millió egységnyi vízben 1 egységnyi vér észlelésére is képesek (tehát $10^{-8}$-szoros hígításban érzékelnek).

3. Egy kaliforniai cég olyan karórához hasonló műszert fejlesztett ki GlucoWatch néven, amely a bőrből fájdalmatlanul vérpróbát vesz, melyből a szőlőcukor mennyiséget folyamatosan meghatározza, s így ellenőrzi a beteg vércukor szintjét. A készülék nem tűvel veszi próbáját, hanem két elektródja közti feszültség hatására a sejtnedvek elektrolitjai vándorlása során ionos részecskék, melyek magukkal sodornak szőlőcukor molekulákat is, így kerülnek a készülék érzékelőjébe. A cukormolekulák mennyiségét fokozatosan méri a készülék, s az adatokat az elektronikus memóriájában tárolja, ahonnan bármikor előhívhatók, s az "óra" számlapján leolvashatók. A kutatók már azon dolgoznak, hogy a mért eredmények alapján a szükséges inzulin mennyiséget is folyamatosan adagolhassa az óra a beteg szervezetébe a próbavétellel ellentétes irányú áramoltatással.

Popular Science nyomán **Máthé Enikő**

# Kerekasztal-problémák

1. Egy kerekasztal körül naponta találkozik $2n+1$ miniszter. Hogyan kell elhelyezkedniük, hogy mindennap különböző szomszédjaik legyenek? Hány napot tarthat a konferencia?

2. Általánosítsuk a feladatot tetszőleges $n$-re!

3. $n$ fiú és $n$ lány körtáncot járnak. Hány kört alkothatnak úgy, hogy minden fiú csak lányokkal legyen szomszéd és minden körben különböző lányokkal?

Az első és a harmadik feladatot megpróbáljuk matematikailag megközelíteni. Kezdjük az elsővel!

**1.** Ha a minisztereket csomópontoknak tekintjük egy gráfban, akkor a feladatot visszavezethetjük egy gráfelméleti problémára, vagyis egy $2n+1$ csomópontú teljes gráf különböző Hamilton-köreinek a meghatározására.

**Definíció:**

*A **gráf** matematikai objektum, amely geometriailag csomópontokból és az őket összekötő élekből áll. Egy gráfot akkor nevezünk **teljesnek**, ha minden egyes csomópontja össze van kötve az összes többivel.*

*A **Hamilton-kör** olyan zárt út a gráfban, amely minden egyes csomóponton egyszer halad át. Kivételt képez a kezdő csomópont, amely egyben végcsomópont is. Két Hamilton-kör élfüggetlen ha nincs közös élük.*

Matematikailag bizonyítható, hogy egy $2n+1$ csomópontot tartalmazó teljes gráfnak, $n$ élfüggetlen Hamilton-köre létezik.

Helyezzük a csomópontokat egy körre a következőképpen:

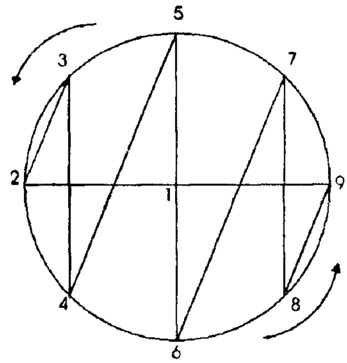
*(i) Az első csomópontot tegyük a kör középpontjába.*

*(ii) A többit helyezzük egy szabályos $2n$ oldalú sokszög csúcspontjaiba.*

9 miniszter esetében az 1. ábrán látható felállításhoz jutunk. Ez lenne az első lehetséges megoldás. A többi megoldást az ábrán látható ciklus forgatásából kapjuk. 9 miniszter esetében a következő elhelyezések léteznek:

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

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



1. ábra

A fenti algoritmust használja a következő Pascal program:

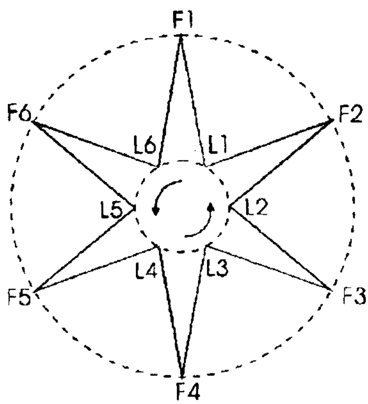
```
var
  x: array[ 1..100] of Integer;
{ Az elhelyezések tárolására}
  n: Integer;
{ A miniszterek száma }
{ A kezdeti felállítás }

procedure Helyez;
var
  b, j, k: Integer;
begin
   x[ 1] :=2; x[ 2] :=3;
   k:=4;b:=3; j:=n-1;
   while (k< r., do begin
      x[ j] :=k; k:=k+1;j:=j-1;
      x[ b] :=k; k:=k+1;b:=b+1;
   end
end;

procedure Kiir;
var
  j,b: Integer;
begin
   writeln;
   write(' 1' ); write(x[ 1] ,' ' ,x[ 2] ,' ' );
   b:=3; j:=n-1;
   while(b<=j) do begin
     write(x[ j] ,' ' ,x[ b] ,' ' );
     b:=b+1;j:=j-1;
   end;
   write(' 1' );
end;

procedure Forgat;
var
  seged, i, l: Integer;
begin
   Kiir;
   for l:=1 to n div 2 -1 do begin
     seged := x[ 1] ;
     for i:= 2 to n-1 do
       x[ i-1] := x[ i] ;
     x[ n-1] := seged;
     Kiir
    end
end;

{ Fôprogram }

begin
   write(' Paratlan szamot: ' );
   readln(n);
   Helyez;
   Forgat;
end.
```

2. Tetszőleges $n$-re a feladatot a következő visszalépéses algoritmussal oldottuk meg

```
var
  x: array[ 1..100] of byte;
  szomszed: array[ 1..100, 1..100] of byte;
  n, nr: Byte;

procedure Init; forward;
procedure Back; forward;
procedure Kiir(var k: Integer); forward;
procedure Vissza(var k: Integer); forward;
function Probal(k: Byte): Boolean;forward;

{ Feltölti 0-val a tömböt}
procedure Init;
var
  i, j: Integer;
begin
   for i:= 1 to n do
     for j:=1 to n do
       szomszed[ i,j] := 0;
end;
procedure Kiir(var k: Integer);
var
  i: Integer;
begin
   writeln;
   write('Megoldás',nr,'.');
   for i:= 1 to n do write(x[ i] :4);
   inc(nr);
   k:= 2;
{ A kovetkezô megoldást a második elem újraválasztásával kezdjuk}
end;

{ Szomszéd törlése visszalépéskor}
procedure Vissza(var k: Integer);
var
  i,j,l: Integer;
  ok: Boolean;
begin
   dec(k);
   if(k1) then begin
       szomszed[ x[ k] ,x[ k-1]] :=0;
       szomszed[ x[ k-1] ,x[ k]] :=0
   end;
end;

procedure Back;
var
  k: Integer;
begin
{ Az elsô elem rögzített. k=2 }
   k:=2; x[ 1] := 1;
   while ( k1) do
      if probal ( k ) then
        if (k=n) then
          Kiir(k)
        else begin
              inc(k);
```

```
            x[ k] :=1;
        end
      else
        vissza(k);
end;

Function Probal(k: Byte) : Boolean;
var
  ok: Boolean;
  i, j: Integer;
begin
   ok:= False;
   while( x[ k] < n ) and ( not ok ) do
   begin
       inc(x[ k] );
       ok:= True;
  for i:=1 to k-1 do
    if x[ i] =x[ k] then   ok:=False;
  if ok then begin
   if (szomszed[ x[ k-1] , x[ k] ] =1) then ok := false;
   if (k=n) and
       (szomszed[ x[ n] ,x[ 1] ] =1)then ok:=False;
  end;
end;
if ok then begin
     szomszed[ x[ k-1] ,x[ k] ] := 1;
     szomszed[ x[ k] ,x[ k-1] ] := 1;
     if (k=n) then begin
       szomszed[ x[ n] , x[ 1] ] := 1;
       szomszed[ x[ 1] , x[ n] ] := 1
     end;
end;
Probal := ok;
end;

begin
   readln(n);
   nr:= 1;
   back;
   readln
end.
```

**3.** A harmadik feladat is visszavezethető gráfokra. Ebben az esetben egy *2n* csomópontú bikromatikus gráf különböző Hamilton-köre megtalalálásával egyenértékű.




2. ábra

**Definíció:**

*Egy gráfot* ***p-kromatikusnak*** *nevezünk $p \in N^*$, ha a csúcspontjait kiszínezhetjük p különbüző színnel úgy, hogy bármely két szomszédos csomópontnak különböző színe legyen. Azt a legkissebb p számot, amelyre a gráf p-kromatikus a gráf kromatikus számának nevezzük.*

Az algoritmust n=6 esetén vázoljuk. A fiúkat egy külső kör mentén helyezzük el, míg a lányokat egy belső kör mentén a 2. ábrán látható módon.

Minden egyes elhelyezés után a lányok körét kettővel elforgatjuk egy adott irányba. Látható, hogy [ n / 2 ] különböző elhelyezés lehetséges. n=6 esetében a következő elhelyezések lehetségesek:

```
F1L1F2L2F3L3F4L4F5L5F6L6F1
F1L3F2L4F3L5F4L6F5L1F6L2F1
F1L5F2L6F3L1F4L2F5L3F6L4F1
```

Ez a módszer páratlan *n* -re is helyesen működik. A következő Pascal program ezt az algoritmust használja az elhelyezések generálására.

```
var
 lanyok: array[ 1..100] of Integer;
 n,i,j,l,seged: Integer;
procedure kiir;
var
 i: Integer;
begin
 writeln;
 for i:=1 to n do
 write(' F' ,i,' ' ,' L' ,lanyok[ i] ,' ' );
end;
{ Fôprogram}
begin
  write(' n:' );readln(n);
  for i:=1 to n do
    lanyok[ i] := i ;
  kiir;
  for i:= 1 to n div 2 -1 do begin
    for l:=1 to 2 do begin
      seged:= lanyok[ 1] ;
      for j:= 2 to n do
       lanyok[ j-1] := lanyok[ j] ;
      lanyok[ n] :=seged;
    end;
    kiir
  end;
end.
```

*Felhasznált irodalom:*

Claude Berge: Teoria grafurilor și aplicațiile ei – Editura Tehnică, București, 1969

**Antal Margit**
Marosvásárhely

# Hogyan viselkedjünk az Interneten?

## Hálózati etikett 2. rész

### Levelezési listák, hírcsoportok

Sally Hambridge eredeti dolgozata, amelynek az alábbi szöveg csak egy része, elérhető a

```
http://www.stanton.dtcc.edu/stanton/cs/rfc1855.html
```

WWW-címről (pl. Netscape, Lynx, Internet Explorer böngészőkkel). A magyar változat (fordító: Négyesi Károly) szintén letölthető a