DOMJ1R
skorka.bence@gmail.com

Feladat

Valósítsa meg az egész számokat tartalmazó felsőháromszög mátrixtípust (a mátrixok a főátlójuk alatt csak nullát tartalmaznak)! Ilyenkor elegendő csak a főátló és afeletti elemeket reprezentálni egy sorozatban, amelyet egy dinamikus helyfoglalású tömbben helyezzünk el. Implementálja önálló metódusként a mátrix iedik sorának j-edik elemét visszaadó műveletet, valamint hatékony összeadás és szorzás műveleteket, továbbá a mátrix (négyzetes alakú) kiírását, és végül a másoló konstruktort és az értékadás operátort!

Felsőháromszög mátrix osztály

A feladat lényege egy felsőháromszög mátrix típusnak a megvalósítása.

Típusérték-halmaz

Olyan számokat tartalmazó $n \times n$ $(n \in \mathbb{N})$ -es négyzetes mátrixokkal akarunk dolgozno, amelyek csak a felső háromszögben tartalmaznak nullától különböző elemeket. Az $n \in \mathbb{N}$ ennek a típusnak egy paramétere, amely a típusérték-halmaz mátrixának méretét határozza meg.

Típus-műveletek

Lekérdezés

A mátrix i-edik sorának j-edik pozícióján $(i, j \in [1..n])$ álló érték kiolvasása $e \coloneqq a[i, j]$

Megjegyezzük, hogy ez a művelet csak $i \geq j$ esetén igényel tényleges tevékenységet, hiszen egyébként a visszaadott elem nulla.

Felülírás

A mátrix i-edig sorának j-edig pozíciójára $(i, j \in [1..n])$ új érték beírása: $a[i, j] \coloneqq e$. A felső háromszögön kívüli elemet nem szabad felülírni, azaz $i \geq j$.

Megjegyezzük, hogy ez a művelet csak $i \geq j$ esetén igényel tényleges tevékenységet; $i < j$ esetén hibás, amennyiben egy nemnulla értéket akarunk a mátrixba tenni.

Összeadás

Két mátrix összeadása: $c \coloneqq a + b$. Az összeadásban szereplő mátrixos azonos méretűek.

$$\forall i, j \in [1..n]: c[i, j] \coloneqq a[i, j] + b[i, j]$$

Szorzás mátrixal

Két mátrix összeszorzása: $c \coloneqq a * b$. A szorzásban szereőlő mátrixok azonos méretűek.

$$\forall i, j \in [1..n]: c[i, j] \coloneqq \sum_{k=1}^{n} a[i, k] * b[k, i]$$

De mivel felső háromszög mátrixról van szó, így a szorzások száma csökkenthető azzal, hogy kihagyjuk a főátló alatti területeket:

$$\forall i, j \in [1..n]: c[i, j] \coloneqq \sum_{k=M_1}^{M_2} a[i, k] * b[k, i]\,, M_1 = Min\{i, j\},\ M_2 = Max\{i, j\}$$

Szorzás számmal

Egy mátrix szorzása egy számmal: $c \coloneqq a * \lambda$ ($\lambda \in \mathbb{N}$).

$$\forall i, j \in [1..n]: c[i,j] \coloneqq a[i,j] * \lambda$$

DOMJ1R
skorka.bence@gmail.com

Reprezentáció

Egy $n \times n$ -es felsőháromszög mátrixban csak a felső háromszöget kell ábrázlni:

$$\begin{matrix} a_{1,1} & a_{2,1} & a_{3,1} \dots a_{n,1} \\ 0 & a_{2,2} & a_{3,2} \dots a_{n,2} \\ 0 & 0 & a_{3,3} \dots a_{n,3} \end{matrix}$$

Ehhez egy 0-tól $\frac{n(n+1)}{2}$-ig indexelt egydimenziós tömbre (v) van szükségünk. Ennek segítségével a felsőháromszög mátrix bármelyik elemét meghatározhatjuk az alábbi függvény alapján: (az $index$ függvény az implementációban)

$$a[i,j] = \begin{cases} v[index(i,j)] & i \geq j \\ 0 & i < j \end{cases}$$

Implementáció

Segédfüggvények

$VektorMeret(n)$

$Return\left(\frac{n(n+1)}{2}\right)$

$index(i,j)$

$Return\left(i - j + \frac{(2n - j + 1) \times j}{2}\right)$

Lekérdezés

$Lekerdezes(i,j)$

$i \geq j$	
$Return(v[index(i,j)])$	$Return(0)$

Felülírás

$Módosítás(i,j,ertek)$

$i \geq j$	
$v[index(i,j)] \coloneqq ertek$	$SKIP$

DOMJ1R
skorka.bence@gmail.com

Összeadás

$Összeg(m_1, m_2)$

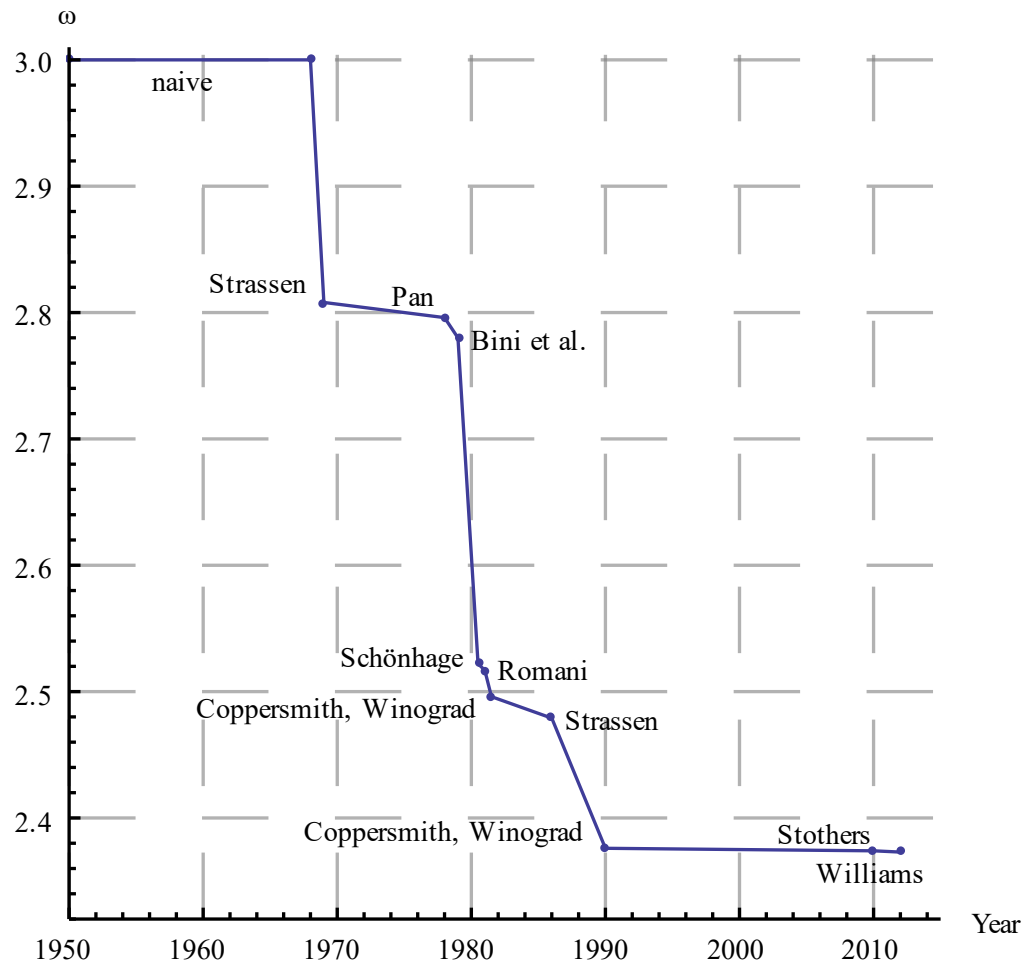
$m_1.meret() = m_2.meret()$	
$o \coloneqq new\ Matrix(m_1.meret())$	$SKIP$
$i \coloneqq 1..VektorMeret(o.meret())$	
$o.v[i] \coloneqq m_1.v[i] + m_2.v[i]$	
$Return(o)$	

Szorzás

$Szorzat(m_1, m_2)$

$m_1.meret() = m_2.meret()$	
$o \coloneqq new\ Matrix(m_1.meret())$	$SKIP$
$x \coloneqq 1..o.meret()$	
$y \coloneqq 1..x$	
$tmp \coloneqq 0$	
$i \coloneqq Min\{x, y\}..Max\{x.y\}$	
$tmp \coloneqq tmp + m_1.v[index(x, i)] * m_2.v[index(i, y)]$	
$o.v[index(x, y)] \coloneqq tmp$	
$Return(o)$	

Igen, ez egy n^3 -ös algoritmus, de egyszerű implementációban nincs is sokkal jobb sajnos:



Osztály

A felsőháromszög matrixok típusát osztály segítségével úgy valósítjuk meg, hogy a matrix méretét a konstruktorban paraméterként át kell adni. Ennek hátránya, hogy minden művelet előtt ellenőrizni kell, hogy a műveletet végző mátrixok azonos méretűek e. Előnye, hogy a matrix mérete később dinamikusan változhat másolás, vagy fileból betöltés esetén.

$\boldsymbol{UpperTriangularMatrix}$
$-size: int$ $-data: int[0..k], ahol\ k\ értékét\ a\ calculateVectorLength\ függvény\ számolja.$
$-calculateVectorLength(): int$ $-calculateVectorPosition(int, int): int$ $-copyFrom(UpperTriangularMatrix)$ $+UpperTriangularMatrix(int)$ $+UpperTriangularMatrix(UpperTriangularMatrix)$ $+operator()(int, int): int$ $+operator+(UpperTriangularMatrix, UpperTriangularMatrix): UpperTriangularMatrix$ $+operator*(UpperTriangularMatrix, int): UpperTriangularMatrix$ $+operator*(UpperTriangularMatrix, UpperTriangularMatrix): UpperTriangularMatrix$ $+writeData(ostream)$ $+readData(istream)$ $+operator\ll(ostream), nem\ ugyanaz, mint\ a\ writeData!!$ $+getSize(\ ): int$

Az adatokat tároló tomb egy saját vektortípus, ami kezeli a dinamikus helyfoglalást. Így a hely lefoglalásával nem a felső háromszög matrix osztály törődik, mert nem neki a feladata.

DOMJ1R
skorka.bence@gmail.com

Tesztelési terv

Vektorosztály

A vektorosztályból csak a mátrixot érintő funkciók tesztelése érdemes, így a $push\_back$, pop_back, egyéb függvények nem.

1. Vektor méretezhetősége
    a. Létrehozható e vektor adott mérettel
    b. Átméretezhető e a vektor
2. Vektorelemek elérhetősége
    a. Egy n méretű ektor $0..n-1$ elemei közül mindegyik elérhető?
    b. ~ módosítható?
    c. ~ megtartja az értékét?

Mátrixosztály

1. Új n méretű üres
    a. A mátrix létrejön?
    b. Minden eleme 0?
2. Helyes pozíciók elérése
    a. Elérhető?
    b. Alapértelmezetten 0?
3. Helytelen pozíciók elérése
    a. Hibát dob?
4. Szorzás számmal
5. Összeadás mátrixal
6. Szorzás mátrixal