Name: Bakhtishod Umurzokov

Neptun code: Y62UF3

Assignment 1st. Task 2nd.

Group: №6

Task

Implement the X matrix type which contains integers. These are square matrices that can contain nonzero entries only in their two diagonals. Don't store the zero entries. Store only the entries that can be nonzero in a sequence. Implement as methods: getting the entry located at index (i, j), adding and multiplying two matrices, and printing the matrix (in a square shape).

X matrix type

Set of values

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

Operations

1. Getting an entry

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

Formally: $A : \underset{a}{Diag(n)} \times \underset{i}{\mathbb{Z}} \times \underset{j}{\mathbb{Z}} \times \underset{e}{\mathbb{Z}}$

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

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

This operation needs any action only if i=j or i+j=Size+1, otherwise the output is zero.

2. Setting an entry

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

Formally: $A : \underset{a}{Diag(n)} \times \underset{i}{\mathbb{Z}} \times \underset{j}{\mathbb{Z}} \times \underset{e}{\mathbb{Z}}$

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

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

This operation needs any action only if i=j or i+j=Size+1, otherwise it gives an error if we want to modify a zero entry.

3. Sum

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

Formally: $A : \underset{a}{\mathrm{Diag}(n)} \times \underset{b}{\mathrm{Diag}(n)} \times \underset{c}{\mathrm{Diag}(n)}$

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.

Formally: $A : \underset{a}{\mathrm{Diag}(n)} \times \underset{b}{\mathrm{Diag}(n)} \times \underset{c}{\mathrm{Diag}(n)}$

Pre = (a=a’ ∧ b=b’)

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

Representation

Only the diagonal of the n×n matrix has to be strored.

$$a = \begin{matrix} a_{1\,1} & 0 & 0 & \dots & a_{1\,n} \\ 0 & a_{2\,2} & 0 & \dots & 0 \\ 0 & 0 & a_{3\,3} & \dots & 0 \\ \dots & \dots & \dots & \dots & 0 \\ a_{n\,1} & 0 & 0 & \dots & a_{n\,n} \end{matrix} \quad \leftrightarrow v = < a_{1\,1}\ a_{1\,n}\ a_{2\,2}\ a_{3\,3}\ a_{n\,1}\ a_{n\,n} >$$

Only a one-dimension array (v) is needed, with the help of which any entry of the diagonal matrix can be get:

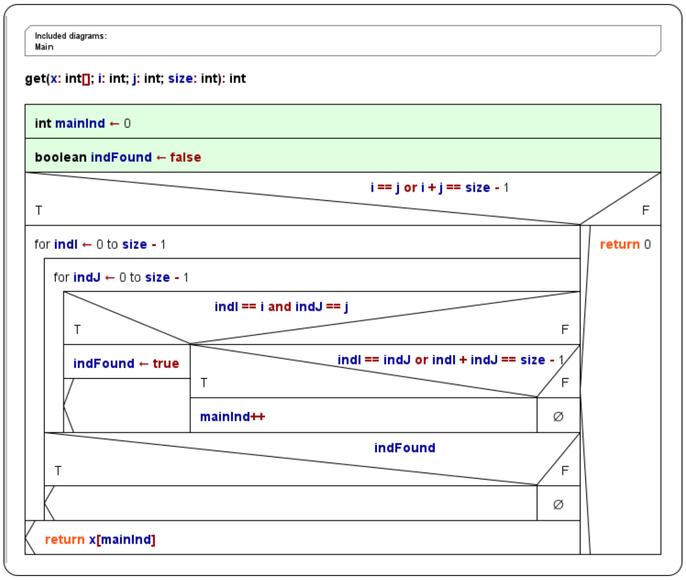
$a[i,j] = \{\ v[i]\ if\ i = j\ 0\ if\ i \neq j\ \}$

$$a[i,j] = \left\{ \begin{matrix} v[i] & if\ \ i = j \text{ or i+j=Size+1} \\ 0 & if\ \ i \neq \mathrm{j} \text{ or } \mathrm{i} + \mathrm{j} \neq \mathrm{Size} + 1 \end{matrix} \right\}$$

Implementation

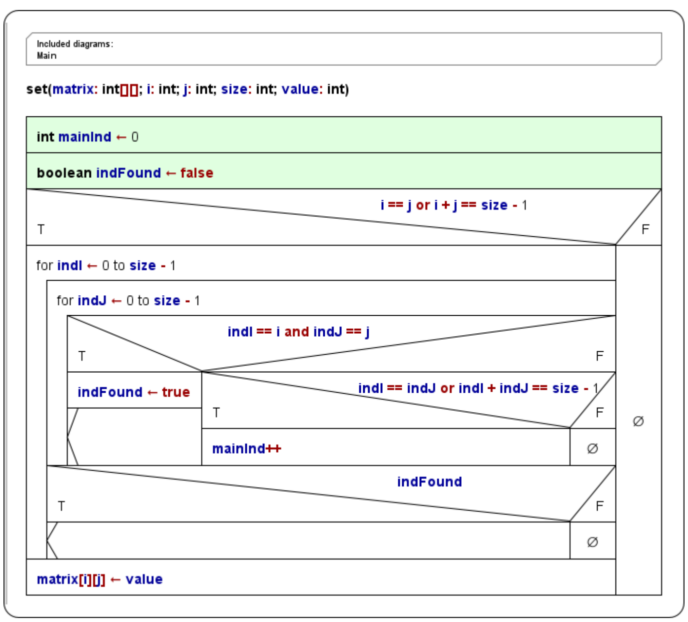
1. Getting an entry

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



2. Setting an entry

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



3. Sum

The sum of matrices a and b (represented by arrays t and u) goes to matrix c (represented by array u), where all of the arrays have to have the same size.

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

4. Multiplication

The product of matrices a and b (represented by arrays t and u) goes to matrix c (represented by array u), where all of the arrays have to have the same size.

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

Testing

Testing the operations (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 in the main and secondary diagonals

    b) Getting and setting an entry outside the main and secondary diagonal

    c) Illegal index, indexing a 0-size matrix

3) Assignment operator

    a) Executing command b=a for matrices a and b (with and without same size), comparing the entries of the two matrices. Then, changing one of the matrices and comparing the entries of the two matrices.

    b) Executing command c=b=a for matrices a, b, and c (with and without same size), comparing the entries of the three matrices. Then, changing one of the matrices and comparing the entries of the three matrices.

    c) Executing command a=a for matrix a.

4) Sum of two matrices, command c:=a+b.

    a) With matrices of different size (size of a and b differs, size of c and a differs)

    b) Checking the commutativity (a + b == b + a)

    c) Checking the associativity (a + b + c == (a + b) + c == a + (b + c))

    d) Checking the neutral element (a + 0 == a, where 0 is the null matrix)

6) Multiplication of two matrices, command c:=a*b.

    a) With matrices of different size (size of a and b differs, size of c and a differs)

    b) Checking the commutativity (a * b == b * a)

    c) Checking the associativity (a * b * c == (a * b) * c == a * (b * c))

    d) Checking the neutral element (a * 0 == 0, where 0 is the null matrix)

    e) Checking the identity element (a * 1 == a, where 1 is the identity matrix)

Testing based on the code (white box testing)

1. Creating an extreme-size matrix (-1, 1, 1000).

2. Generating and catching exceptions.