OOP Daniya Safiullina: Documentation, 1st assignment

Daniya Safiullina

KNBSLM

knbslm@inf.elte.hu

Group 6

March 31, 2024

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

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

Operations

1. Getting an entry
   Getting an entry of the i^{th} column and j^{th} row ($i,j \in [1..n]$): $e:=a[i,j]$.

   Formally: $A : XMatrix(n) \times \mathbb{Z} \times \mathbb{Z} \times \mathbb{Z}$
   $\quad\quad a \quad\quad i \quad j \quad e$
   $Pre = (a=a' \wedge i=i' \wedge j=j' \wedge i,j \in [1..n])$
   $Post = (Pre \wedge e=a[i,j])$

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

2. Setting an entry
   Setting an entry of the i^{th} column and j^{th} row ($i,j \in [1..n]$): $a[i,j] := e$. The entries outside the X matrix cannot be modified($i=j \vee i + j = n + 1$).

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

$\qquad\qquad a \qquad\quad i \quad j \quad e$

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

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] *))*

This operation needs any action only if i=j or i + j = n + 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: $A : XMatrix(n) \times XMatrix(n) \times XMatrix(n)$

$\qquad\qquad a \qquad\qquad\quad b \qquad\qquad\qquad c$

Pre = (a=a' ∧ b=b')

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

For X-type matrix:

∀i∈[1..n]: c[i,i]= a[i,i] + b[i,i] and ∀i,j∈[1..n]: i≠j ∧ (i + j ≠ n + 1) → c[i,j]=0.

4. Multiplication

Multiplication of two matrices: c:=a*b. The matrices have the same size.

Formally: $A : XMatrix(n) \times XMatrix(n) \times XMatrix(n)$

$\qquad\qquad a \qquad\qquad\quad b \qquad\qquad\qquad c$

Pre = (a=a' ∧ b=b')

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

For X-type matrix:

∀i∈[1..n]: c[i,i]= a[i,i] * b[i,i] and ∀i,j∈[1..n]: i≠j ∧ (i + j ≠ n + 1) → c[i,j]=0.

Representation

Only the elements which are in diagonals of X matrix have to be stored($i=j \vee i + j = n + 1$).

$$\begin{array}{cccccc} a11 & 0 & 0 & 0 & \dots & an1 \\ 0 & a22 & 0 & 0 & \dots & an2 \\ 0 & 0 & a33 & 0 & \dots & an3 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a1n & 0 & 0 & 0 & \dots & ann \end{array} \quad \leftrightarrow \quad v=\langle a11,a22,a33,...,ann,a1n,a2(n-1),a3(n-2),...,an1\rangle$$

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

$$a[i,j] = \begin{cases} v[i] \ if \ i = j \\ 0 \ if \ i \neq j \\ 0 \text{ if } i + j \neq n + 1 \end{cases}$$

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 v,$1 \leq i \leq n$, and n stands for the size of the matrix can be implemented as

i=j	
e:=v[i-1]	SKIP

j = n – i + 1	
e:= v[n - i]	SKIP

2. Setting an entry

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

\ i=j /	
v[i-1]:=e	SKIP

\ j = n – i +1 /	
v[n - i]:=e	SKIP

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, 3, 4- size matrix
2. Getting and setting an entry
   a) Getting and setting an entry outside the range
   b) Getting and setting valid entry
3. Executing command a=a for matrix a
4. Sum of two matrices, command c:=a+b.
   a) Checking summation of matrices with different sizes

b) Checking summation of matrices with the same sizes

c) Checking if result of matrix is not null and size is correct

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

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

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

   a) Checking multiplication of matrices with different sizes

   b) Checking multiplication of matrices with the same sizes

   c) Checking if result of matrix is not null and size is correct

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

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