Ég és Föld vonzásában – a természet titkai

Informatikai tehetséggondozás:

Visszalépéses kiválogatás

TÁMOP-4.2.3.-12/1/KONV

MAGYARORSZÁG MEGÚJUL

A projektek az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósulnak meg.

1. Az összes lehetséges sorrend

Sokszor előfordul feladatként, hogy elő kell állítani valamilyen elemek összes lehetséges sorrendjét. Az egyszerűség kedvéért legyenek az elemek az 1,…,N sorszámok. Ha ugyanis bármi más elemekről lenne szó, akkor azokat helyezzük el egy tömbbe és a sorszámukat használjuk indexelésre!

Így például 4 elem összes lehetséges sorrendje:

```
1.: 1 2 3 4
2.: 1 2 4 3
3.: 1 3 2 4
4.: 1 3 4 2
5.: 1 4 2 3
6.: 1 4 3 2
7.: 2 1 3 4
8.: 2 1 4 3
9.: 2 3 1 4
10.: 2 3 4 1
11.: 2 4 1 3
12.: 2 4 3 1
13.: 3 1 2 4
14.: 3 1 4 2
15.: 3 2 1 4
16.: 3 2 4 1
17.: 3 4 1 2
18.: 3 4 2 1
19.: 4 1 2 3
20.: 4 1 3 2
21.: 4 2 1 3
22.: 4 2 3 1
23.: 4 3 1 2
24.: 4 3 2 1
```

Látszik ezen a számsorozaton, hogy mindegyik tagja 1 és 4 közötti (általános esetben 1 és N közötti), de minden szám pontosan egyszer fordul elő egy sorozatban.

Másképp ránézve: Vannak 1-essel, 2-essel, 3-assal és 4-essel kezdődő típusúak. A megoldás is erre fog építeni, adjuk meg az összes 1-essel kezdődőt, az összes 2-essel kezdődőt, …

Az alábbi programban N az elemek száma, i pedig azon helyek száma, ahova már tettünk elemet.

```
Összes sorrend(N,i):
  Ha i>N akkor db:=db+1; Y(db):=X
  különben Ciklus j=1-től N-ig
             Ha nem volt(i,j) akkor X(i):=j
                                    Összes sorrend(N,i+1)
           Ciklus vége
  Elágazás vége
Eljárás vége.
```

```
volt(i,j):
  k:=1
  Ciklus amíg k<i és X(k)≠j
    k:=k+1
  Ciklus vége
  volt:=(k<i)
Függvény vége.
```

2. Az összes lehetséges speciális sorrend

Az 1,…,n sorozat összes olyan permutációját állítsuk elő, ahol minden elem maximum 1 hellyel mozdul el a helyéről, azaz i-1≤X(i)≤i+1!

Így például 4 elem összes lehetséges olyan sorrendje, ahol az elemek a sorszámuktól legfeljebb 1 távolságra lehetnek:

```
1.: 1 2 3 4
2.: 1 2 4 3
3.: 1 3 2 4
4.: 2 1 3 4
5.: 2 1 4 3
```

Kiszámolhatjuk egy egyszerű rekurzív függvénnyel, hogy hány ilyen sorozat lehet. Egy ilyen sorrend vagy 1-essel kezdődik és mögötte N-1 szám összes lehetséges jó sorrendje van, vagy 2 1-gyel kezdődik és mögötte N-2 szám összes lehetséges jó sorrendje van.

Azaz az N szám összes lehetséges jó sorrendje megegyezik N-1 szám összes lehetséges jó sorrendje + N-2 szám összes lehetséges jó sorrendjével: ami ezek szerint egy Fibonacci-szám.

A rekurzív megoldásban tehát j értéke csak i-1, i vagy i+1 lehet, két kivétellel, az 1-est nem lehet előre hozni, az N értéket nem lehet hátrább tenni.

```
Összes jó sorrend(N,i):
  Ha i>N akkor db:=db+1; Y(db):=X
  különben Ciklus j=i-1-től i+1-ig
             Ha j≥1 és j≤N és nem volt(i,j)
                akkor X(i):=j; Összes jó sorrend(N,i+1)
           Ciklus vége
  Elágazás vége
Eljárás vége.
```

3. Szóösszeadó fejtörő

Jól ismert fejtörő, amelyben egy aritmetikai művelet kapcsol egybe szavakat. A feladat az, hogy a szavak egyes betűinek feleltessünk meg egy számjegyet úgy, hogy a művelet helyes eredményt szolgáltasson a szavakon.

Pl. `SEND + MORE = MONEY`, vagy hagyományosan leírva:

```
  SEND
+ MORE
 MONEY
```

A szavakban előforduló jelekhez (SENDMORY) keressük a 0..9 számjegyek egyértelmű hozzárendelését!

1. megoldási ötlet („algebrai hozzáállás”):

D+E=Y, N+R=E, … M=1

D+E=10+Y, N+R+1=E, … M=1

2. megoldási ötlet:

Az összes permutáció algoritmusára építünk. A betűkhöz számjegyeket rendelünk az összes lehetséges sorrendben, majd megnézzük, hogy ez a hozzárendelés jó-e az összegzés szerint.

A Jó eljárás ellenőrzi a permutáció – a feladat szempontjából való – helyességét, és gondoskodik az esetleges megoldás gyűjtéséről vagy kiírásáról.

A megfelelőség a (*) SEND + MORE – MONEY = 0 egyenletre. Ha

- 'S‘ X(1) értékű, akkor a (*)-ban X(1) *1000-rel van jelen;
- 'E' X(2) értékű, akkor X(2)*(100+1-10)=X(2)*91-gyel;
- 'N' X(3) értékű, akkor X(3)*(10-100)= X(3)*(-90)-nel;
- 'D' X(4) értékű, akkor X(4)*(1)-gyel;
- 'M' X(5) értékű, akkor X(5)*(1000-10000)=X(5)*(-9000)-rel;
- 'O' X(6) értékű, akkor X(6)*(100-1000)= X(6)*(-900)-zal;
- 'R' X(7) értékű, akkor X(7)*10-zel;
- 'Y' X(8) értékű, akkor X(8)*(-1)-gyel van jelen.
- továbbá az S és az M betűhöz nem rendelhetünk nullát, azaz X(1)≠0 és X(5) ≠0!

```
Megoldás(n,i):
  Ha i=n+1 és jó(X) akkor db:=db+1; Y(db):=X; Kiírás(X)
  különben Ciklus j=0-tól 9-ig
             Ha nem volt(i,j)
                akkor X(i):=j; Megoldás(n,i+1)
           Ciklus vége
  Elágazás vége
Eljárás vége.
```

```
jó(X):
  jó:=(X(1) *1000+X(2)*91+X(3)*(-90)+X(4)*1+X(5)*(-9000)+
       X(6)*(-900)+X(7)*10+X(8)*(-1)=0 és X(1)≠0 és X(5)≠0)
Függvény vége.
```

Egy megoldás kiírása a következőképpen nézhet ki:

```
Konstans szöveg='SENDMORY'

Kiírás(X):
  Ciklus i=1-től 8-ig
    Ki: szöveg(i),X(i)
  Ciklus vége
Eljárás vége.
```

Szebben kiírva:

```
  SEND     9567
+ MORE   + 1085
------   ------
 MONEY    10652
```

Megjegyzés: Hatékonyabb lenne, ha az utolsó feltételt kihasználnánk menet közben: továbbá az S és az M betűhöz nem rendelhetünk nullát, azaz X(1)≠0 és X(5) ≠0.

```
Megoldás(n,i):
  Ha i=n+1 és jó(X) akkor db:=db+1; Y(db):=X; Kiírás(X)
  különben Ha i=1 vagy i=5 akkor jj:=1 különben jj:=0
           Ciklus j=jj-től 9-ig
             Ha nem volt(i,j)
                akkor X(i):=j; Megoldás(n,i+1)
           Ciklus vége
  Elágazás vége
Eljárás vége.
```

Néhány hasonló fejtörő több összeadással:

```
  THIS
+  ISA
+GREAT
+ TIME
------
WASTER

  SATURN
+ URANUS
+NEPTUNE
+  PLUTO
--------
 PLANETS
```

```
   MARS
+ VENUS
+SATURN
+URANUS
-------
NEPTUNE
```

```
  HEART
+  EARS
+  NOSE
+THROAT
-------
 HEALTH
```

És egy fejtörő szorzással:

```
HE * HE = SHE
```

4. N-ből K az összes lehetséges sorrendben

Egy iskolai versenyen N tanuló vett részt. Meg szeretnénk tudni, hogy az első 3 helyen milyen lehetséges tanuló sorrendek fordulhatnak elő.

A feladat általánosítása: Válasszunk ki N elemből K különböző elemet az összes lehetséges sorrendben (ismétlés nélküli variáció)!

A megoldás ugyanolyan, mint N elem összes lehetséges sorbarendezése, csupán az így kapott sorozatoknak az első K tagját kell nézni. Ehhez nem is kell előállítani az összes sorrendet, hanem meg lehet állni akkor, amikor már az első K tagot ismerjük az adott sorozatból.

Így például 4 elemből 2 elem összes lehetséges sorrendje:

```
1.: 1 2
2.: 1 3
3.: 1 4
4.: 2 1
5.: 2 3
6.: 2 4
7.: 3 1
8.: 3 2
9.: 3 4
10.: 4 1
11.: 4 2
12.: 4 3
```

```
Összes sorrend(N,K,i):
  Ha i>K akkor db:=db+1; Y(db):=X
  különben Ciklus j=1-től N-ig
             Ha nem volt(i,j) akkor X(i):=j
                                    Összes sorrend(N,K,i+1)
           Ciklus vége
  Elágazás vége
Eljárás vége.
```

5. N-ből K az összes lehetséges növekvő sorrendben

Válasszunk ki N elemből K különböző elemet az összes lehetséges növekvő sorrendben (ismétlés nélküli kombináció)!

A megoldás ugyanolyan, mint az N elemből K különböző elemet az összes lehetséges sorrendben feladaté, de csak olyan megoldásokat fogadunk el, ahol a kapott sorozat monoton növekvő.

Így például 4 elemből 2 elem összes lehetséges növekvő sorrendje:

```
1.: 1 2
2.: 1 3
3.: 1 4
4.: 2 3
5.: 2 4
6.: 3 4
```

Ha a kapott sorozat csak monoton növekvő lehet, akkor persze az i-edik tagot elég az I-1. tagnál nagyobb lehetséges értékekre kipróbálni. Ugyancsak igaz, hogy a mögötte levőknek is kell hagyni értékeket, azaz a K. érték N-ig mehet, a K-1. érték N-1-ig, a K-2. érték N-2-ig, és így tovább.

```
Összes sorrend(N,K,i):
  Ha i>K akkor db:=db+1; Y(db):=X
  különben Ciklus j=X(i-1)+1-től-től N-(K-i)-ig
             ~~Ha nem volt(i,j) akkor~~ X(i):=j
                                    Összes sorrend(N,K,i+1)
           Ciklus vége
  Elágazás vége
Eljárás vége.
```

Megjegyzés: Azaz ez a feladat nem is visszalépéses kiválogatás, mert minden úton megoldáshoz jutunk.

Ha megengednénk az ismétlődéseket, akkor az algoritmus – a ciklus határai – egyszerűsödne.

```
Összes sorrend(N,K,i):
  Ha i>K akkor db:=db+1; Y(db):=X
  különben Ciklus j=X(i-1)-től-től N-ig
             Ha nem volt(i,j) akkor X(i):=j
                                   Összes sorrend(N,K,i+1)
           Ciklus vége
  Elágazás vége
Eljárás vége.
```

6. Feladatok a sakktáblán

N bástya elhelyezése egy NxN-es sakktáblán

Helyezzünk el egy NxN-es sakktáblán az összes lehetséges módon N bástyát úgy, hogy ne üssék egymást!

A bástyákról azt kell tudnunk, hogy a sorukban és az oszlopukban álló bábukat üthetik. Tehát úgy kell elhelyezni a bástyákat, hogy minden sorban és minden oszlopban is pontosan 1 bástya legyen!

Ebből következik, hogy a megoldásokban elég azt megmondani, hogy az 1. oszlopban levő bástya melyik sorban áll, a második oszlopban levő bástya melyik sorban áll, … és így tovább.

Mivel mindegyiknek különböző sorokban kell állnia, ezért a feladat ugyanaz, mint az N elem összes lehetséges sorrendjének előállítása.

Egy lehetséges megoldás N=8-ra:

				B			
			B				
					B		
		B					
						B	
	B						
							B
B							

```
Összes bástya(N,i):
  Ha i>N akkor db:=db+1; Y(db):=X
  különben Ciklus j=1-től N-ig
             Ha nem volt(i,j) akkor X(i):=j
                                    Összes bástya(N,i+1)
           Ciklus vége
  Elágazás vége
Eljárás vége.
```

```
volt(i,j):
  k:=1
  Ciklus amíg k<i és X(k)≠j
    k:=k+1
  Ciklus vége
  volt:=(k<i)
Függvény vége.
```

N vezér elhelyezése egy NxN-es sakktáblán

Helyezzünk el egy NxN-es sakktáblán az összes lehetséges módon N vezért úgy, hogy ne üssék egymást!

A vezérekről azt kell tudnunk, hogy a sorukban, az oszlopukban és az átlójukban álló bábukat üthetik. Tehát úgy kell elhelyezni a vezéreket, hogy minden sorban és minden oszlopban is pontosan 1 vezér legyen, és minden átlóban legfeljebb 1 vezér legyen!

Ebből következik, hogy a megoldásokban elég azt megmondani, hogy az 1. oszlopban levő bástya melyik sorban áll, a második oszlopban levő bástya melyik sorban áll, … és így tovább.

Egy lehetséges megoldás N=5-re és N=4-re:

		V		
				V
	V			
			V	
V				

	V		
			V
V			
		V	

Itt tehát nem elég azt megnézni egy új vezér elhelyezésekor, hogy volt-e már a j. sorban vezér, hanem azt is kell, hogy volt-e már az (i,j) mezővel egy átlóban vezér.

```
Összes vezér(N,i):
  Ha i>N akkor db:=db+1; Y(db):=X
  különben Ciklus j=1-től N-ig
             Ha nem volt(i,j) akkor X(i):=j
                                    Összes vezér(N,i+1)
           Ciklus vége
  Elágazás vége
Eljárás vége.
```

```
volt(i,j):
  k:=1
  Ciklus amíg k<i és X(k)≠j és i-k≠abs(j-X(k))
    k:=k+1
  Ciklus vége
  volt:=(k<i)
Függvény vége.
```

Megjegyzés: Amíg a bástyás feladatnak mindig van (méghozzá nagyon sok) megoldása, addig a vezéres feladatnak N=2 és N=3 esetén biztosan nincs egy megoldása sem, más N-ek esetén pedig vannak megoldások, de sokkal kevesebb, mint a bástyás feladatnál.

Példa: az alábbi sakktáblákra nem lehet elhelyezni 2, illetve 3 vezért úgy, hogy ne üssék egymást: