phụ trách việc quét vôi các phòng. Mỗi rôbôt chỉ quét vôi một số phòng nhất định. Việc quét vôi được thực hiện nhờ một chương trình cài sẵn theo qui tắc:

- nếu phòng đang có mầu trắng thì quét mầu xanh,
- nếu phòng đang có mầu xanh thì quét mầu vàng,
- nếu phòng đang có mầu vàng thì quét mầu trắng.

Cần phải gọi lần lượt một số các rôbôt ra quét vôi (mỗi lần một rôbôt, một rôbôt có thể gọi nhiều lần và có thể có rôbôt không được gọi. Rôbôt được gọi sẽ quét vôi tất cả các phòng mà nó phụ trách) để cuối cùng các phòng đều có mầu trắng. Hãy tìm một phương án như vậy sao cho lượng vôi phải quét là ít nhất. Giả thiết rằng lượng vôi cho mỗi lượt quét đối với các phòng là như nhau.

Dữ liệu: Đọc từ file văn bản ROBOT.INP gồm các dòng:

- 9 dòng đầu, mỗi dòng mô tả danh sách các phòng được quét vôi bởi một rôbôt theo thứ tự từ rôbôt 1 đến rôbôt 9. Mỗi dòng như vậy gồm các số hiệu phòng viết sát nhau. Chẳng hạn dòng thứ 2 có nội dung:

3578

mô tả rôbôt 2 phụ trách việc quét vôi các phòng 3, 5, 7, 8.

- dòng cuối mô tả mầu vôi ban đầu của các phòng. Dòng gồm 9 ký tự viết sát nhau, ký tự thứ i biểu diễn mầu vôi của phòng i với quy ước: ký tự T chỉ mầu trắng, ký tự X chỉ mầu xanh, ký tự V chỉ mầu vàng.

Kết quả: Đưa ra file văn bản ROBOT.OUT gồm một dòng dưới dạng:

- nếu không có phơng án thì ghi một chữ số 0,
- trái lại ghi dãy thứ tự các rôbôt được gọi (các số hiệu rôbôt viết sát nhau).

Thí dụ:

ROBOT.INP	ROBOT.OUT
159 123 357 147 5 369 456 789 258 XVXVXVTXT	2455688

Hướng dẫn :

Chúng ta sẽ tìm hết tất cả các cách quét của các con robot. Nhưng chúng ta thấy nếu gọi một con robot quá 3 lần thì nó sẽ lặp lại như cách gọi 0 , 1 , 2 lần. Như vậy với mỗi con chúng ta có ba khả năng 0 , 1 , 2 số lần tham gia quét . Như vậy chơng trình chạy với 39 phép toán thì không phải là lớn .

Mở :

Đối với bài toán N con robot và N phòng thì chúng ta có cách giải tổng quát sau :

Chúng ta sẽ khai triển nó ra dạng bài toán Gaus thì hoàn có thuật toán :

Trước tiên chúng ta quy định trạng thái của một phòng như sau :

P[i]=0,1,2 . Nếu như màu của nó là T,X,V .

Gọi A[i,j] là số :

A[i,j]=0 nếu như robot i không quét phòng j và ngược lại thì A[i,j]=1 .

T[i] là số lần gọi robot i .

Ta sẽ có hệ :

(A[1,1]*T[1]+A[2,1]*T[2]+..+A[N,1]*T[N]+K[1])Mod 3=0;
(A[1,2]*T[1]+A[2,2]*T[2]+..+A[N,2]*T[N]+K[2])Mod 3=0;
.....
(A[1,N]*T[1]+A[2,N]*T[2]+...+A[N,N]*T[N]+K[N])Mod 3 = 0 ;

Đây là hệ N ẩn : T[1],T[2],...T[N] và có N phương trình trên. áp dụng Cho Gaus ta có thể giải quyết bài toán .

Bài toán 29 : Adequate Square

Đề bài :

Một bảng vuông A kích thước N được gọi là đầy đủ bạc N nếu mọi số trong bảng A nằm trong khoảng [1..N2] và đôi một khác nhau . Ví Dụ : bảng sau là bảng đầu đủ bậc 3

1 9 6	1 6 7 4
8 2 7	3 2 8 1
4 3 5	4 5 9 3
	1 6 7 4
A	B

Một bảng vuông B kích thước m được gọi đầy đủ bậc n nếu mọi bảng con vuông kích thước N của B đều là đầy đủ bậc N. Ví dụ bảng B là bảng đầy đủ bậc 3

Yêu cầu : Cho một bảng vuông kích thước m đã điền một số vị trí . Hãy tìm cách điền vào những ô còn lại để được bảng vuông đầy đủ bậc N .

Dữ liệu : File văn bản Square.inp :

Dòng đầu tiên là số M , N

M dòng tiếp theo, mỗi dòng là M số thể hiện cho bảng ban đầu với số 0 thể hiện cho ô chưa điền

Kết quả : File văn bản Square.Out:

Dòng đầu tiên ghi số -1 nếu không điền được. Số 1 cho trường hợp điền được.

M dòng kế tiếp , mỗi dòng là N số thể hiện cho ma trận sau khi điền

Ví Dụ :

SQUARE.INP	SQUARE.OUT
4 3 1 0 7 4 3 2 8 0 0 5 9 3 1 6 7 4	1 1 6 7 4 3

Hướng dẫn :

Chúng ta sẽ duyệt từng vị trí một của bảng với các giá trị có thể có. Nếu có kết quả thì chúng ta sẽ thoát hẳn khỏi chương trình . Nhưng trong quá trình duyệt chúng ta sẽ dùng các hàm kiểm tra để ma trận thoả mãn ma trận đầy đủ bậc N. Chúng ta thấy điều sau :

a1	a2	a3
a4	a5	a6

a7 a8 a9
a10 a11 a12

Ta xét ví dụ một phần của ma trận nào đó chứa đồ thị đầy đủ bậc 3 (ví dụ). ta sẽ có :
(a1,a2,a3) và (a10,a11,a12) chính là hoán vị của nhau. Hoàn toàn tương tự cho các cột của mảng. Nếu vậy khi chúng ta biết được một hàng nào đó thì ta sẽ hạn chế đi rất nhiều bước duyệt .
Mặt khác ta còn có :

a1 a2 a3
a4 a5 a6
a7 a8 a9
a10 a11 a12
a13 a14 a15
a16 a17 a18

Chúng ta sẽ có giả sữ chúng ta biết được đã có số a1(hoặc là a2 hoặc a3) thì chắc chắn a10 ,a11,a12 sẽ phải có số đó . Chính vì thế , a13,a14,a15 ,a16,a17,a18 sẽ không thể là các số đó . Vậy chúng ta sẽ dùng một thủ tục kiểm tra toàn bộ bảng . Như vậy sẽ giảm rất nhiều số bước duyệt .

Bài toán 30 : Nhà du hành vũ trụ

Đề bài:

Một nhà du hành vũ trụ bị lạc vào một hành tinh được thống trị bởi các con rô bốt (người máy). Giả sử hành tinh được chia thành một mảng chữ nhật các ô vuông mà mảng đó có kích thước M x N (M dòng, N cột; M, N 8). Tất cả mọi con rô bôt tìm cách tiến lại gần nhà du hành để tiêu diệt, còn nhà du hành tìm cách tránh xa các con rô bốt. Cứ sau một đơn vị thời gian thì nhà du hành có quyền đi theo 4 hướng Đông, Tây, Nam, Bắc hoặc đứng yên. Tất cả các con rôbôt được lập trình đi theo 1 trong 8 hướng (kể cả đờng chéo) sao cho khoảng cách từ con rôbôt đến nhà du hành là nhỏ nhất (khoảng cách này không lớn hơn khoảng cách ở thời điểm trước) trong đó đường chéo được tính theo công thức hình học thông thường. Các con rôbôt luôn chuyển động theo nhịp thời gian.

Nếu hai con rôbôt cùng đến 1 ô thì cả hai con rôbôt này bị nổ tung và để lại trên ô đó một nhiệt độ rất lớn (chẳng hạn là 1tỷ0C) được duy trì mãi và nhiệt độ này đủ để phá huỷ các con rôbôt hoặc nhà du hành chẳng may đặt chân lên ô đó. Ô này trở thành ô nguy hiểm.

Tất cả các con rôbôt không được lập trình để tránh các ô nguy hiểm hoặc tránh đụng độ nhau mà chỉ đươc lập trình để tiến gần một cách máy móc đến nhà du hành.

Khi 1 hoặc nhiều rôbôt cùng nhà du hành ở cùng một ô vuông thì nhà du hành bị tiêu diệt. Nhà du hành không thể đi tới một ô đang có rôbôt

Bài toán đặt ra: Cho bản đồ của hành tinh (có vị trí nhà du hành và các con rôbôt). Hãy tìm cách di chuyển nhà du hành theo một lịch trình sao cho nhà du hành có thể tồn tại lâu nhất trên hành tinh (thông báo trường hợp nhà du hành theo một lịch trình sao cho nhà du hành có thể tồn tại lâu nhất trên hành tinh (thông báo trường hợp nhà du hành tồn tại mãi mãi trên hành tinh).

Dữ liệu: Vào có trong file văn bản INP.B2 có cấu trúc:

- Dòng đầu tiên chứa hai số M và N.
- M dòng tiếp theo chứa bản đồ của hành tinh: Mỗi dòng chứa một xâu văn bản (đặt ngay từ dòng đầu) gồm các kí tự '0', '1', '2' với ý nghĩa sau:

'0': ô rỗng '1': ô có 1 rô bốt '2': ô có nhà du hành

Kết quả: Ghi ra lên màn hình và ra file OUT.B2 có dạng sau:

- Theo từng thời điểm của lịch trình đã chọn cho nhà du hành, hãy thể hiện lên màn hình số liệu thời điểm và bản đồ hành tinh tại thời điểm đó. Ô nguy hiểm có dấu '*'.
- Ghi ra file theo nội dung sau:

+ Dòng đầu ghi số đơn vị thời gian nhà du hành tồn tại trên hành tinh (quy ước ghi -1 nếu anh ta tồn tại mãi mãi).

+ ứng với mỗi thời điểm có M dòng trên file bản đồ hành tinh trong lịch trình. Ví dụ, phần đầu của một lịch trình được thể hiện nh sau:

00000000	00000000	00000000
01000000	00000000	00021000
00002000	00*20100	00*00000
01000010	01000000	000*0000
10000000	00111000	00010000
01010100	00010000	00000000
00100000	00000000	00000000

Hướng dẫn :

Chúng ta có một khẳng định sau : Nếu người tồn tại sau K=min(M,N) bước thì người tồn tại mãi hay “Nếu người bị robot ăn thịt thì không thể tồn tại sau K bước . Chứng minh điều này như sau :

Sau mỗi bước hoặc robot gần người thêm một đơn vị khoảng cách hoặc không gian người bị hạn chế thêm một đơn vị độ dài (cả về chiều rộng lẫn chiều dài) .

Với khẳng định này thì số bước lặp không quá K bước trong đó mỗi bước lặp liên quan đến việc người thực hiện 5 cách đi trong bước . Như vậy tổng số bước không vượt quá 5K bước thuật toán. Cho nên chúng ta duyệt từng bước đi của người đó . (Sử dụng thêm mảng để lưu lại các giá trị của mảng).

Bài toán 31 : Đôminô

Đề bài :

Bộ bài domino gồm 28 quân đánh số từ 1 đến 28. Mỗi quân bài là một thanh hình chữ nhật được chia làm hai hình vuông bằng nhau, trong đó người ta ghi các số từ 0 (để trống) đến 6 bằng cách trổ các dấu tròn trắng. Dưới đây liệt kê 28 quân bài domino :

Số quân bài	TT quân bài Số	Số TT quân bài		TT quân bài		Số	TT
01	0/0	08	1/1	15	2/3	22	3/6
02	0/1	09	1/2	16	2/4	23	4/4
03	0/2	10	1/3	17	2/5	24	4/5
04	0/3	11	1/4	18	2/6	25	4/6
05	0/4	12	1/5	19	3/3	26	5/5
06	0/5	13	1/6	20	3/4	27	5/6
07	0/6	14	2/2	21	3/5	28	6/6

Sắp xếp 28 quân bài domino ta có thể tạo ra một hình chữ nhật kích thước 7*8 ô vuông. Mỗi cách xếp như vậy sẽ tạo ra một bản đồ số. Ngược lại, mỗi bản đồ số có thể tương ứng với một số cách xếp .

Ví Dụ : Bản đồ số

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

Tương ứng với hai cách xếp mô tả bởi số :

16	16	24	18	18	20	12	11		16	16	24	18	18	20	12	11

06	06	24	10	10	20	12	11		06	06	24	10	10	20	12	11
08	15	15	03	03	17	14	14		08	15	15	03	03	17	14	14
08	05	05	02	19	17	28	26		08	05	05	02	19	17	28	26
23	01	13	02	19	07	28	26		23	01	13	02	19	07	28	26
23	01	13	25	25	07	04	04		23	01	13	25	25	07	21	04
27	27	22	22	09	09	21	21		27	27	22	22	09	09	21	04

Yêu cầu : Cho trước bản đồ số, hãy liệt kê tất cả các cách xếp có thể tạo được bản đồ số đó .

Dữ liệu : Vào từ file DOMINO.INP gồm 7 dòng, mỗi dòng chứa các phần tử của dòng tương ứng trong bản đồ số .

Kết quả : Ghi ra file văn bản DOMINO.OUT :

Dòng đầu tiên là số lượng cách xếp P

Sau đó là P nhóm dòng, mỗi nhóm dòng gồm 7 dòng ghi các dòng của các bảng số tương ứng với một cách xếp tìm được. Các bộ số cách nhau một dòng trắng .

Hướng dẫn :

Chúng ta sẽ duyệt bảng theo tuần tự từ trên xuống dưới, từ trái sang phải. Mối lần như thế ta sẽ đánh dấu các ô đã được chọn và các vị trí đã có quân bài. Cứ như thế chúng ta phải duyệt toàn bộ để tìm tất cả các kết quả. Nên sử dụng một file phụ để lưu kết quả có thể có .

Bài toán 32 : Sơ đồ sterberg

Đề Bài :

Giáo s Jacob Tractenberg xây dựng một sơ đồ cho phép tính một cách nhanh chóng và đơn giản tích 2 số nguyên . Ví Dụ : để tính 23*14 ta phải dựa trên bộ dữ liệu trung gian 12 , 11 , và 2 . Các số trung gian này độcgị là bộ dữ liệu Tractenberg . Từ bộ dữ liệu này , ta có thể dễ dàng suy ra tích cần tìm là 322 .

Để hiểu thêm về bộ dữ liệu Tractenberg , ta xét thêm một số ví dụ :

241304 x 32 => 8 12 6 11 11 16 6 => 7721728

527 x 463=> 21 48 55 38 20 => 244001

3214 x 5643 => 12 19 34 43 29 28 15 => 18136602

1245 x 8 => 40 32 16 8 => 9960

Yêu Cầu : Hãy xác định cách xây dựng bộ dữ liệu Tractenberg và từ bộ dữ liệu này xác định các cặp thừa số cũng như tích của chúng

Dữ liệu : Vào từ file văn bản Multp.Inp các số của 1 bộ dữ liệu Tractenberg , số lượng số không quá 50, ghi trên một hay nhiều dòng (cách nhau ít nhất một dấu trắng)

Kết quả : Ghi ra file Multp.Out thông tin dạng tích sau :

Thừa số 1 * Thừa số 2 = tích

Mỗi dòng tương ứng với một kết quả khác nhau tìm được. Các thừa số không chứa số 0 không có nghĩa ở đầu. Hai kết quả được coi là giống nhau, nếu chỉ khác nhau ở trình tự thừa số trong tích .
Ví Dụ :

MULTP.INP	MULTP.OUT
8 12 6 11 11 16 6	241304*32=7721728

Hướng dẫn :

Duyệt các vị trí của các hàng đơn vị trở lên, có những cặp số nào có tích bằng số thoả mãn thì chúng ta sẽ lấy .

Bài toán 33 : Lát Gạch

Đề bài :

Cho một lưới hình chữ nhật gồm MxN (2<=M,N<=10) ô vuông có cạnh bằng đơn vị (ở đây được gọi là ô vuông đơn vị). Hãy tìm tất cả các cách lắp đầy các ô vuông của lưới bởi các khối được ghép bởi các ô vuông đơn vị có hình dạng như sau :

Yêu cầu : Mỗi khối (bao gồm cả các khối thu đợc từ 2 khối đã cho thông qua một phép quay 900 ,1800 ,2700) chỉ được sử dụng 1 lần .
Dữ liệu : Vào từ file Khoi.Inp :
Dòng thứ nhất ghi số M
Dòng thứ hai ghi số N
Kết quả : GHi ra file Khoi.OUT :
In ra các cách sắp xếp có thể
Nếu không thể xếp đợc thì ghi " No solution "

Ví Dụ :

KHOI.INP	KHOI.OUT
4 4	1 1 1 2 3 1 2 2 3 3
4 2 3 4 4 4 hoặc 1 2 2 2 1 1 2 3 1 4 3 3 4 4 4 3	

Hướng dẫn :

Duyệt các cách để vào các vị trí có thể có .

Bài toán 34 : Bật bít

Đề bài :

Cho mảng A[NxN] , gồm các ký tự 0 và 1. Mỗi một ô trên đó là một bit. Biết rằng khi chúng ta tác động lên một bít nào đó , thì tất hẳn bốn bít bên cạnh (có chung cạnh với nó) sẽ bị đổi bit (tức là 0 thành 1 và 1 thành 0) và cả nó nữa . Người ta muốn điều khiển bảng bit đó thành một bảng nào đó , thế nhưng cần biến đổi sao cho số lần tác động bít là ít nhất .

Dữ liệu : Vào từ file BatBit.Inp :

Dòng đầu tiên ghi số N (N<=50)

N dòng sau , mỗi dòng ghi N số biểu diễn bảng bít ban đầu

N dòng sau ghi bảng bit cần biến đổi tới .

Kết quả : Ghi ra file BatBit.Out :

Dòng đầu tiên ghi 'YES' Hoặc 'NO' nếu nh có thể và không thể bật bit về nhau

Nếu có thì làm các yêu cầu sau :

Dòng kế tiếp ghi số lần bấm

Các dòng còn lại ghi toạ độ các ô cần bật bit

Ví dụ :

BATBIT.INP BATBIT.OUT

4 0 1 0 1 1 0 0 0 0 0 0 1 1 1 1 0 0 0 0 1 0 1 0 0 0 0 1 0 1 1 0 0 YES 2 2

Hướng dẫn :

Gọi c[i,j] là số lần bật bit của ô [i,j] . Thế nhưng ta có một đặc điểm rất hay: Khi chúng ta biết được C[1,i] , i =1,..N thì ta có thể biết được toàn bộ mảng c[i,j] .

Thật vậy , khi C[1,1] và c[1,2] biết thì ta biết được C[2,1] , tương tự như vậy ta sẽ biết được hết hàng2 , tương tự cho đến hàng thứ N .

Nói cách khác , bài toán chỉ cần xác định số phép biến đổi bit ở các hàng biên thì chúng ta hoàn toàn xác định được cả bảng. Để thuận lợi , trước tiên chúng ta sẽ xét các hệ phương trình để giải cho các số ở biên. Nếu bên nào thoả mãn chỉ có một nghiệm thì chúng ta lấy nó làm nghiệm bài toán , ngược lại ta xét duyệt các phép toán có thể có của một đường bên (giả sử hàng trên cùng) .

Giống bài toán Turn Around ,..

Bài toán 35 : Các thành phần liên thông

Đề bài :

Cho một lưới A gồm MxN ô vuông. Trên ô [i,j] của lới, 1<=i<=M, 1<=j<=N, ta ghi một số nguyên dương A[i,j]. Từ mảng A[1..M,1..N] ta lập mảng B [1..M,1..N] với B[i,j] = f(A[i,j]) là số lượng các số hạng trong biểu diễn A[i,j] thành tổng các số nguyên dương sao cho tích các số hạng

đó đạt giá trị lớn nhất, nếu có nhiều biểu diễn có cùng tích lớn nhất, ta chọn biểu diễn có ít số hạng nhất. Ví dụ f(1)=1, f(4)=1, f(8)=3.

Câu 1. Cho trước mảng A, hãy xây dựng mảng B.

Câu 2. Ta làm việc với mảng B nhận được từ câu 1.

Một tập hợp H các ô trong mảng B được gọi là một *hình* nếu nó thoả mãn các điều kiện sau:

H1. Từ một ô bất kỳ của H ta có thể di chuyển đến một ô bất kỳ của H bằng một dãy các di chuyển qua các ô kề cạnh cũng thuộc H. Từ một ô ta có thể di chuyển đến một ô kề cạnh với nó nếu trên hai ô ghi hai số nguyên tố cùng nhau,

H2. Nếu thêm vào H một ô bất kỳ không thuộc H, điều kiện H1 bị vi phạm.

Một tập hợp D các ô (của mảng B) đợc gọi là một *lỗ thủng* của hình H nếu các điều kiện sau được thoả mãn:

T1. Không có ô nào của D thuộc H và mọi ô của D đều không là ô biên của lới B,

T2. Mỗi ô kề cạnh với một ô bất kỳ của D đều hoặc thuộc D hoặc thuộc H,

T3. Từ một ô bất kỳ của D ta có thể di chuyển đến một ô bất kỳ khác của D (không nhất thiết hai số ghi trên hai ô đó nguyên tố cùng nhau) qua một số ô kề cạnh cũng thuộc D,

T4. Nếu thêm vào D một ô không thuộc nó, một trong các điều kiện T1-T3 bị vi phạm.

Yêu cầu : Hãy tìm các hình của B với số lỗ thủng nhiều nhất (số lỗ thủng nhiều nhất có thể bằng 0).

Dữ liệu : Vào được cho bởi file INP.B1 gồm M dòng, mỗi dòng ghi N số nguyên dương thể hiện mảng A cho trước, M, N<=50.

Kết quả : Ghi ra các file theo quy định sau:

Câu 1. Ghi mảng B ra M dòng của file OUT1.B1, mỗi dòng của mảng ghi trên một dòng của file theo thứ tự từ dòng thứ nhất đến dòng thứ M.

Câu 2. Đọc dữ liệu vào từ file OUT1.B1 và ghi kết quả ra file OUT2.B1. Dòng thứ nhất ghi hai số U và V, số U là số lượng hình (có số lỗ thủng nhiều nhất), số V là số lỗ thủng. Tiếp theo là U nhóm dòng, mỗi nhóm gồm M dòng liên tiếp thể hiện một hình, dòng thứ i trong nhóm ghi N số 0 hoặc 1, số 0 thể hiện ô tương ứng không thuộc hình, số 1 thể hiện ô tương ứng thuộc hình.

Hướng dẫn :

Chúng ta loang các miền có thể có của hình đã cho . Rồi sau đó loang một lần nữa để tìm các hình cần tìm .

Bài toán 36 : Bảng Dữ liệu

Đề bài :

Cho một bảng hình chữ nhật kích thước M x N với các cạnh M, N là các số nguyên dương, M, N 250. Hình chữ nhật này được chia thành M x N ô vuông bằng nhau với kích thớc đơn vị, bởi các đường song song với các cạnh, trên ô vuông [i, j] ghi số nguyên A [i, j] 50.

Từ mảng A hãy lập mảng B mà B [i, j] đợc tính như sau: biểu diễn số nguyên không âm A [i, j] thành tổng nhiều nhất các số nguyên tố trong đó có nhiều nhất chỉ một số nguyên tố có thể được xuất hiện 2 lần, B [i, j] bằng số hạng của biểu diễn này kể cả số bội (nếu có).

Ví dụ: Nếu A[i, j] = 10 = 2 + 3 + 5 thì B [i, j] = 3

Nếu A[i, j] = 12 = 2 + 2 + 3 + 5 thì B [i, j] = 4

Chú ý: không biểu diễn A[i, j] = 10 = 2 + 2 + 2 + 2 + 2 để có B[i, j] = 5 vì số 2 đã xuất hiện quá 2 lần; cũng vậy, không biểu diễn A[i, j] = 10 = 2 + 2 + 3 + 3 để có B [i, j] = 4 vì có hai số là số 2 và số 3 đều xuất hiện quá 1 lần.

Bài toán đặt ra: Hãy tìm hình chữ nhật lớn nhất gồm các ô của mảng B ghi cùng các số như nhau.

Dữ liệu: Vào từ File INP.B4 trong đó:

Dòng đầu ghi 2 số M và N.

M dòng sau ghi M dòng của mảng A (không cần kiểm tra dữ liệu).

Kết quả: Ghi vào File OUT.B4.

Giá trị của mảng B, mỗi dòng của File là một dòng của bảng.

Ghi tiếp ra File OUT.B4 một dòng gồm 5 số là: diện tích lớn nhất tìm được, toạ độ trên trái và dưới phải của hình chữ nhật có diện tích lớn nhất đó.

(yêu cầu thời gian chạy với một bộ dữ liệu không quá 5 giây)

File INP.B4

```
8  10 3  3  3  3  3  3  3  3  3  3 3  3  3  3  3  3  3  3  3  3 3  3  3  3  3  3  3  3
3  3 3  3  3  3  3  3  3  3  3  3 3  3  3  3  3  3  3  3  3  3 3  3  3  3  3  3  3  3
3  3 3  3  3  3  3  3  3  3  3  3 3  3  3  3  3  3  3  3  3  3
80  1  1  8  10
10  10
15  10  14  11  11  11  13  10  10  14
15  13  15  13  14  14  13  14  14  12
10  12  12  12  15  15  12  11  11  13
15  11  15  10  15  11  10  10  12  13
13  12  14  13  11  13  10  14  14  12
11  15  13  13  12  13  10  12  12  15
15  11  12  13  13  10  11  15  13  15
14  14  11  12  12  15  12  14  13  13
```

File OUT.B4

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

14 15 11 11 11 15 10 14 10 12 3 4 3 4 4 3 3 3 4 4
11 13 11 12 15 10 10 11 15 15 12 2 2 3 7

Hướng dẫn :

Đầu tiên chúng ta dùng quy hoạch động để phân tích các số đã cho thành các số nguyên tố . Sau đó xây dựng bảng cần tìm . Chúng ta sẽ giải bài toán hình chữ nhật có diện tích cực đại của bảng :

Bài toán : *Cho một tệp văn bản có tên CNMAX.INP gồm các dòng dài bằng nhau, mỗi dòng có đúng m ký tự và không chứa dấu cách. Dòng đầu tiên của tệp chứa số m. Tìm hình chữ nhật chứa cùng một loại ký tự và có diện tích lớn nhất. Biết 3 m 78 và diện tích lớn nhất có thể đạt tới chữ số hàng tỷ. Kết quả cần đợc hiển thị trên màn hình theo dạng sau:*

Dòng đầu tiên: Diện tích của hình chữ nhật tối đại ABCD tìm được,

Dòng thứ hai: Toạ độ dòng và cột của đỉnh A,

Dòng thứ ba: Toạ độ dòng và cột của đỉnh C.

CNMAX.INP
20
bcccddddeabc00000000
bbbbbccccccccccbbbbb
00000ccccccccccccbb
00ccccccccccccbbbbb
44444ccccccccccabbbb
44444ccccccccccczzzzz
33ccccccccccccczzzzz
33333ccccccccccccczz
22222ccccccccccczzzzz
uuuuuuuuczzzzzzzzzzz

Hình 2

C. Thí dụ, với dữ liệu được cho như hình 2 thì kết quả phải đạt sẽ như sau:

80
2 6
9 15

Tính tổng quát của bài toán 2 thể hiện ở các điểm sau:

Số dòng của tệp CNMAX.INP là *không hạn chế*, tức là ta không thể đọc toàn bộ dữ liệu vào một mảng để xử lý như trong bài toán 1,

Tệp không chỉ chứa các ký tự *0/1*.

Bài giải. Do không thể đọc toàn bộ dữ liệu từ tệp CMAX.INP vào một mảng để xử lý nh trong bài toán 1 nên chúng ta sẽ đọc mỗi lần một dòng vào biến kiểu xâu ký tự *y*. Vì hình chữ nhật cần tìm chứa cùng một loại ký tự cho nên các dòng của hình sẽ liên thông nhau. Để phát hiện tính liên thông chúng ta cần dùng thêm một biến kiểu xâu ký tự *x* để lưu dòng đã đọc và xử lý ở bước trước. Tóm lại là ta cần xử lý đồng thời hai dòng: *x* là *dòng trước* và *y* là *dòng đang xét*.

Nếu xét các cột trong hình chữ nhật cần tìm ta thấy chúng phải chứa cùng một loại ký tự. Ta dùng một mảng *h* với ý nghĩa phần tử *h[i]* của mảng cho biết tính từ vị trí thứ i của dòng *y* trở lên có bao nhiêu ký tự giống nhau (và giống với ký tự *y[i]*). Ta gọi *h[i]* là độ cao của cột *i* và mảng *h* khi đó sẽ được gọi là độ cao của dòng đang xét.

Thí dụ, độ cao của dòng thứ 5 trong hình 2 sẽ là:

h = (1,1,1,1,1,1,4,4,4,4,4,4,4,5,4,4,4, 1,2,2,4,4)

Biết độ cao, với hai dòng *x* và *y* chúng ta dễ dàng tính được diện tích của mỗi hình chữ nhật ABCD chứa phần tử *y[i]* trên cạnh DC. Thật vậy, giả sử ta đang xét ký tự thứ *i = 8* trên dòng thứ 5 nh đã nói ở phần trên. Ta có *h[8] = h[7] = h[6] = 4; h[9] = h[10] = h[11] = 4; h[12] = 5; h[13] = h[14] = h[15] = 4; h[16] = 1* ... Vậy thì, khi ta đi từ *i* về hai phía, trái và phải, nếu gặp các ký tự giống ký tự *y[i]* còn độ cao thì không nhỏ hơn *h[i]* ta sẽ thu được hình chữ nhật lớn nhất chứa ký tự *y[i]*.

Với dòng thứ 5 đang xét, ta có:

x = ‘00ccccccccccccbbbbb’;
y = ‘44444ccccccccccabbbb’;

trong đó *x* chứa dữ liệu của dòng thứ 4, *y* chứa dữ liệu của dòng thứ 5 trong tệp CNMAX.INP.

Từ điểm *i = 8* dịch chuyển về bên trái ta thu được điểm *c1 = 6*; dịch chuyển về bên phải ta thu được điểm *c2 = 15*. Điều kiện để dịch chuyển là:

(y[c1-1] = y[i]) and (h[c1-1] >= h[i]) ; *nếu qua trái và*
(y[c2+1] = y[i]) and (h[c2+1] >= h[i]); *nếu qua phải*.

Hai thao tác trên được đặt trong hàm tính diện tích của hình chữ nhật ABCD lớn nhất chứa điểm *y[i]*. Hàm cho ra 3 giá trị:

c1: điểm trái nhất hay là toạ độ cột của đỉnh D,
c2: điểm phải nhất hay là toạ độ cột của đỉnh C, và
Diện tích của hình.

```
(* Dien tich cua hinh chua diem y[i] *)
function DienTich(i: byte; var c1, c2: byte): longint;
```

```
begin
  { Qua phai }
  c1 := i;
  while (y[c1-1] = y[i]) and (h[c1-1] >= h[i]) do dec(c1);
  { Qua trai }
  c2 := i;
  while (y[c2+1] = y[i]) and (h[c2+1] >= h[i]) do inc(c2);
  DienTich := h[i]*(c2 + 1 - c1);
end;
```

Phần xử lý chính được đặt trong thủ tục Run.

Mảng *h[1..m]* lúc đầu được khởi trị toàn 0. Sau mỗi lần đọc dòng *y* ta chỉnh lại độ cao *h* theo tiêu chuẩn sau.

Tại điểm *i* đang xét trên dòng *y*, nếu *y[i] = x[i]* độ cao *h[i]* được tăng thêm 1. Ngược lại, nếu *y[i] x[i]* ta đặt lại độ cao *h[i] = 1*.

```
{ Chinh do cao }
  for i := 1 to m do
    if y[i] = x[i] then inc(h[i]) else h[i] := 1;
```

Một vài chú giải

1. Chúng ta sử dụng phần tử *h[0] = 0* và *h[m+1] = 0* để chặn vòng lặp, cụ thể là để các điêù kiện *(h[c1-1] >= h[i])* và *(h[c2+1] >= h[i])* trở thành *False* ở cuối vòng lặp.
2. Một con đếm dòng *d* kiểu longint cho biết ta đang xử lý dòng nào của tệp.
3. Đều kiện "*trong tệp không chứa dấu cách*" được thể hiện qua việc kiểm tra dòng đã đọc *y*. Nếu phần tử đầu tiên của *y* là dấu cách ta kết thúc chương trình.
4. Dòng *x* lúc đầu đọc khởi trị toàn dấu cách.
5. Mỗi khi xử lý xong dòng *y* ta cần sao chép giá trị của *y* cho *x* để lưu giữ cho bước tiếp theo. Khi đó *x* sẽ trở thành *dòng trớc*.
6. Mỗi khi tìm được một hình chữ nhật, ta so sánh diện tích của hình với diện tích lớn nhất hiện có (*dtmax*) để luôn luôn đảm bảo rằng *dtmax* chính là diện tích lớn nhất trong vùng đã khảo sát.

Thủ tục Run được thiết kế theo sơ đồ sau:

1. Mở tệp dữ liệu CNMAX.INP;
2. Đọc giá trị chiều dài mỗi dòng vào biến m;
3. Khởi trị:
   Con đếm dòng d := 0;
   Dòng trước x toàn dấu cách;

Mảng chiều cao h toàn 0;
dtmax := 0;

4. Lặp cho đến khi hết tệp CNMAX.INP:
   Đọc dòng y;
   Thêm dấu cách vào cuối dòng y;
   Kiểm tra điều kiện kết thúc: (y[1] = BL) ? { BL = dấu cách }
   Tăng con đếm dòng d;
   Chỉnh độ cao h[1..m];
   Xử lý mỗi ký tự y[i] của dòng y; i = 1..m
   Tìm diện tích dt của hình chữ nhật lớn nhất chứa phần tử y[i];
   Nếu dt > dtmax: chỉnh lại các giá trị
      Diện tích max: dtmax := dt;
      Toạ độ đỉnh A(Axmax , Aymax):
         Axmax := d - h[i] + 1;
         Aymax := cot1;
      Toạ độ đỉnh C(Cxmax , Cymax):
         Cxmax := d;
         Cymax := cot2;
   Sao chép dòng y sang x: x := y;
5. Đóng tệp CNMAX.INP
6. Thông báo kết quả.

Độ phức tạp tính toán

Giả sử tệp CNMAX.INP chứa *n* dòng, mỗi dòng chứa *m* ký tự. Khi xử lý một dòng, tại mỗi ký tự thứ *i* trên dòng đó ta dịch chuyển qua trái và qua phải, tức là ta phải thực hiện tối đa *m* phép duyệt. Vậy, với mỗi dòng gồm *m* ký tự ta phải thực hiện *m-2* phép duyệt. Tổng cộng, với *n* dòng ta thực hiện tối đa $t = n \cdot m-2$ phép duyệt.

Bài toán 37 : ma trận Idently

Đề bài :

Cho một ma trận nhị phân A kích thước NxN . Người ta gọi ma trận con B của A là một ma trận nhận được bằng cách xoá hàng và cột của ma trận A .

Một ma trận B đợc gọi là ma trận đồng nhất (Idently) A nếu B là ma trận được từ A mọi phần tử của ma trận B đều bằng 0 trừ các phần tử trên đường chéo chính của ma trận B bằng 1 .

với ma trận A đưa vào , hãy tìm ma trận đồng nhất của A có kích thước lớn nhất .

Dữ liệu :

Dòng đầu tiên là số N (N<=30)

N dòng kế tiếp là ma trận A

kết quả :

Dòng đầu tiên là số m ,kích thước của ma trận đồng nhất A

Dòng kế tiếp là m số thể hiện cho những hàng còn lại của ma trận đồng nhất

dòng cuối cùng là m số thể hiện cho những cột còn lại của ma trận đồng nhất

ví dụ :

matrix.inp	matrix.out
5 1 1 0 0 0 1 0 1 0 1 0 0 0 1 0 1 0 0 0 1 0 1 0 1 0	3 1 2 3 2 3

Hướng dẫn :

Chúng ta sẽ duyệt : Một hàng , hoặc môt cột nào đó có thể có mặt trong ma trận đọc không . Trong quá trình duyệt thì , với mỗi lần ghi nhận hang , cột nào đó vào ma trận cũ thì ta sẽ cập nhật và sử dụng các hàm kiểm tra đảm bảo tính chất của ma trận : đường chéo chính phải có số 1 , còn lại là số 0 (phải luôn luôn update mỗi lần cập nhật) .

Bài toán 38 : Xâu ba

Đề bài :

Cho một số nguyên dương N<=1000 . Hãy tìm Xâu S độ dài N chỉ gồm các ký tự 0 hay 1 sao cho S không có xâu con nào xuất hiện liên tiếp 3 lần trong nó.

Dữ liệu : vào từ File Xauba.Inp :

Một dòng duy nhất ghi số N

Kết quả : ghi ra file Xauba.Out :

Một dòng duy nhất ghi xâu S

Hướng dẫn :

Chúng ta sẽ duyệt bài toán này giống như bài toán “ABC” nhưng ở bài toán này thì chỉ có hai giá trị nên chúng ta dễ giải quyết hơn . Mặt khác đây chỉ là đệ quy không có nhớ (tức là không có điều kiện Max hoặc min nào cả) .

Bài toán 39 : Lát nền

Đề bài :

Có một mảnh sân hình chữ nhật MxN (M,N<=30). Mảnh sân này cần được lát bằng gạch hoa. Thế nhng có rất nhiều viên gạch hình vuông, chữ nhật kích cỡ khác nhau . Mỗi viên gạch có một giá nhất định cho từng loại . Chính vì thế người chủ sân muốn lát sân này . Bạn hãy giúp ông đó lát lại sân đó mà tốn ít tiền nhất

Dữ liệu : Vào từ file LatNen.Inp :

Dòng đầu tiên ba số M ,N , K : M ,N là kích thước sân cần lát . K là số loại gạch

K dòng sau , mỗi dòng hai số ghi kích thước của mỗi loại gạch và giá tiền của gạch đó

Kết quả : Ghi ra file LatNen.Out :

Dòng đầu tiên là số viên cần dùng P và số tiền cần mất

P dòng sau mỗi dòng ghi 3 sô : I,J,T : I,J là toạ độ góc trái dưới của viên gạch thứ T đặt vào

Ví Dụ :

LATNEN.INP	LATNEN.OUT
5 5 5 1 2 1 3 3 1 4 3 1 2 2 1 5 3 1	4 4 1 1 5 1

Hướng dẫn :

Chúng ta sẽ duyệt cách cách để có thể có của bảng . Với mỗi lần kết nạp một viên gạch nào đó vào thì chúng ta sẽ làm các công việc đánh dấu bảng . Như thế thì khi các viên sau để vào thì nếu viên nào có gạch chồng lên các viên đã lát thì sẽ loại. Chúng ta nên sử dụng cận : Ta sẽ tính tiền cho mỗi ô vuông cho khi lát một viên gạch nào đó . Như thế khi chúng ta lát đến một đoạn nào đó mà giá tiền các viên đã lát mà cộng thêm với số diện tích chưa được lát mà nhân với số tiền nhỏ nhất khi lát một ô vuông mà lớn hơn Best thì chúng ta sẽ loại . Đại thể hàm đó như sau :

Nếu Tiền hiện tại + Diện tích chưa lát *Tiền (TBnhỏ nhất) >Best thì loại (không duyệt đến)

Bài toán 40 : Mass

Đề bài :

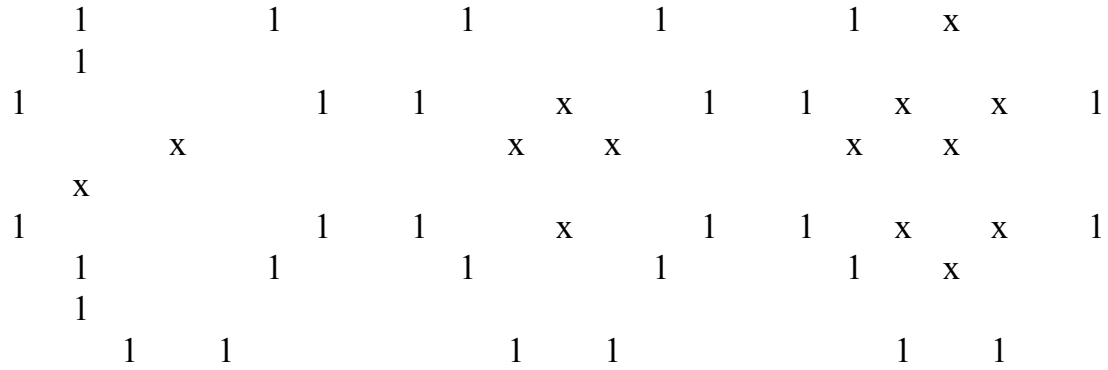
Một người được cử lên sao hảo thám thính. Để dễ dàng người ta coi bề mặt sao hoả như một hình chữ nhật kích thước m x n . Mỗi một ô có thể chứa núi lửa gọi là ô chết hoặc ô trống (ô an toàn) . Tại mỗi bước người đó luôn luôn bước sang một ô chung cạnh với ô đang đứng nếu ô đó không phải là ô chứa núi lửa .

Ví Dụ :

Tại thời điểm 0, nhà thám hiểm được đặt tại vị trí (4,4). Tại thời điểm 1 nhà thám hiểm có thể ở 4 vị trí (4,3) , (4,5) ,(3,4) , (5,4). Tại thời điểm 2 nhà thám hiểm có thể tại các vị trí (2,4) , (3,3) , (3,5) , (4,2) ,(4,4) ,(4,6) ,(5,3),(5,5) ,và (6,4)

hình :

1 1 1 1 1 1



Thời điểm 0 Thời điểm 1 Thời điểm 2

Sau một thời gian chuyển động thì nhà thám hiểm biến mất tại ô (x,y) (dự đoán anh ta có thể gặp một nàng Eva trên sao hoả) và muốn cử một đội tầu đi tìm . Mỗi tàu trong đội tầu sẽ hạ cánh xuống một ô an toàn và có khả năng quan sát trong vòng bán kính là r (tính theo toạ độ của hình chữ nhật) . Tức là nếu tàu ở ô (x,y) thì có thể quan sát mọi ô (i,j) mà | i-x| +|j-y|<=r .

Yêu cầu : hãy tìm các vị trí để các con tàu hạ cánh để tìm ra nhà thám hiểm trong một khoảng thời gian t2 và số tàu cần dùng là ít nhất có thể được

Dữ liệu : Vào từ file Mass.Inp :

Dòng đầu tiên là các số m , n , r , (x,y) ,t1 và t2 ,thể hiện cho kích thước của bảng , khả năng quan sát của tàu tìm kiếm , vị trí nhà thám hiểm biến mất , thời gian để các tầu hạ cánh xuống sao hoả và thời gian để tìm ra nhà thám hiểm . (m ,n <=20) .

Dòng kế tiếp là bản đồ của sao hoả thể hiện bằng một bảng với ô số 0 là ô trống , ô số 1 là ô chứa núi lửa

Kết quả : Ghi ra file Mass.Out :

Dòng đầu tiên là số P , số tầu tìm kiếm cần dùng

P dòng kế tiếp mỗi dòng là một cặp số thể hiện cho toạ độ của từng tầu tìm kiếm .

Ví Dụ :

MASS.INP MASS.OUT

7 7 2 4 4 3 2 0 0 1 0 1 0 0 0 1 0 0 0 1 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 1 0 0 0 1 0 0 0 1 0 1 0 0 1 4 4

Hướng dẫn :

Chúng ta sẽ tìm các vị trí có thể có ngời thám hiểm sau t thời gian . Sau đó duyệt các khả năng có thê có đặt các con tàu. Cách duyệt hoàn toàn giống bài toán phân bố các vị trí .

Bài toán 41 : Rotation

Đề Bài :

Cho một số X(số chữ số của X <=14). Chúng ta gọi số đó là một số xoay khi chúng ta xoay số X 180 thì ta vẫn được số X. Ví Dụ : 11,69,96 là những số xoay . Yêu cầu đặt ra là : Khi cho một số K , Hãy tìm xem với những số có K Chữ số thì có bao nhiêu số K và đó là những số nào ?.

Dữ liệu : Vào từ File Input: Rotation.Inp Chỉ ghi duy nhất một số nguyên dương K(1<=K<=14) .

Kết quả : Xuất ra File rotation.Out : chỉ một dòng ghi tất cả các số thoả mãn, chúng được ghi cách nhau bởi một dấu cách.

Ví Dụ :

ROTATION.INP	ROTATION.OUT
2	11 69 88 96

Hướng dẫn :

Chúng ta có thể thấy ngay trong hệ thập phân thì các số từ 0 đến 9 thì có những số :0,1,6,8,9 là những số khi xoay 180 thì tạo ra một số có nghĩa.Mặt khác khi xoay nh thế để toạ thành một số xoay thì số đó phải có một trục đối xứng (tức là các chữ số từ 1K Div 2 thì là kết quả của phép xoay của các số từ K Div 2+1 K). Chính vì thế chúng ta có thể duyệt các chữ số trong K div 2 chữ số để tìm ra một số mà khi xoay 180 thì vẫn tạo ra số đó. Công việc giải quyết vấn đề trên có thể được mô tả cụ thể nh sau :

Đầu tiên chúng ta dùng hàm xoay một số (09) tạo một số mới có nghĩa : 00 , 11, 69, 96, 88.

Xây dựng hàm Try(I:Integer) để duyệt các khả năng của số thứ I .

Nếu K Chẵn thì chúng ta chỉ xét các chữ số từ 1 đến K Div 2 rồi sau đó dùng hàm xoay để xác định các số ngược lại .

Nếu K Lẻ thì số thứ (K+1) div 2 chỉ có thể là các số 0,1,8. còn các số từ 1K div 2 thì ta dùng thủ tục try để tìm các số đó rồi sau đó dùng hàm ngược để xác định các giá trị ngược .

Tôi có thể viết rõ các thủ tục trên như sau :

```
Function  Xoay(I:Byte):Byte;
Begin
        Case I Of
        0: Xoay:=0;
        1: Xoay:=1;
```

```
        6: Xoay:=9;
        8: Xoay:=8;
        9: Xoay:=6;
        End;
End;

Procedure Try(I:Integer);
Var J:integer;
Begin
        For J:=1 To 5 Do
        Begin
                If (I=0)And(J=1) Then Break;
                B[I]:=A[J]To K Do
                If I=K Div 2 Then Xuat
                Else Try(I+1);
        End;
End;
```

Mảng A[1..5]=(0,1,6,8,9)
Mảng C[1..3]=(0,1,8);
Trong đó thủ tục Procedure Xuat Nh sau :

```
Procedure Xuat;
Var I,J:Integer;
Begin
        For I:=K Div 2+1 To K Do
        B[I]:=Xoay(B[K -I+1]);
        If Odd(K) Then
        Inc(Dem,3);
        For J:=1 To 3 Do
        Begin
                B[(K+1) Div 2]:=C[J];
                For I:=1 To K Do Write(F,B[I]);
                Write(F,' ');
        End;
End;
```

Bài toán 42 : Ghép tranh

Đề bài :

Một bức tranh lớn hình chữ nhật được cắt thành N hình vuông nhỏ , đánh số từ 0 đêbs N-1 . Bốn cạnh hình vuông được đánh số từ 0 đến 4 ngược chiều kim đồng hồ , bắt đầu từ cạnh dưới (hình 1)

Trước khi cắt nhỏ tranh , người ta đã đánh số các hình vuông và đánh dấu 1 số cho biết canh nào của một hình vuông khớp với cạnh nào của hình vuông khác . Thông tin đủ để khôi phục lại tranh ban đầu . Sau khi cắt nhỏ tranh ,người ta mới bắt đầu đánh số các cạnh của chúng . Các hình vuông có thể đã bị xoay , nhưng không bị lật úp

Yêu cầu : Chỉ ra cách ghép để nhận được tranh ban đầu . ở kết quả ,vị trí các hình vuông được xác định theo toạ độ đánh số từ 0 trở đi , từ trái sang phải và từ trên xuống dưới .

Dữ liệu : Vào từ file văn bản JIGSAW.INP :

Dòng đầu tiên chứa 2 số nguyên N,K , trong đó N-số hình vuông ,0<N<500 ,K-số dòng thông tin ghép nối .

K dòng sau : mỗi dòng chứa 4 số nguyên không âm P,Q,U,V , cho biết cạnh Q của hình vuông P khớp với cạnh V của hình U .

Ví Dụ : Dòng thông tin vào là 7 2 9 1 thì các cách ghép thể hiện là :

Kết quả : Đa ra file JIGSAW.OUT :

N dòng , mỗi dòng 4 số nguyên I,J,S,L cho biết là đặt tranh S vào vị trí (I,J) có hướng tranh xuống dưới là L .

Ví Dụ :

JIGSAW.INP JIGSAW.OUT

12 13 0 0 5 0 0 2 6 2 1 2 11 0 1 3 9 2 2 0 5 1 2 1 4 2 3 0 10 3 3 3 8 0 4 0

7 1 4 1 11 1 6 1 7 2 8 2 9 3 10 0 11 2 0 0 3 3 0 1

Hướng dẫn :

Đầu tiên ta tìm cách đặt một hình nào đó vào vị trí cần đặt , sau đó sử dụng thuật toán loang sẽ biết được các vị trí của các bức tranh mà liên kết với nó . Cứ như thế các vị trí còn lại ta cứ làm như vậy cho khi phát hiện được bức tranh .

Bài toán 43 : Cờ nhảy

Đề bài :

Hãy thiết kế phần mềm trò chơi PCgame giới thiệu trò chơi cờ nhảy : trên lưới tam giác đều có 15 nút , đánh số từ 1 đến 15 (xem hình vẽ) . Người ta đặt 14 quân cờ , mỗi quân trên một nút và để một nút trống . Một quân cờ có thể nhảy qua hoặc một số quân liên tiếp nhau tới nút trống trên đường thẳng . Các quân bị nhảy qua sẽ bị loại khỏi bàng cờ và các nút có các quân bị loại cũng nh nút quân vừa nhảy trở thành nút trống .

Các số được đánh số từ trên xuống dưới , từ trái sang phải theo số hiệu tăng dần .

ở hình trên , quân 12 hoặc 14 có thể nhảy tới nút 5 . Nếu quân 12 nhảy tới bút 5 thì quân 9 sẽ bị loại và các nút trống sẽ là 9 và 12 .

Yêu cầu : Hãy lập trình tìm số nước đi ít nhất để sau đó trên bàn cờ chỉ còn lại một quân . Nếu điều này không thể làm được thì thông báo IMPOSSIBLE .

Dữ liệu : Vào từ file Peg.Inp :

Dòng đầu tiên là số bộ dữ liệu Test T (nguyên dơng)

Mỗi dòng sau tương ứng với một test , chứa một số nguyên cho biết nút trống ban đầu

Kết quả : Ghi vào file Peg.Out : mỗi bộ test tơng ứng với 2 dòng , dòng đầu chứa số bước nhảy tối thiểu , dòng thứ hai xác định các bước nhảy cần thực hiện , mỗi bước nhảy được đặc trưng bởi hai số nguyên : số đầu chỉ nút có quân sẽ nhảy , số sau -nút tới (rỗng)

Ví Dụ :

PEG.INP	PEG.OUT
1 5	8 12 5 10 8 1 10 11
1 14 12 12 3 1 6 15 3	

Hướng dẫn :

Duyệt các vị trí cuối cùng có thể có của bảng . Chúng ta lưu trạng thái của bảng bằng cách : Ta coi mỗi bảng như vậy gồm 15 số . Thì vị trí của các quân cờ không ảnh hưởng đến cách nhảy để thu hết các quân cờ . Chính vì thế 15 vị trí chỉ cần lưu 0 hoặc 1 để nói rằng vị trí đó có quân cờ hay trống mà thôi. Ta thấy một số ở dạng Byte có 8bit , thì như vậy với một bảng là15 số thì ta chỉ cần lưu dưới dạng một số nằm trong dạng word với 16bit mà thôi. Bằng các thủ tục bật bit và tắt bit ta có thể hạn chế được bộ nhớ rất nhiều .

Bài toán 44 : Dàn đèn màu

Đề bài :

Cho một lưới toạ độ nguyên, hoành độ từ 0 đến M, tung độ từ 0 đến N (M, N 200). Trên K nút cho trước của lưới, mỗi nút cần đặt một đèn màu sao cho hai đèn ở hai nút có cùng hoành độ hoặc có cùng tung độ phải có

màu khác nhau. Hãy tìm cách bố trí dàn 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 1.

Dữ liệu vào: file văn bản BL1.INP:

Dòng đầu tiên ghi ba 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, ... K).

Kết quả : ghi ra file văn bản BL1.OUT:

Dòng đầu ghi số lợng mầu cần sử dụng P,

Dòng thứ i trong số K dòng tiếp theo ghi màu của đèn ở nút thứ i (i=1, 2,... , K).

Ví dụ : các file dữ liệu - kết quả có thể

BL1.INP	BL1.OUT
4 5 13	4
1 1	1
1 2	2
1 5	3
3 1	2
4 1	3
3 2	1
2 3	1
3 3	3
4 3	2
2 4	3
4 4	1
2 5	2
4 5	4

Các số liệu trên cùng một dòng ghi cách nhau ít nhất một dấu trắng.

Hướng dẫn :

Dùng thuật toán tô màu các đỉnh của đồ thị . (Nhưng trong trường hợp này , chúng ta sẽ tham lam một cách tô hiệu quả nhất) .

Bài toán 45 : Các hình tròn lồng nhau

Đề bài :

Cho dãy gồm N hình tròn khác nhau trên mặt phẳng đánh số từ 1 đến N (N<=100) . Hình tròng 2 gọi là nằm trong hình tròn một nếu mọi điểm thuộc 2 đều là điểm thuộc 1 .

Yêu cầu : Hãy phân lớp các hình tròn đã cho để không có hai hình nào thuộc cùng một lớp lại nằm trong nhau sao cho số lớp là ít nhất .
Dữ liệu : Vào từ file văn bản BL3.Inp :
Dòng đầu tiên là số N
Dòng thứ i trong số N dòng tiếp theo ghi thông tin của hình tròn i gồm 3 số thực (x,y,r) là toạ độ tâm và bán kính
Kết quả : Ghi ra file BL3.OUT :
Dòng đầu tiên ghi M là số lớp tìm đợc
Mỗi dòng trong số M dòng tiếp theo ghi thông tin của một lớp bao gồm số hình tròn thuộc lớp , tiếp theo là số hiệu của các hình tròn thuộc lớp .
Ví Dụ :

BL3.INP	BL3.OUT
6 2 2 1 4 1 2 6 4 2 6 4 1 8 5 1 7 4 3	3 2 4 5 3 1

Hướng dẫn :
Thuật toán duyệt -phân lớp của một đồ thị . Có thể tham lam đúng (giống như bài toán tô màu chẳng hạn) để tìm ra một kết quả chỉ sai khác với độ chính xác là 1 là cùng . Sau đó làm một phép duyệt có một cận trên đã biết thì tốc độ chương trình rất nhanh .

Bài toán 46 : Tập hợp

Đề bài :
Cho trước là tập hợp gồm một số chữ cái tiếng Anh in hoa. Ta xét tập hợp bao gồm mọi tập con trong (mỗi phần tử thuộc là một tạp nào đó các chữ in hoa thuộc): chúng ta không xét tập rỗng và như vậy, mọi phần tử trong chứa ít nhất với mỗi phần tử có trong XY thì phần tử đó hoặc thuộc X, hoặc thuọc Y. hoặc thuộc cả hai tập con X và Y (XY chính là "hợp của X và Y")
Trong cùng ta xét quan hệ "suy ra được" (Ký hiệu là) theo định nghĩa như dưới đây (X, Y, Z là các phần tử thuộc hay cũng chính là các tập con của)
(1) Cho sẵn một số bộ X, Y nào đó mà giả thiết X suy ra được Y (X Y) đã có. Ký hiệu là tập hợp các bộ suy ra được X Y cho trước đó. Từ tập các bộ suy ra được có trước này và 3 quy tắc dới đây cho phép nhận được các bộ suy ra được mới.

(2) nếu X chứa Y thì X suy ra dợc Y (X Y)

Nếu X Y thì X Y (dấu dùng với nghĩa "bao hàm", "chứa")

(3) Nếu đã có X suy ra đợc Y và X suy ra được Z thì cũng kết luận được X suy ra được tập hợp YZ (là "hợp" của Y và Z)

Nếu X Y và X Z thì có X YZ.

(Tính mở rộng)

(4) Nếu X suy ra được Y và Y suy ra được Z thì X suy ra được Z

Nếu X Y và Y Z thì X Z

(Tính bắc cầu)

Giải hai bài toán sau đây:

Câu 1: Cho trước một tập con W các phàn tử thuộc . Hãy tìm tập con lớn nhất W* để cho W W*

Câu 2: Tìm tập con W có ít phần tử nhất để cho W

Dữ liệu vào có trong File văn bản INP. B3 có cấu trúc nh sau:

- Dòng đầu tiên chứa các chữ cái thuộc : các chữ cái in hoa được viết ngay từ đầu dòng, viết liên tiếp nhau.
- Dòng thứ 2 có một số nguyên dương N (N <=20) chỉ số lượng bộ suy ra được cho trước của .
- Trong N dòng tiếp theo, mỗi dòng cho biết từng bộ X, Y mà suy ra đợc cho trước (X Y) và X hay Y là một từ gồm các chữ cái in hoa; hai từ này cách nhau chỉ một dấu cách.
- Dòng cuối cùng chứa một xâu các chữ cái in hoa cho biết các chữ cái thuộc vào tập W cho trớc mà cần phải tìm W* (nh câu 1)

Kết quả ra cho vào File văn bản OUT.B3 gồm 2 dòng (mỗi dòng được viết giống dòng cuối cùng của File dữ liệu).

- Dòng đầu tiên chứa W* cần tìm theo câu 1
- Dòng thứ 2 chứa V là tập có ít phần tử nhất để cho V theo câu 2.

Hướng dẫn :

Câu1 :

Theo bài ra, tập có hữu hạn phần tử . Thiết lập dãy tập W1,W2,...Wk theo cách sau đây :

W0=W

Wi=Wi-1 (UY)

X Y

X Wi-K

K là chỉ số đầu tiên mà Wi=Wi+1(WK-1 Wk=Wk+1)

Do giả thiết tập là hữu hạn nên số K tồn tại .

Ta cần chứng minh W*=Wk .

Rõ ràng WWk chỉ cần chứng minh Wk là lớn nhất .Giả sử phản chứng W*-Wk có nghĩa là tồn tại x W* mà x không thuộc Wk . Do W là tập con của Wk nên x không thuộc W . Nh vậy , trong quá trình suy từ W để có được W* phải có một lúc nào đó để WW'W'' mà từ lúc đó trở đi về

phải có x (W'cha có x) .Để làm được như vậy ,cần tồn tại một quy tắc XYthuộc sao cho X là tập con của W'và Y có chứa x . Tập W' và W'' là những tập con của Wi trên đáy .

Câu 2 :

Thuật toán trên đây ,cho phép từ tập V bất kỳ luôn tìm được tập V* tương ứng .Qua phân tích thuật toán dới đây :

Đặt L= UX R =UY
XY XY

Đặt V0=\ LR
V1=L\R

Tập chắc chắn chứa hai tập V0 và V1 . Thiết lập tập V2=(V0V1)* .
Còn lại tập V3 =(L R)\V2 và duyệt theo các phần tử của tập này từ tập gốc là (V0V1) song lại xuất phát từ V2 cho nhanh .

Bài toán 47 : Chặt Cây

Đề Bài :

Một gia đình nọ có một khu vườn cây và muốn rào khu vờn đó lại . Tuy nhiên , để rào được toàn bộ cây đó thì chúng ta phải chặt một số cây trong vờn để lấy gỗ rào các cây còn lại . Mỗi cây gỗ sau khi chặt cho một lượng gỗ đủ để rào một đoạn Li .

Hãy tìm cách chặt ít cây nhất để có được một hàng rào bao được tất cả các cây của khu vờn (hàng rào phải là một đa giác bao đợc tất cả các cây còn lại) .

Dữ liệu Vào từ file văn bản : CAY.INP nh sau :
Dòng đầu tiên ghi N (N<=50).
N dòng tiếp , mỗi dòng ghi 3 số (x,y,L) là biểu diễn toạ độ của cây ấy và lượng gỗ thu được khi chặt nó .

Kết Quả Ghi ra file CAY.OUT như sau :
Dòng đầu tiên ghi số cây tối thiểu phải chặt
Dòng kế tiếp là các cây cần chặt .

Bài toán 48 : In đĩa CD

Đề bài :

Trong một chuyến khảo sát , một đoàn khảo sát đã thu thập được các dữ liệu quí giá đợc lưu dưới dạng các file . Để thuận tiện , đoàn quyết định thu đĩa CD các file dữ liệu .

Một file dữ liệu không đợc chia nhỏ mà phải lưu nguyên trong một đĩa , một đĩa có thể lưu nhiều file miễn là tổng dung lượng <= 640M . Hãy giúp đoàn xắp xếp các file in vào các đĩa sao cho cần dùng ít đĩa CD nhất.

Dữ liệu : File BaoTon.Inp :
Dòng đầu tiên ghi N là số file. (n <= 50)

N dòng tiếp theo , dòng thứ i ghi 1 số là dung lượng của một file dữ liệu thú i tính theo Mega Byte.

Dung lượng các file đều <= 640M.

Kết quả : Ghi ra file Baoton.Out

M là số ít nhát đĩa.

M dòng tiếp theo ,dòng thú i ghi danh sách các file in vào đĩa thứ i.

Hướng Dẫn :

Đây chính là bài toán BIN PACKING hiện chưa có thuật toán đa thức cho bài toán này . Chúng ta chỉ có thẻ duyệt tham .