38. A gráfalgoritmusok alkalmazása

Állapotok és átmenetek

A gráf adattípus nagyon sokféle feladat megoldásánál alkalmazható. Rejtvények, játékok, közlekedési és szállítási problémák, stratégiai feladatok vizsgálhatók gráfokkal. A képfelismerés, szövegkritika, nyelvészeti analízis területén szintén bevethető a gráfelmélet. A gráfokat felhasználja az egymástól látszólag távol eső közgazdaságtan, pszichológia, szociológia, esztétika, biológia stb.

A legtöbb esetben a gráf felépítése is a program feladata. A feladatban előforduló lehetséges eseteket a feladat **állapotainak** tekintjük. A sakk egy állapota például a bábuk egy elhelyezkedése a táblán. Az állapotokat **átmenetek** kötik össze, melyek jelzik, hogy egy-egy állapotból egy lépésben mely más állapotokba juthatunk. A sakkjáték egy állapotában minden lehetséges lépés egy-egy átmenetet jelent egy másik állapotba.

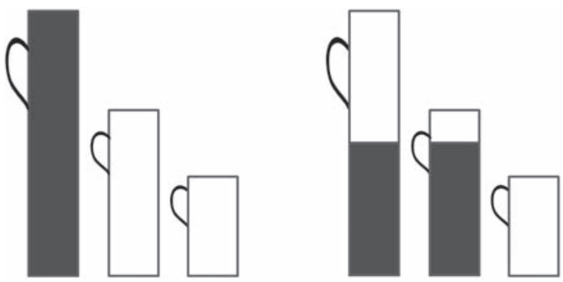
A feladat megoldásakor keressük a **kezdőállapotból** a kívánt **végállapotba** vezető átmenetek sorozatát.

A gráfok alkalmazásához az egyes állapotokat egy gráf csúcsainak feleltetjük meg. A gráf élei az átmeneteket jelzik. Útkeresési algoritmusokkal döntjük el, hogyan lehet a kezdőállapotból a végállapotba eljutni.

Kupák

Tekintsük a következő, közismert feladatot:

Egy 8 literes kupa tele van borral. Hogyan tölthetjük szét két egyenlő részre, ha ehhez egy 5 literes és egy 3 literes kupa áll még a rendelkezésünkre? Az öntögetések során egy kupa vagy teljesen tele lesz, vagy pedig teljesen kiürül.



A feladat kezdő- és végállapota

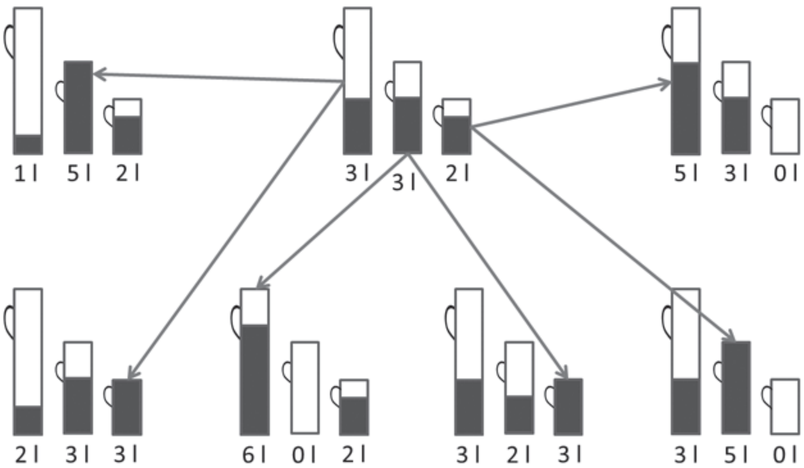
A feladatot némileg általánosítva oldjuk meg. A *Térfogat* tömb tartalmazza a kupák térfogatát, a *Kezd* tömb a kezdeti bormennyiséget, a *Vég* tömb a végső bormennyiséget az egyes kupákban. A fenti feladat szerint:

Térfogat(1) = 8	Térfogat(2) = 5	Térfogat(3) = 3
Kezd(1) = 8	Kezd(2) = 0	Kezd(3) = 0
Vég(1) = 4	Vég(2) = 4	Vég(3) = 0

Az állapotok tárolása

Feladatunknak látszólag semmi köze a gráfokhoz. A gráf kialakításához először határozzuk meg, milyen állapotokkal jellemezhetjük az előforduló eseteket! Az öntögetés egy állapotát a kupákban lévő bor mennyisége jellemzi. Egy lehetséges állapotot és a hozzá tartozó átmeneteket például az alábbi ábrán szemléltetjük.

Az állapotokat struktúrával írjuk le, amely egy háromelemű tömböt tartalmaz, a kupákban lévő bor mennyiségének megfelelően:



Lehetséges átmenetek egy meghatározott állapotból kiindulva
Az öntögetéseket a nyilak jelölik (honnan hová).

```
Struktúra TÁllapot
    VÁLTOZÓ Kupa(3) MINT Egész ' az egyes kupákban lévő bormennyiség
STRUKTÚRA VÉGE
```

Vegyük észre, hogy egy állapotot elegendő lenne két mennyiséggel jellemezni. Ha ismerjük az 5 és a 3 literes kupákban lévő bor térfogatát, akkor egyértelműen meghatározhatjuk a 8 literes kupában lévő mennyiséget. Csupán az áttekinthetőbb kód kedvéért tároljuk mindhárom értéket. Ez nem jelent túl nagy memóriafelhasználást.

Az egyes kupákban 0-tól a kancsó térfogatáig változhat a bor mennyisége. Ezért a gráf csúcsainak száma:

```
Gráf.CsúcsSzám = (Térfogat(2) + 1)*(Térfogat(3) + 1)
```

Az állapotokat egy *TÁllapot* típusú tömbben tároljuk:

```
VÁLTOZÓ Állapot(Gráf.CsúcsSzám) MINT TÁllapot
```

Megtehettük volna, hogy struktúra helyett kétdimenziós tömböt használunk, melynek első dimenziója az állapot, második dimenziója pedig a kupa sorszámát indexeli. A struktúra alkalmazása azonban könnyebben olvasható forráskódot eredményez.

1. gyakorlat. A *Kupa* forrásfájlban található kód alapján kövessük nyomon a feladat megoldását! Figyeljük meg, hogy a mondatszerű leírás utasításait hol kell kiegészíteni vagy felbontani további utasításokra a tanult programozási nyelv sajátosságai miatt. Elemezzük ezeket az eltéréseket, vizsgáljuk meg az eltérések okát!

A Visual Basic nem engedi meg a struktúra definíciójában szereplő tömb méretének megadását. Ezért a *Kupa* tömböket külön ciklussal kell dimenzionálni. Ennek módját az 1. gyakorlat megoldásában mutatjuk be.

Megjegyezzük, hogy az *Állapot* tömb lehetett volna a *TGráf* struktúra tagja is, hiszen a gráf csúcsainak tulajdonságát (a hozzájuk tartozó állapotot) tartalmazza. Nem akartuk azonban módosítani az előzőleg kialakított gráftípus definícióját.

Az állapottömb feltöltése

A kupákban lévő bormennyiség és az állapotok sorszáma (tehát a gráf csúcsainak sorszáma) között függvénnyel teremtünk kapcsolatot. A függvénynek tulajdonképpen egy kétdimenziós tömb (a 2. és 3. kupában lévő bor mennyisége) elemeihez kell sorszámot rendelnie. Ezt például sorfolytonos számozással érhetjük el:

		Bor a 3 literes kupában (Bor3)			
		0	1	2	3
Bor az 5 literes kupában (Bor2)	0	1	2	3	4
	1	5	6	7	8
	2	9	10	11	12
	3	13	14	15	16
	4	17	18	19	20
	5	21	22	23	24

Sorszámok hozzárendelése az állapotokhoz a 3 és 5 literes kupában lévő bor mennyisége alapján

```
FÜGGVÉNY TérfogatbólCsúcs(Bor2, Bor3 MINT Egész) MINT Egész
    ' Bor2 liter bor van a 2., Bor3 liter bor van a 3. kupában
    TérfogatbólCsúcs = (Térfogat(3) + 1) * Bor2 + Bor3 + 1
FÜGGVÉNY VÉGE
```

Egyetlen képlet miatt általában nem érdemes függvény készíteni. Itt ismét a forráskódot tesszük áttekinthetőbbé.

Ezek után az összes lehetséges módon fel kell írni az előforduló állapotokat, és hozzárendelni az *Állapot* tömb elemeihez:

```
CIKLUS Bor2 = 0-tól Térfogat(2)-ig
    CIKLUS Bor3 = 0-tól Térfogat(3)-ig
        Csúcs = TérfogatbólCsúcs(Bor2, Bor3)
        Állapot(Csúcs).Kupa(1) = ÖsszesBor – Bor2 – Bor3
        Állapot(Csúcs).Kupa(2) = Bor2
        Állapot(Csúcs).Kupa(3) = Bor3
    CIKLUS VÉGE
CIKLUS VÉGE
```

ahol:

```
ÖsszesBor = Kezd(1) + Kezd(2) +Kezd(3)
```

Az öntögetések kódolása

A gráf élei a bor öntését jelzik az egyik kupából a másikba. Egy átöntés eredményét az *ÉlVége* függvénnyel írjuk le. A függvényt akkor fogjuk felhasználni, amikor értéket adunk a csúcsmátrix elemeinek.

A függvény argumentumként megadjuk a kiindulási állapot/csúcs sorszámát (*ÉlKezdete*), és azt, hogy melyik kupából (*Melyikből*) öntünk bort melyik másik kupába (*Melyikbe*). A függvény meghatározza az öntés utáni állapot sorszámát, tehát annak a csúcsnak a sorszámát, ahová befut az állapotváltozást leíró él:

```
FÜGGVÉNY ÉlVége(Élkezdete, Melyikből, Melyikbe MINT Egész) MINT Egész
    …
FÜGGVÉNY VÉGE
```

Az alábbi utasításokban a *Kupa* tömbök az *Állapot*(*Élkezdete*) struktúra tagjai. Az áttekinthetőség kedvéért a pont elől lehagyjuk az *Állapot*(*Élkezdete*) tömbelemmel való minősítést. (Ezt az egyszerűsítést minősítőblokk alkalmazásával a forráskódban is megtehetjük.)

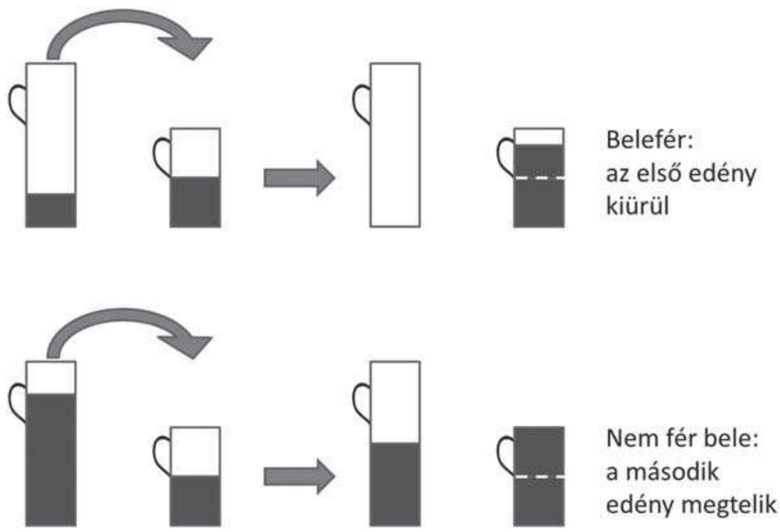
Az átöntésnél két eset fordulhat elő.

1. A *Melyikből* kupában lévő bor belefér a *Melyikbe* kupába, azaz:

```
.Kupa(Melyikből) < Térfogat(Melyikbe) – .Kupa(Melyikbe)
```

Ekkor a teljes mennyiséget átöntjük, így a *Melyikből* kupa kiürül. Az öntés utáni bormennyiségek:

```
.Kupa(Melyikbe) = .Kupa(Melyikbe) + .Kupa(Melyikből)
.Kupa(Melyikből) = 0
```



Öntés az egyik edényből a másikba

2. Ha a *Melyikből* kupában lévő bor nem fér bele teljes egészében a *Melyikbe* kupába, akkor átöntünk annyit, hogy ez utóbbi tele legyen. Ami nem fér bele, az marad a *Melyikből* kupában. Ekkor az öntés utáni bormennyiségek:

```
.Kupa(Melyikből) = .Kupa(Melyikből) – (Térfogat(Melyikbe) – .Kupa(Melyikbe)
.Kupa(Melyikbe) = Térfogat(Melyikbe)
```

Az *ÉlVége* függvény visszatérési értékét, azaz a gráf ezen élének végpontját a *TérfogatbólCsúcs* függvénnyel határozzuk meg:

```
ÉlVége = TérfogatbólCsúcs(.Kupa(2), .Kupa(3))
```

Ügyeljünk arra, hogy az *Élvége* függvény *Élkezdete* paraméterét feltétlenül érték szerint adjuk át, különben megváltoztatjuk az *Élkezdete* csúcshoz tartozó állapotot!

2. gyakorlat. A *Kupa* forrásfájlban található kód alapján tekintsük át az *ÉlVége* függvény definícióját!

Bár a Visual Basic-ben a struktúrák érték típusú adatszerkezetek, a struktúra tömb tagjai már hivatkozás típusúak. Így a struktúra érték szerinti paraméterátadásánál is az eredeti tömbre hivatkozik a változó, tehát módosítani tudja az elemeit. Ezért a *Kupa* forráskódjában bevezettünk egy ideiglenes változót (*Temp*), melynek átadjuk a *Kupa* tömb elemeit. Ezzel elkerüljük az eredeti állapot módosítását.

A csúcsmátrix feltöltése

A csúcsmátrix az állapotok között lehetséges átmeneteket tartalmazza. A gráf egyik csúcsából akkor vezet él egy másik csúcsba, ha egyetlen öntéssel eljuthatunk az egyik csúcsnak megfelelő állapotból a másik csúcsnak megfelelő állapotba.

A csúcsmátrix elkészítéséhez sorra vesszük az egyes állapotokat. Minden egyes állapotnál elvégezzük a lehetséges öntögetéseket. A csúcsmátrixban feljegyezzük, hogy egy-egy öntés során melyik állapotba, azaz melyik csúcsba jutottunk el.

Minden állapot esetén hatféle öntés képzelhető el. Bármely *nem üres* kupából önthetünk bort bármely *másik* kupába. (Tele kupába 0 litert öntünk át.) Az öntések utáni állapotot az *ÉlVége* függvény adja meg. Így a csúcsmátrix feltöltésének algoritmusa:

```
Gráf.Szomszédpont() = Hamis ' az összes elemre
CIKLUS Csúcs1 = 1-től Gráf.CsúcsSzám-ig
    CIKLUS Melyikből = 1-től 3-ig ' ebből a kupából öntünk
        CIKLUS Melyikbe = 1-től 3-ig ' ebbe a kupába öntjük
            HA Melyikből ≠ Melyikbe ÉS Állapot(Csúcs1).Kupa(Melyikből) > 0 AKKOR
                Csúcs2 = ÉlVége(Csúcs1, Melyikből, Melyikbe)
                Gráf.Szomszéd(Csúcs1, Csúcs2) = Igaz
            ELÁGAZÁS VÉGE
        CIKLUS VÉGE
    CIKLUS VÉGE
CIKLUS VÉGE
```

3. gyakorlat. Egészítsük ki a *Kupa* forrásfájl programját a csúcsmátrix kiírásával!

A feladat megoldása

Miután elkészítettük a feladatnak megfelelő adatszerkezetet, a megoldást már nagyon könnyen megkaphatjuk. Nem kell mást tennünk, mint az útkeresés algoritmusát lefuttatni a kezdőállapot és a végállapot között. Az útkeresés algoritmusán semmit nem módosítunk. A szélességi bejárásra alapozott útkeresés mindjárt a legegyszerűbb (legkevesebb öntögetéssel járó) átmenetet határozza meg.

Vegyük észre, hogy egy ügyesen megválasztott adatszerkezet nagyon egyszerűvé teszi a feladat megoldását!

4. gyakorlat. Egészítsük ki a *Kupa* forrásfájl programját a szélességi útkeresés algoritmusával! Futtassuk a programot, és határozzuk meg a feladat megoldását!

Feladatok

Az 1–6. feladatok a leckében bemutatott kupás feladatra vonatkoznak.

1. A megoldás kiírásánál az érintett csúcsok/állapotok sorszáma helyett jelenítsük meg a kupákban lévő bormennyiséget! Módosítsuk a kiírást úgy, hogy az útkeresésnél kapott fordított sorrend helyett a kezdőállapotból induljon a felsorolás!
2. Bővítsük a feladat megoldásának megjelenítését! Jelöljük meg, hogy az egyes lépésekben melyik kupából melyik kupába kell bort önteni!
3. A csúcsmátrix kiírásával vizsgáljuk meg, hogy mely állapotokba nem lehet egyáltalán eljutni az adott kupatérfogatok felhasználásával!
4. Ábrázoljuk síkbeli koordináta-rendszerben az egyes állapotokat! Tekintsük x koordinátának az 5 literes, y koordinátának a 3 literes kupában lévő bor mennyiségét! Hol helyezkednek el azok a pontok, melyek nem elérhető állapotokat jelölnek?
5. Oldjuk meg a kupás feladatot, ha 12, 7 és 4 literes kupák állnak a rendelkezésünkre! Kezdetben a 12 literes kupa tele van, és meg kell felezni a bormennyiséget. Módosítsuk a programot úgy, hogy ezeket az adatokat a felhasználótól kérje be!