Jamil Shahverdiyev

CDWCRZ

cdwcrz@inf.elte.hu

Group 5

Documentation to 1st assignment.

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)

Diagonal matrix type

Set of values

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

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

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

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

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

2. Sum

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

Formally: $A = Diag(n) \times Diag(n) \times Diag(n)$

Pre = $(a=a' \wedge b=b')$

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

3. Multiplication

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

Formally: $A = Diag(n) \times Diag(n) \times Diag(n)$

Pre = $(a=a' \wedge b=b')$

Post = $(Pre \ \wedge \ \forall i,j \in [1..n]: c[i,j]= \sum_{k=1..n} a[i,k] * b[k,j])$

Representation

Only the diagonals of the n×n matrix has to be stored.

V = < A[1,1], A[2,2], A[3,3], A[4,4], A[5,5], A[1,5], A[2,4], A[4,2], A[5,1]>

Lets represent “A” matrix:

A[1,1]	0	0	0	A[1,5]
0	A[2,2]	0	A[2,4]	0
0	0	A[3,3]	0	0
0	A[4,2]	0	A[4,4]	0
A[5,1]	0	0	0	A[5,5]

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

If n is even:

A[i,j] = v[i] if (i = j or i + j - 1 = n)

A[i,j] = 0 if (i ≠ j or i + j - 1 ≠ n)

If n is odd:

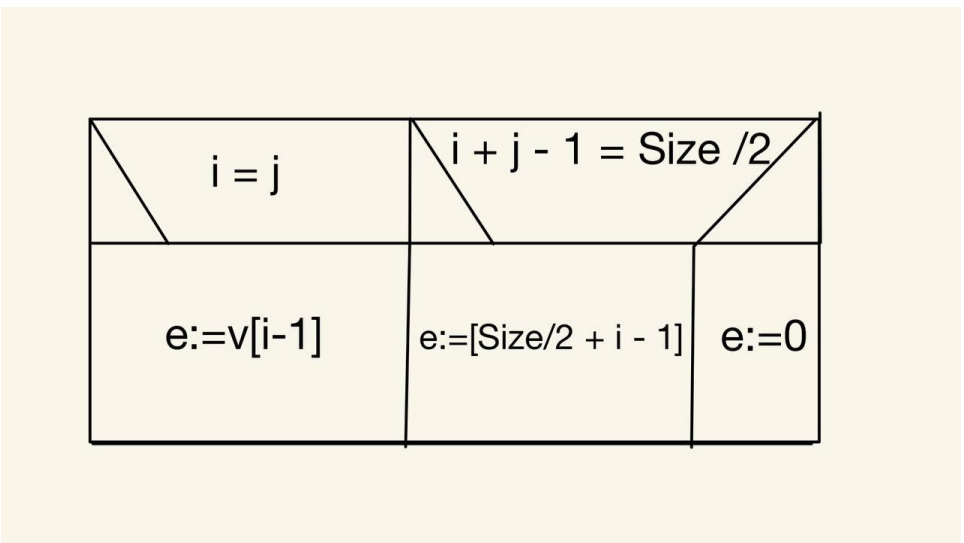
A[i,j] = v[i] if (i = j or i + j - 2= n)

A[i,j] = 0 if (i ≠ j or i + j - 2 ≠ n)

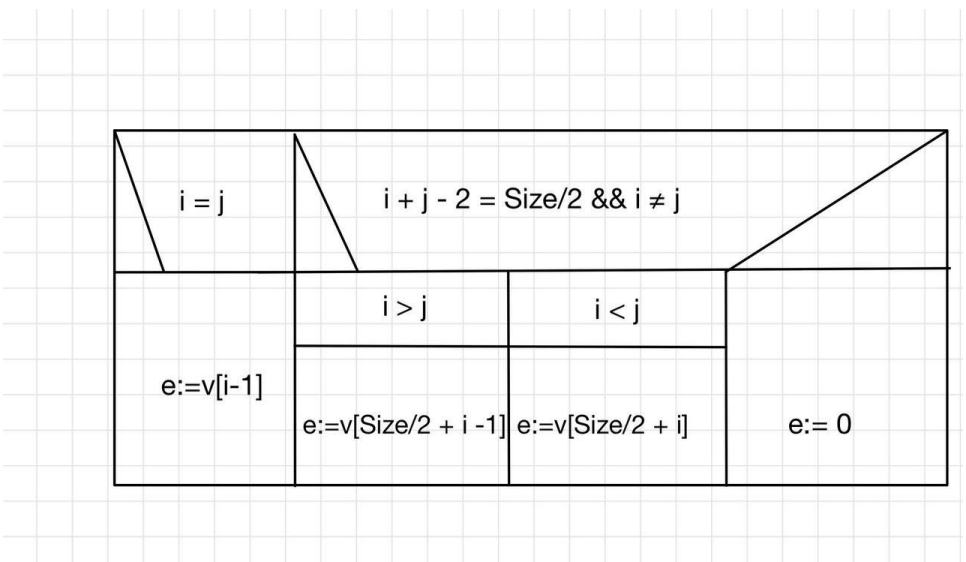
Implementation[1]

1. Getting an entry of the *i*-th column and *j*-th row (i,j ∈ [1..n]) e:= a[i,j] where matrix is represented by v, 1 ≤ i ≤ n, and n stands for the size of the matrix can be implemented in two ways:

If n is even number:

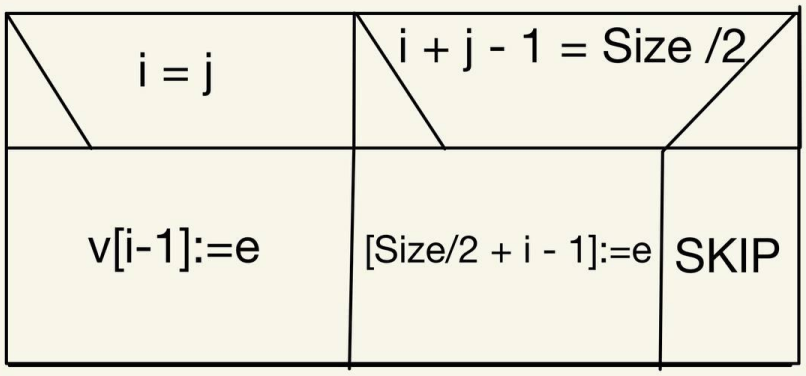


If n is odd number:

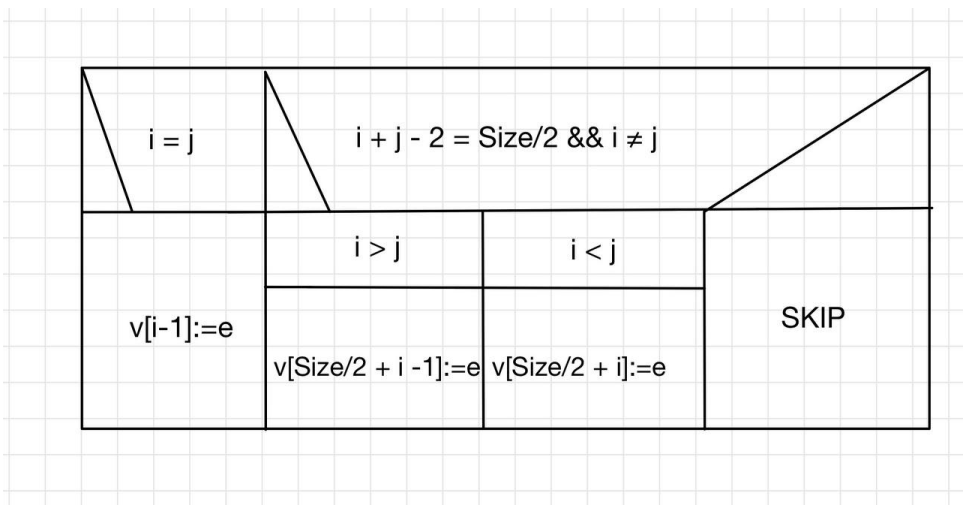


2. Setting an entry of the *i*-th column and *j*-th row (i,j ∈ [1..n]) e:= a[i,j] where matrix is represented by v, 1 ≤ i ≤ n, and n stands for the size of the matrix can be implemented in two ways: (**NOTE**: I have the setting part because of the multiplication!)

If n is even number:



If n is odd number:



3. 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 must be the same size:

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

4. 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 must be the same size:

$\forall i,j \in [0..n-1]: c[i,j]= \sum_{k=1..n} a[i,k] * b[k,j]$

Testing

Testing the operations (black box testing)

1) Creating and testing matrices of different sizes

A) -1, 0, 1, 2, 5 , 1000-size matrix

2) Getting and setting entry
   A) Getting and setting entry inside the diagonal
   B) Getting and setting entry outside the diagonal
   C) Illegal index, index out of range
3) Copy Constructor

A) Creating matrix b based on matrix a, comparing the entries of the two matrices. Then, changing one of the matrices and comparing the entries of the two matrices.

4) Assignment operator

A) Checking the command a = b with and without same size, changing entry of matrix b and then comparing two matrices

B) Executing command c=b=a for matrices a, b, and c, comparing the entries of the three matrices. Then, changing one of the matrices and comparing the entries of the three matrices.

C) Checking command a=a for matrix a.

5) Summation of two matrices, c:= a + b
   A) With matrices of different size (size of a and d differs, size of b and d 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 d differs, size of b and d 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) Generating matrices of extraordinary size(-1, 0, 1, 1000).
2) Generating and catching exceptions.