Tuguldur Sosorbaram
EERDEY
Sutu0318@gmail.com

Group 4
Assignement 1
18th March 2022

TASK

Implement the block matrix type which contains integers. These are square matrices that can contain nonzero entries only in two blocks on their main diagonal. Let the size of the first and second blocks be b1 and b2, where 1≤b1,b2≤n-1 and b1+b2=n (in the example, b1=4 and b2=5). Don't store the zero entries. Store only the entries that can be nonzero in a sequence or two smaller matrices. Implement as methods: getting the entry located at index (i, j), adding and multiplying two matrices, and printing the matrix (in a square shape).

Operations

1.Getting entry

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

$A : Matrix(n) \times \mathbb{Z} \times \mathbb{Z} \times \mathbb{Z}$

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

$Post = (Pre \wedge e=a[i,j])$

This operation needs any action only if inFirstSquare(I,j) or inSecondSquare is true, otherwise the output is 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 squares cannot be modified.

$A = Matrix(n) \times \mathbb{Z} \times \mathbb{Z} \times \mathbb{Z}$

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

$Post = (e=e' \wedge i=i' \wedge j=j' \wedge a[i,j]=e \wedge \forall k,l \in [1..n]: (k \neq i \vee l \neq j) \rightarrow a[k,l]=a'[k,l])$

This operation needs any action only if inFirstSquare(I,j) or inSecondSquare is true, otherwise the output is zero.

3. Sum

Adding 2 matrices even outside the 2 squares.

$A = Matrix(n1) \times Matrix(n2) \times Matrix(n3)$

$Pre = (a=a' \wedge b=b' \wedge n1=n2=n3)$

Post = (Pre $\wedge$ $\forall i,j \in [1..n]$: sum[i,j] = a[i,j] + b[i,j])

3. Sum

Multiplying 2 matrices even outside the 2 squares.

A = Matrix(n1) × Matrix(n2) × Matrix(n3)

Pre = (a=a’ $\wedge$ b=b’ $\wedge$ n1=n2=n3)

Post = (Pre $\wedge \forall i,j \in [1..n]$: mul[i,j]= $\sum$k=1..n a[i,k] * b[k,j])

Representation

Only in the 2 small squares on the diagonal can contain nonzero entity.

```
a11   a12   a13   ..   a1b1   ..     0    0    ..   0
a21   a22   a23   ..   a2b1   ..     0    0    ..   0
a31   a32   a33   ..   a3b1   ..     0    0    ..   0
..    ..    ..    ..   ..     ..     0    0    ..   0
ab11 ab12   ab13  ..   ab1b1  ..     0    0    ..   0
0     0     0     ..   0      ab2b2  x    x    ..   x
0     0     0     ..   0      x      x    x    ..   x
..    ..    ..    ..   ..     ..     ..   ..   ..   x
0     0     0     0    0      x      x    x    ..   x
```

I need 2 separate one-dimension array to save 2 small squares.

Vec1 and Vec2

inFirstSquare (i,j) = ($1 \le i \le b1$ $\wedge$ $1 \le j \le b1$)

inSecondSquare (i,j) = ($b2 \le i \le n$ $\wedge$ $b2 \le j \le n$)

Implementation

1.Getting an entry

Getting the entry of the ith column and jth row ($i,j \in [1..n]$) e:=a[i,j] where the matrix is represented by a and n stands for the size of the matrix can be implemented as

If (i,j) is both less than n and inside one of the 2 small squares e=a[i,j]

If (i,j) is both less than n and outside of 2 small squares e = 0

2. Setting an entry

Setting the entry of the ith column and jth row ($i,j \in [1..n]$) $a[i,j]:=e$ where the matrix is represented by a and n stands for the size of the matrix can be implemented as

If (i,j) is both less than n and inside one of 2 small squares a[i,j] = e;

3. Summation

The sum of matrices a and b goes to matrix sum, where all of the arrays have to have the same size.

$\forall i \in [0..n]$: sum[i,j]:= a[i,j] + b[i,j]

4. Multiplication

The product of matrices a and b goes to matrix mul where all of the arrays have to have the same size.

$\forall i,j \in [1..n]$: mul[i,j]= $\sum$k=1..n a[i,k] * b[k,j]

Testing

Black box testing

1. Creating, reading, and writing matrices of different size.
    a. 0, 1, 2, 5-size matrix
2. Getting and setting an entry
    a. Getting and setting an entry inside first and second squares
    b. Getting and setting an entry outside first and second squares
    c. Try getting and setting an entry outside bigger matrix
3. Sum of two matrices, command c:=a+b.
    a. Try with different size of matrices
    b. Checking the commutativity (a + b == b + a)
    c. Checking the neutral element (a + 0 == a, where 0 is the null matrix)
4. Multiplication of two matrices, command c:=a*b.
    a. Try with different size of matrices
    b. Checking the commutativity (a * b == b * a)
    c. Checking the neutral element (a * 0 == 0, where 0 is the null matrix)
    d. Checking the identity element (a * 1 == a, where 1 is the identity matrix)

White box testing

1. Creating an extreme-size matrix (-1, 0, 1, 1000).
2. Generating and catching exceptions.