Informatika

I. 81. Egy autóbuszjegyen az *n*n*-es négyzethálóban összesen ***k*** lyukasztás lehet. Ha a buszjegyet fordítva helyezzük a lyukasztóba, akkor a jegy tükörképét kapjuk. (Csak egyféleképpen lehet fordítva betenni a jegyet, mivel be van fogva egy jegytömbbe).

Adott *n*-re és *k*-ra generáljuk az összes lehetséges lyukasztást úgy, hogy egyetlen lyukasztásnak se legyen meg a tükörképe az addig generáltak között.

Bemeneti adatok.

n ($2 \le n \le 9$) és k ($1 \le k \le 4$), melyeket a billentyűzetről visszük be.

Eredmény.

Egy szövegállományba, amelynek nevét kérjük be a billentyűzetről, egy-egy sorba írjunk be egy lyukasztást a következőképpen:

$i_1j_1i_2j_2...i_kj_k$

ahol i_pj_p (p=1,2,...,k) egy adott lyuk koordinátája a jegyen (i_p a sor, j_p az oszlop száma). A lyukasztások az állományban lexikografikus sorrendben szerepeljenek.

Példa:

n=3, k=2 esetében a kimeneti állománynak a következő adatokat kell tartalmaznia:

1112 1113 1121 1122 1123 1131 1132 1133 1221 1222 1231 1232 2122 2123 2131 2132 2133 2231 2232 3132 3133

A programnak 1 percen belül kell eredményt szolgáltatnia.

Kása Zoltán (Olimpiai válogató versenyfeladat, Kolozsvár, 1996)

Megoldás:

Alapötlet, hogy a lyukasztásokat (konfigurációkat) lexikografikus sorrendben generáljuk, s csak akkor írjuk be a kimeneti állományba, ha a tükörképe "nagyobb" (azaz, még nincs beírva). Az *n* x *n*-es mátrixot linearizálva tekintjük. Egy adott lyuk koordinátái a programban (s1, o1), (s2, o2) stb.

```
program buszjegy;
type vektor = array[ 1..10] of byte;
var m,i1,i2,j1,j2,j3 : integer;
  s1,s2,s3,s4,s5,o1,o2,o3,o4,o5  : integer;
  f  : text;
  x,y : string[ 2] ;
  nr : longint;
  n,k : byte;
  b,c : vektor;
{ ------------------------------------------------------------}
  procedure tukor (b:vektor; var c:vektor) ; { egy konfiguráció}
  var t,j,i : byte;                                  { tükörképe }
  begin
   for i:=1 to k do c [ i* 2-1] :=b [ i* 2-1] ;
   for i:=1 to k do c [ 2* i] :=n+1-b [ 2* i] ;
    for i:=1 to k do
     for j:=i+1 to k do
```

```
    if c [ 2*i-1] =c [ 2*j-1] then
     if c [ 2*i] > c [ 2*j] then
      begin t:=c [ 2*i] ; c [ 2*i] :=c [ 2*j] ; c [ 2*j] :=t; end;
  end;
{ ------------------------------------------------------}
  function kisebb (b,c:vektor) : boolean; { konfigurációk }
  var i : byte;                           { összehasonlítása}
  begin
    i := 1;
    while (i <= 2*k) and (b [ i] =c [ i] ) do i:=i+1;
    if (i > 2*k) or (b [ i] < c [ i] ) then kisebb := true
                                       else kisebb := false;
  end;
{ ------------------------------------------------------}
BEGIN                                                 { fôprogram}
 write (' n=' ); readln (n);
 write (' k=' ); readln (k);
 str (n,x); str (k,y);
 assign(f,' out' +x+' .' +y); rewrite (f); nr := 0;
 m := n*n;

 case k of
 1: for i1:= 1 to m do
    begin
      s1:= (i1-1) div n+1 ; o1:= (i1-1) mod n+1;
      b [ 1] := s1; b [ 2] := o1;
      tukor (b,c);
      if kisebb (b,c) then
      begin
        writeln (f,s1,o1); nr := nr+1;
      end;
    end;
 2: for i1 := 1 to m do
     for j1 := i1+1 to m do
      begin
        s1:= (i1-1) div n+1 ; o1:= (i1-1) mod n+1;
        s2:= (j1-1) div n+1 ; o2:= (j1-1) mod n+1;
        b [ 1] := s1; b [ 2] := o1; b [ 3] := s2; b [ 4] := o2;
        tukor (b,c);
        if kisebb (b,c) then
          begin
            writeln (f,s1,o1,s2,o2); nr := nr+1;
          end;
      end;
 3:  for i1 := 1 to m do
     for j1 := i1+1 to m do
     for j2 := j1+1 to m do
      begin
        s1:= (i1-1) div n+1 ; o1:= (i1-1) mod n+1;
        s2:= (j1-1) div n+1 ; o2:= (j1-1) mod n+1;
        s3:= (j2-1) div n+1 ; o3:= (j2-1) mod n+1;
        b [ 1] := s1; b [ 2] := o1; b [ 3] := s2; b [ 4] := o2;
        b [ 5] := s3; b [ 6] := o3;
        tukor (b,c);
        if kisebb (b,c) then
        begin
         writeln (f,s1,o1,s2,o2,s3,o3); nr := nr+1;
        end
      end;
```

```
4:  for i1 := 1 to m do
    for j1 := i1+1 to m do
     for i2:= j1+1 to m do
      for j2 := i2+1 to m do
      begin
        s1:=(i1-1) div n+1 ; o1:= (i1-1) mod n+1;
        s2:=(j1-1) div n+1 ; o2:= (j1-1) mod n+1;
        s3:=(i2-1) div n+1 ; o3:= (i2-1) mod n+1;
        s4:=(j2-1) div n+1 ; o4:= (j2-1) mod n+1;
        b [ 1] := s1; b [ 2] :=o1; b [ 3] := s2; b [ 4] :=o2;
        b [ 5] := s3; b [ 6] :=o3; b [ 7] := s4; b [ 8] :=o4;
        tukor (b,c);
        if kisebb (b,c) then
        begin
         nr := nr+1;
         writeln (f,s1,o1,s2,o2,s3,o3,s4,o4)
        end
       end;


 5:  for i1 := 1 to m do
     for j1 := i1+1 to m do
      for i2:= j1+1 to m do
       for j2 := i2+1 to m do
        for j3 := j2+1 to m do
       begin
         s1:=(i1-1) div n+1 ; o1:= (i1-1) mod n+1;
         s2:=(j1-1) div n+1 ; o2:= (j1-1) mod n+1;
         s3:=(i2-1) div n+1 ; o3:= (i2-1) mod n+1;
         s4:=(j2-1) div n+1 ; o4:= (j2-1) mod n+1;
         s5:=(j3-1) div n+1 ; o5:= (j3-1) mod n+1;
         b [ 1] := s1; b [ 2] :=o1; b [ 3] := s2; b [ 4] :=o2;
         b [ 5] := s3; b [ 6] :=o3; b [ 7] := s4; b [ 8] :=o4;
         b [ 9] := s5; b [ 10] :=o5;
         tukor (b,c);
         if kisebb (b,c) then
         begin
          nr := nr+1;
          writeln (f,s1,o1,s2,o2,s3,o3,s4,o4,s5,o5)
         end
        end;
 end{ case} ;
 close( f);
 writeln (' Konfigurációk száma: ', nr);
end.
-------------------------------------------------------
-------------------------------------------------------
```

Tesztprogram

A program úgy teszteli az eredményállomány helyességét, hogy véletlenszerűen generál megadott számú konfigurációt és megvizsgálja, hogy ezek benne vannak az állományban van sem, és ha igen akkor csupán csak egyszer vagy kétszer.

```
program teszt; {----------------------------------------+
       | Véletlenszerűen generált konfigurációkat vizsgál |
       +-------------------------------------------------}
var a  : array [ 1..9] of byte;
  b,c : array [ 1..18] of byte;
  o  : array [ 1..20] of char;
  lo,lob,loc                          : string;
  hossz, bal, jobb, kozep             : longint;
  t,kul,m,k,n,tsz,igazc,igazb,i,j     : integer;
  f1 : file of char;
  f2 : text;

BEGIN
 write('Állománynév:');readln(lo);
 write('n:');readln(n);
 write('k:');readln(k);
 write('Tesztek száma:');readln(tsz);
 assign(f1,lo);
 assign(f2,'x'+lo);
 rewrite(f2);
 randomize;
 for m:=1 to tsz do
 begin
   for i:=1 to k do
     begin
      repeat
       kul:=1;
       a[i]:=random(n*n)+1;
       for j:=1 to i-1 do
         if a[i]=a[j] then kul:=0;
       until kul=1;
     end;
   for i:=1 to k do
     for j:=i+1 to k do
       if a[i]>a[j] then
        begin t:=a[i]; a[i]:=a[j]; a[j]:=t; end;
   for i:=1 to k do
     begin
      b[2*i-1]:=(a[i]-1) div n+1;
      b[2*i]:=(a[i]-1) mod n+1
     end;
   for i:=1 to k do c[i*2-1]:=b[i*2-1];
   for i:=1 to k do c[2*i]:=n+1-b[2*i];
   for i:=1 to k do
    for j:=i+1 to k do
      if c[2*i-1]=c[2*j-1] then
       if c[2*i]>c[2*j] then
        begin t:=c[2*i]; c[2*i]:=c[2*j]; c[2*j]:=t; end;
   lob:='';
   for i:=1 to k*2 do
     begin
      write(f2,b[i]);
      lob:=lob+chr(b[i]+48);
     end;
   writeln(f2);
   loc:='';
   for i:=1 to k*2 do
     begin
      write(f2,c[i]);
```

```
      loc:=loc+chr(c[i]+48);
    end;
  write(f2,' ');
  igazb:=0;
  igazc:=0;
  hossz:=2*k+2;
  reset(f1);

  bal:=0; jobb:=(filesize(f1) div hossz)-1; { binaris kereses}
  while (bal) and (igazb=0) do
  begin
     kozep := (bal+jobb) div 2;
     seek (f1, kozep*hossz);
     for i:=1 to hossz-2 do read(f1,o[i]);
     lo:='';
     for i:=1 to hossz-2 do lo := lo+o[i];
     if lob=lo then igazb:=1
          else if lob< lo then
            jobb := kozep-1 else bal:=kozep+1;
  end;

  bal:=0; jobb:=(filesize(f1) div hossz)-1; { binaris kereses}
  while (bal ≤ jobb) and (igazc=0) do
  begin
     kozep := (bal+jobb) div 2;
     seek (f1, kozep*hossz);
     for i:=1 to hossz-2 do read(f1,o[i]);
     lo:='';
     for i:=1 to hossz-2 do lo := lo+o[i];
     if loc=lo then igazc:=1
          else if loc< lo then
            jobb := kozep-1 else bal:=kozep+1;
  end;

  if igazb+igazc=2 then
    if lob=loc
    then writeln(f2,'OK')
    else writeln(f2,'Mindketto benne van!');
  if igazc+igazb=0 then
    writeln(f2,'Egyik sincs benne!');
  if igazc+igazb=1 then writeln(f2,'OK');

  close(f1);

 end;
 close(f2);
END.
```

Kémia

K.L. 132. Az *A* anyag vegyelemzésekor egy 0,312 g tömegű próba égetésekor 0,66 g CO_2-t és 0,324 g vizet kaptak. Oxigénre vonatkoztatott sűrűségét 3,25-nek mérték. amennyiben egy 416 mg-os tömegű próbát fölös nátriummal kezeltek, 89,6 ml normál állapotú hidrogén képződédést észlelik. Megállapították, hogy az A kénsavval nem képes intramolekuláris vízvesztésre. Határozd meg az A szerkezeti képletét.