Asztali alkalmazások fejlesztése – Java – 10. hét

November 3

2021

Mátrixok

Feladatlap

Tartalomjegyzék

01. feladat – Mátrix elmélet

Az eddig használt egydimenziós tömbökön túl használhatunk többdimenziós tömböket is. A többdimenziós tömbök valójában tömbök tömbjei. A dimenziók száma elméletileg nincs korlátozva, gyakorlatilag 3 dimenziónál többel dolgozni nem feltétlenül praktikus. Példaként nézzük meg, hogyan is képzelhetünk el egy 4x5 méretű kétdimenziós tömböt és hogyan hivatkozhatunk a tömb elemeire:

a[0,0] 0	a[0,1] 0	a[0,2] 0	a[0,3] 0	a[0,4] 0
a[1,0] 0	a[1,1] 0	a[1,2] 0	a[1,3] 0	a[1,4] 0
a[2,0] 0	a[2,1] 0	a[2,2] 0	a[2,3] 0	a[2,4] 0
a[3,0] 0	a[3,1] 0	a[3,2] 0	a[3,3] 0	a[3,4] 0

Láthatjuk, hogy az egyes elemekre az elem sorának és oszlopának megadásával hivatkozhatunk, például a harmadik sor ötödik eleme: a[2,4]. Az ábrán látható tömbben minden elem értéke 0. Mivel a többdimenziós tömbökkel egymásba ágyazott ciklusok segítségével dolgozunk (a külső ciklus adja meg a sorindexeket, a belső az oszlopindexeket), a tömb összes elemének 0-ra állítása is két ilyen egymásba ágyazott ciklus segítségével oldható meg:

```
int M = 4;
int N = 5;
int[,] a = new int[M,N];

for (int i = 0; i < M; i++)
{
    for (int j = 0; j < N; j++)
    {
        a[i,j] = 0;
    }
}
```

02. feladat – Mátrix és NégyzetesMátrix

Hozz létre egy **MatrixOOP** projektet, konzol-alkalmazásként, majd adj a projekthez egy **Matrix** osztályt!

A **Matrix** osztályban definiáld a következő adattagokat!

- A **sor, oszlop** adattagok a mátrix sorainak és oszlopainak a számát tárolják
- Az **m** adattag egy kétdimenziós egész elemű tömb, a mátrix elemeit tárolja
- A statikus **rnd** adattag az elemek véletlenszerű generálásához szükséges

Definiálj a Matrix osztályban egy két paraméteres konstruktort, ami véletlen egész értékekkel feltölti a mátrix elemeit. A sor és oszlop paraméterek inicializálják a sor és oszlop adattagokat, majd ezek alapján létrehoz egy ennek megfelelő méretű kétdimenziós tömböt. Sorfolytonosan bejárja a létrehozott kétdimenziós tömböt, és véletlenszerű egész értékekkel feltölti azt.

Írj ToString metódust, ami a Matrixot jeleníti meg úgy, hogy a mátrix elemeket 3 karakter szélességben látszódjanak.

Hozz létre és írass ki egy Matrix típusú objektumot! Hozz létre és írass ki más méretű mátrixokat is!

Bővítsd egy újabb metódussal a Matrix osztályt, amely

- meghatározza a Matrix elemeinek az összegét!
- megszámlálja a Matrix pozitív elemeinek a darabszámát!
- visszaadja a Matrix legnagyobb elemének az értékét!
- amely sorfolytonosan keresi az első pozitív elemet, és visszaadja a sor és oszlopindexét!
- amely egy adott sor elemeit összegzi! `public double sorOsszeg(int sor)`
- amely egy adott oszlopban megkeresi a legnagyobb elemet! `public double maxElemOszlop(int oszlop)`

Származtass a Matrix osztályból **NegyzetesMatrix** osztályt! Írj új, egyparaméteres konstruktort az osztályban, a paraméter a sorok száma! Írj getter-t a négyzetes mátrix méretének a lekérdezésére!

Bővítsd új metódusokkal a NegyzetesMatrix osztályt, készíts metódusokat, amelyek összegzik

- a főátló elemeit
- a mellékátló elemeit
- a főátló alatti elemeket
- a mellékátló fölötti elemeket

03. feladat – Galambok

Egy téren galambok csipegetnek. Ha az ember túl közel megy hozzájuk, arrébb totyognak, vagy elrepülnek.
Írjunk programot, ami néhány galamb és egy ember mozgását tartja nyilván a téren.

Hozzunk létre egy **Pozicio** osztályt, ami a galambok vagy az ember helyzetét képes eltárolni majd egy téren. Két adattag legyen az osztályban, x,y koordináták. Legyen egy *tavolsag* metódus is, ami két pozíció távolságát határozza meg.

Hozzunk létre egy **galamb** osztályt. A **Galamb** osztályban lehessen eltárolni a galamb pozícióját. Legyen egy *totyog*, és egy *repul* metódusa. A metódusok kapják meg paraméterként az ember helyzetét, és az embertől távolodva totyogjon (ha az ember 4 egységnél közelebb került hozzá, 5 egységnyit) vagy repüljön (ha az ember 2 egységnél közelebb került hozzá, 2 egységnyit) (Ha esetleg egy galamb a tér szélére kerül, akkor oldal irányban próbál menekülni).

Hozzunk létre egy **Ter** osztályt, ami tárolja a téren csipegető galambokat, és az ember pozícióját. A tér 50X50 egység nagyságú legyen.

Az osztálynak legyen egy mozog metódusa, ami egy új koordinátát kap paraméterként, és erre a pontra mozgatja az embert (a mozgásnak nem kell folyamatosnak lennie, "teleportálhat" az ember). Az ember mozgása során ellenőrizze a madarak és az ember távolságát, és ha az ember túl közel megy hozzájuk, a galambok totyogjanak vagy repüljenek arrébb.

Tervezzük meg, majd implementáljuk a fenti problémát megoldó programot. Írjunk programot, amivel tesztelhetjük az osztályainkat.

Kezdetben a téren 500 galamb csipeget. A galambok elhelyezkedése véletlenszerű. A program az ember mozgásának koordinátáit egy file-ból olvassa ki.
Hozzon létre egy file-t, amibe "rajzolja" ki a tér kezdeti állapotát (karakterekből), ábrázolva rajta a galambokat.

A program végén pedig egy új file-ba "rajzolja" ki a teret (karakterekből), ábrázolva rajta, hogy mely koordinátákon járt az ember, és a program végén hol vannak a galambok.

04. feladat – Balaton

A Balaton fölé egy négyzethálót fektettünk (B(N, M)), s a négyzetháló minden rácspontjában megmértük a hőmérsékletet. Ha egy rácspont a parton volt, akkor ott 0 értéket mértünk, a vízben pedig >0-át. A 'déli' és az 'északi' partot az ábrán 'x'-szel jelölt pontok választják el egymástól (a mért adatok ezekben a pontokban is 0 értékűek).

```
000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000++000
000000000000000000000000000000000000000000000000000+++000
00000000000000000000000000000000000000000++++++++++++++00
0++++++000000++00000000000000000000000+0++++++++++++++++0
x+++++++++++++++0000000000+0+++++++++++0++++++++++++++++x
0++++++++++++++++++++++++++++++++++++++00+++++++++++++++0
00++++++++++++++++++++++++++++++++++++++0+++++++++++++++0
000000+++++++++++++++++++++++++++++++++++++++++++++++0000
000000000000++++++++++++++++++++++++++0000000000000000000
000000000000000000000000000000000000000000000000000000000
```

- Határozzuk meg a déli part mentén, hol legmelegebb a víz hőmérséklete!
- Döntsük el, hogy a Balaton legmelegebb pontja a part mentén található-e!
- Döntsük el, hogy a Balaton partmenti átlaghőmérséklete magasabb-e, mint a Balaton más pontjainak átlaghőmérséklete!
- Határozzuk meg az x-szel jelölt pontokhoz legközelebbi legmelegebb pontokat!
- Döntsük el, hogy a Balaton északi vagy déli partján magasabb-e a víz átlaghőmérséklete!
- Határozzuk meg a Balaton partjának hosszát, ha R a rácspontok távolsága!
- Határozzuk meg a leghidegebb partmenti pontok helyét!
- Határozzuk meg a Balaton déli partjához legközelebbi legmelegebb pont helyét!
- Határozzuk meg az északi és a déli part legkisebb, illetve legnagyobb távolságát!
- Határozzuk meg a Balaton azon pontjait, ahol a leggyorsabban változik a víz hőmérséklete!
- Döntsük el, hogy a déli part mentén haladva keletről nyugatra, a víz hőmérséklete folyamatosan csökken-e! Adjuk meg a legnagyobb monoton csökkenő szakasz hosszát és végpontjait is!
- Döntsük el,hogy tetszőleges helyen észak felől déli irányban haladva, a víz hőmérséklete növekszik-e!
- Határozzuk meg a partmenti víz legmelegebb pontját!

05. feladat – Fertő-tó

Négyzetrácsot fektetünk a Fertő-tóra. 0-t írunk oda, ahol víz van a rácspontban, 1-et oda, ahol szárazföld van. Határozzuk meg a tó legdélibb, legkeletibb, legészakibb és legnyugatibb pontját!

06. feladat – Kutyaszépségverseny

Egy N résztvevőjű kutyaszépségversenyen M különböző szempont szerint pontoznak, s az eredményt a kutya(N, M) mátrix tartalmazza.

Minden szempont alapján MAX(i) pontot adnak. (Vagyis azt, hogy mely kategóriában maximum hány pontot kaphatnak a kutyusok azt egy külön tömb tárolja.)

A versenyből kiesik az a kutya, amelyik a valamelyik kategóriában nem éri el az alsó ponthatárt (Kategóriánként ezt is külön tömb tartalmazza)

- Adjuk meg az automatikusan kieső kutyák sorszámát!
- Döntsük el, hogy van e olyan kutya, amely minden kategóriában győztes!
- Döntsük el, hogy van e olyan kategória, ahol holtverseny alakult ki az első helyért!
- Döntsük el van-e olyan kutya, amely minden kategóriában kiesett!
- Döntsük el, hogy van –e olyan kutya, aki abszolút győztes (összpontszám), de minden kategóriában volt nála jobb!
- Döntsük el, hogy van-e olyan kutya, amely több kategóriában is győztes!
- Adjuk meg azokat a kategóriákat, ahol nem volt kieső kutya!

07. feladat – Sakkverseny

Egy nemzetközi sakkversenyen N ország küldte el versenyzőinek ÉLŐ – pontszám szerint rangsorolt listáját (+a pontszámokat.) Készítsük el az egész mezőny ÉLŐ- pontok alapján rendezett listáját!

08. feladat – Madártani megfigyelés

Egy madártani megfigyelés adatait M(I,J) mátrixban tároljuk, ahol M(I,J) az I városban a J. madár megfigyelt darabszámát jelenti. Rendezzük a mátrix sorait (a városokat) a megfigyelt össz-madár darabszám szerint növekvő sorrendbe!

09. feladat – Adás

Televízíónkon K csatorna adásait fogjuk. Ismerjük mindegyik adásidejét. Írjunk algoritmust, mely meghatározza, hogy mikor nincs egyetlen adás sem!

10. feladat - Karácsonyi vásár

Karácsonyi vásárban N helyen árulnak karácsonyfát. Az egyes árusok többféle fenyőt árulnak különböző árban.

- Állapítsuk meg, hol vehetjük meg a legolcsóbb fenyőt!
- Hányféle fenyőt árulnak összesen az árusok?
- Hány árus árulja az egyes fenyő fajtákat?
- Hol a legolcsóbbak az egyes fenyő fajták?

11. feladat – Metró

A budapesti metró 2-es vonalán forgalomszámlálást végeztek. M megálló esetén N mérés volt: Metro[M][N];

- Adjuk meg azon szakaszok számát, ahol egyre több felszálló volt!
- Adjuk meg vonatonként hány utas volt.
- Adjuk meg mely szakaszon volt a legkevesebb felszálló!

12. feladat – Ökölvívás

Egy kihívásos ökölvívó versenyen ismert minden induló eredménylistája. Írjunk programot, mely három részre választja szét a mezőnyt: az első csoportba azokat, akik a mérkőzéseik nagy részét megnyerték, a második csoportba azokat, akik ugyanannyi számú vereséget szenvedtek el, mint győzelmet, végül a harmadik csoportba azokat tették, akik több vereséget szenvedtek, mint ahány meccsen győztek.

13. feladat – Hőmérséklet

N napig mértük, naponta K alkalommal a hőmérsékletet!

- Adjuk meg azokat a napokat, amelyeken belül legnagyobb volt a hőmérsékletváltozás!
- Adjuk meg azokat a napokat, amelyeken a legnagyobb volt az eltérés a teljes időszak átlaghőmérsékletétől!
- Adjuk meg a teljes időszakra vonatkozó átlag alatti napokat!
- Adjuk meg hányszor volt rossz a műszer! – rossz volt a műszer, ha az adott érték a D értéknél jobban eltér a napi átlagtól!

14. feladat - Baktériumtörzs

A biológiai kísérletek kiértékelését igyekeznek automatizálni. Egy négyzet alakú táptalajon tenyésztett baktériumtörzs példányait lefényképezik, majd a fotókat több lépésben digitálisan feldolgozzák. A baktériumpéldányok különböző méretűek, alakúak és helyzetűek.

A táptalaj fényképét egy képzeletbeli négyzetháló segítségével cellákra osztják. Egy 50×50 cellából álló, az egyes baktériumpéldányokat már számokkal azonosítottan megjelenítő táblázat áll rendelkezésre a meres.txt tabulátorral tagolt, UTF-8 kódolású állományban.

A baktériumok az előfeldolgozott képen 1 és N közötti egész számmal (N≤50) vannak azonosítva. Egy-egy példány összefüggő területet alkot, de egy cella csak egy baktériumhoz tartozik. Ha egy cellában nincs baktérium, akkor ott a táblázatban nincs adat. A mintán két baktériumpéldány látható, az 1-es és a 4-es sorszámú.

			4	4	4	4	
				4	4	4	
			1	1	4		
	1	1	1	1	4		
			1	1			

Értékeljük ki és segítsük a további munkát

- Adjuk meg, hogy összesen hány baktérium van a képen.
- Adjuk meg, hogy hányas sorszámú baktérium foglalja el a legnagyobb területet a képen és ez hány cella.
- Adjuk meg annak a minimális méretű téglalapnak a szélességét és magasságát, amelyben a képen látható összes baktérium benne van.
- Ha van, akkor adjuk meg két érintkező baktérium sorszámát, ha nincs, akkor írjuk ki, hogy ,,Nincsenek egymással érintkező baktériumok.''

15. feladat - Mátrix Szorgalmi feladatsor

A feladatok nehézségei váltakoznak, nem egyre nehezedő feladatokat láthatsz, így nem feltétlen kell egymás után elvégezni azokat.

A legtöbb feladat négyzetes mátrixon értelmezett, amelynek a sor és oszlop méretei is azonosak.

Nevezetes Mátrix

- Dönts el egy mátrixról, hogy négyzetes mátrix-e!
- Dönts el egy mátrixról, hogy nullmátrix-e! (minden eleme 0 értékű)
- Dönts el egy mátrixról, hogy egységmátrix-e! (Egységmátrixnak csak a főátlóban vannak 1 értékei, míg minden más eleme 0)
- Dönts el egy mátrixról, hogy szimmetrikus-e! (Szimmetrikus elemek indexe m[i,j] vs. m[j,i])
- Dönts el egy mátrixról, hogy ferdeszimmetrikus-e!
- Dönts el egy mátrixról, hogy duplán sztochasztikus-e! (Egy A mátrix soraira nézve sztochasztikus, ha elemei nemnegatívak, és minden sorösszege 1, és oszlopaira nézve sztochasztikus, ha elemei nemnegatívak, és minden oszlopösszege 1. Ha soraira és oszlopaira nézve is sztochasztikus, akkor kétszeresen (duplán) sztochasztikusnak nevezzük.)
- Dönts el egy mátrixról, hogy mágikus négyzet-e! Ha igen, add meg a mágikus mátrix kulcsértékét!
- Dönts el egy mátrixról, hogy prím mágikus négyzet-e! (Az olyan mágikus négyzeteket, melyekben minden érték prímszám, prím mágikus négyzeteknek nevezzük)
- Dönts el egy mátrixról, hogy pánmágikus négyzet-e! (A pánmágikus négyzetek olyan mágikus négyzetek, ahol a sorokban és oszlopokban, illetve a főátlókban valamint az úgynevezett törtátlókban lévő elemek összege is egyenlő.)

Nevezetes részei a négyzetes mátrixoknak

- Írasd ki a főátlóban lévő értékeket!
- Határozd meg a mellékátlóban lévő értékek összegét!
- Határozd meg a mátrix felsőháromszögének átlagát! (A háromszögekbe a főátlóbeli elemek nem kerülnek be.)
- Határozd meg a mátrix alsóháromszögének legnagyobb elemét!
- Határozd meg a főátló melletti elemek összegét! (Főátló melletti, azaz közvetlenül alatta és felette elhelyezkedő)
- Határozd meg a mellékátló melletti elemek legkisebb értékét!
- Számítsd ki a főátlókban lévő elemek összegeit egy N elemű vektorba!
- Számítsd ki a törtátlókban lévő elemek átlagait egy N elemű vektorba!

Részmátrix

- Határozd meg a négyzetes (n x n-es) mátrix (n-i x n-i méretű) bal felső részmátrixát, ahol i eleme 1..n-1.
- Határozd meg az előző feladat szerinti jobb felső, bal alsó és jobb alsó részmátrixokat is!
- Határozd meg a mátrix legtöbb pozitív elemet tartalmazó részmátrixát!

Transzformációk

- Cseréld fel a mátrix i. és j. oszlopát!

- Cseréld fel a mátrix i. és j. sorát úgy, hogy az elemek fordított sorrendben legyenek az eredeti pozíciójuk alapján!
- Tükrözd a mátrixot a függőleges/vízszintes szimmetriatengelyére!
- Tükrözd a mátrixot a főátlóra/mellékátlóra!
- Rendezd a mátrix i. sorának/j. oszlopának elemeit növekvően!
- Rendezd a mátrix elemeit sorfolytonos bejárás szerint csökkenő sorrendben!
- Rendezd a mátrix elemeit oszlopfolytonos bejárás szerint növekvő sorrendben!

Matek *(Nézz utána az interneten a fogalmaknak!)*

- Határozd meg a mátrix **nyeregpontját/nyeregpontjait**! Egy A mátrix nyeregpontjának nevezzük azt az elemét, amely a legkisebb a sorában és legnagyobb az oszlopában. Egy mátrixban több nyeregpont is lehet, de elképzelhető, hogy egy sem található.
- Határozd meg a mátrix **paritáspontjait**!
- Határozd meg a mátrix **determinánsát**!
- Határozd meg a mátrix **inverzét**!
- Határozd meg a mátrix **rangját**!

Mátrixréteg elforgatás

Adott egy mátrix MxN-es mérettel és egy pozitív egész szám (R). Feladata elforgatni a mátrixot R-szer és kiírni az eredmény mátrixot. A forgatásnak óramutató járásával ellenkező irányban kell történnie és a forgatásnál, csak egy lépéssel kell eltolni az elemeket.

Megkötések

- $2 \leq m, n \leq 300$
- $1 \leq r \leq 10^9$
- min(m,n) % 2 = 0 Garantált, hogy m és n minimuma páros.

$1 \leq a_{ij} \leq 10^8$, ahol $i \in [1 \ldots m]$ és $j \in [1 \ldots n]$

Például a Start mátrix elforgatása 2-vel:

Start

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

Első forgatás

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

Második forgatás

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