Documentation for first OOP assignment

Saifeddin Alkurdi

Neptun Code: O9FFDC

Group 4

Task

Implement the chessboard matrix type which contains integers. In these matrices, every second entry is zero. The entries that can be nonzero are located like the samecolored squares on a chessboard, with indices (1, 1), (1, 3), (1, 5), ..., (2, 2), (2, 4), ... The zero entries are on the indices (1, 2), (1, 4), ..., (2, 1), (2, 3), ... Store only the entries that can be nonzero in row-major order in a sequence. Don't store the zero entries. Implement as methods: getting the entry located at index (i, j), adding and multiplying two matrices, and printing the matrix (in a shape of m by n).

Diagonal Matrix Type

Set Of Values

Diag(n) = { a $\in \mathbb{Z}^{n \times n}$ | $\forall$ i,j$\in$[1..n]: i≠j → a[i,j]=0 }

Operations

1. **Getting an entry**

Getting the entry of the ith column and jth row (i,j$\in$[1..n]): e:=a[i,j].

Formally:

A : Diag(n) × $\mathbb{Z}$ × $\mathbb{Z}$ × $\mathbb{Z}$

a i j e

Pre = (a=a' ∧ i=i' ∧ j=j' ∧ i,j$\in$[1..n])

Post = (Pre ∧ e=a[i,j])

This method fills out the diagonal numbers that are I and J but anything else is a zero

2. **Setting an entry**

Setting the entry of the ith column and jth row (i,j$\in$[1..n]): a[i,j]:=e. Entries outside the diagonal cannot be modified (i=j).

Formally:

A = Diag(n) × ℤ × ℤ × ℤ

a i j e

Pre = (e=e' ∧ a=a' ∧ i=i' ∧ j=j' ∧ i,j∈[1..n] ∧ i=j)

Post = (e=e' ∧ i=i' ∧ j=j' ∧ a[i,j]=e ∧ ∀ k,l∈[1..n]: (k≠i ∨ l≠j)→ a[k,l]=a'[k,l])

3. **Addition**

Sum of two matrices: c:=a+b. The matrices need to have the same size.

Formally:

A = Diag(n) × Diag(n) × Diag(n)

a b c

Pre = (a=a' ∧ b=b')

Post = (Pre ∧ ∀ i,j∈[1..n]: c[i,j]= a[i,j] + b[i,j])

4. **Multiplication**

Multiplication of two matrices: c:=a*b.

The matrices have the same size [x,y]=[x,y].

Formally: A = Diag(n) × Diag(n) × Diag(n)

a b c

Pre = (a=a' ∧ b=b')

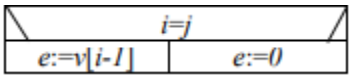
Post = (Pre ∧ ∀i,j∈[1..n]: c[i,j]= Σ k=1..n a[i,k] * b[k,j])

Representation The nn matrix only needs to have its diagonal strored.

Implementation

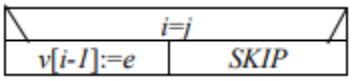
1. Getting an entry

obtaining the item for the jth row and the ith column



2. Making a record

Setting the ith column and jth row as the entry



3. Addition

Matrix c (represented by array u) is the product of matrices a and b (represented by arrays t and u), where all of the arrays must be the same size.

$\forall\ i \in [0..n-1]$: u[i]:= v[i] + t[i]

1. Start

2. Do the following for I between 0 and n-1:

Put v[i] and t[i] together to form u[i].

3. End for

4.End

4. Multiplication

Matrix c (represented by array u) is the product of matrices a and b (represented by arrays t and u), where each array must have the same size.

$\forall\ i \in [0..n-1]$: u[i]:= v[i] * t[i]

1. Start

2. Do the following for I between 0 and n-1:

Now we get the product of v[i] and t[i] together to form u[i].

3. End for

4.End

Testing

<u>Black box testing</u>

- ***Establishing a legitimate list of values on the chess board***

x = [1.2,3.2,3.4] as an input .0]

Expected Output: Without raising an exception, the values on the chess board should be set to the specified list.

- ***Making a chess board that is the right size***

Specify: size = 8

No exceptions should be raised as a result.

<u>White box testing</u>

- ***Test the constructor Chess(int size):***

Input: size = 0

Expected output: NegativeSizeException is thrown

- ***Test the indexer this[int row, int col] for getting an element:***

Input: row = -1, col = 0

Expected output: ReferenceToNullPartException is thrown