MD Jaodun Muntasir
BVLLR5
bvllr5@inf.elte.hu
Group 4

1. assignment/1. task

2nd April 2023

## 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 same-colored 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).**

## Chessboard matrix type

### Set of values

**Cmatrix(n) = { a $\mathbb{Z}^{n \times n}$ i,j [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 [1..n]): e:=a[i,j].**

**Formally: A : Cmatrix(n) × $\mathbb{Z}$ × $\mathbb{Z}$ × $\mathbb{Z}$**
**a i j e**
**Pre = ( a=a' i=i' j=j' i,j [1..n] )**
**Post = ( Pre e=a[i,j] )**

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

**Formally: A = Cmatrix(n) × Cmatrix(n) × Cmatrix(n)**
**a b c**
**Pre = ( a=a' b=b')**
**Post = ( Prei,j [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:** $A = Cmatrix(n) \times Cmatrix(n) \times Cmatrix(n)$
$a \quad b \quad c$
$Pre = ( a=a' \quad b=b')$
$Post = ( Pre \quad i,j \ [1..n]: c[i,j]= \sum_{k=1..n} a[i,k] * b[k,j])$

### Representation

**Only every second element of the n×n matrix has to be stored.**

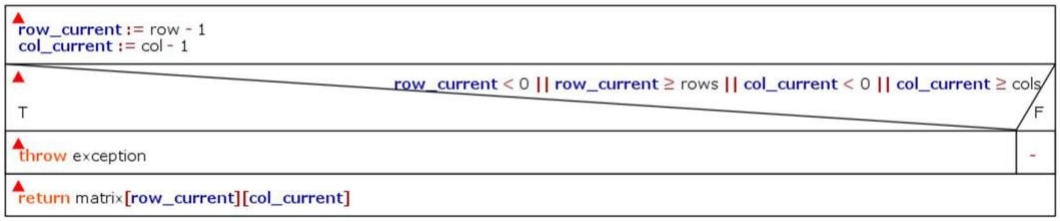
$$a = \begin{matrix} 0 & a_{12} & 0 & \dots & 0 \\ a_{21} & 0 & a_{23} & \dots & a_{2n} \\ 0 & a_{32} & 0 & \dots & 0 \\ a_{41} & 0 & a_{43} & \dots & a_{nn} \end{matrix}$$

## Implementation[1]

### 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**

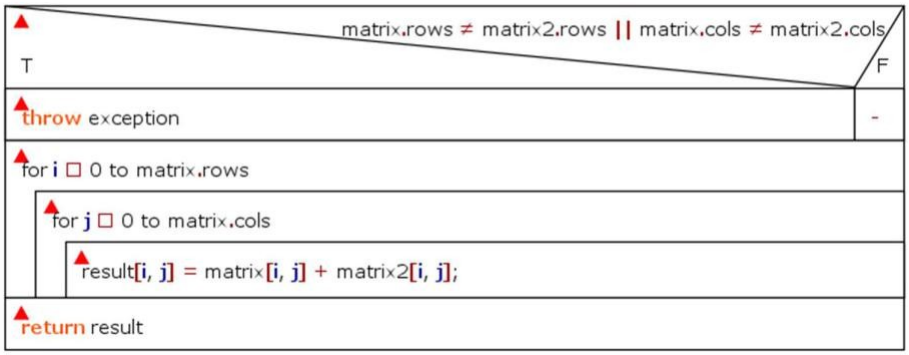
GetValue




### 2. Sum

**The sum of matrices a and b (represented by chessboardmatrix matrix and matrix2) goes to chessboardmatrix result, where all of the matrices have to have the same size.**

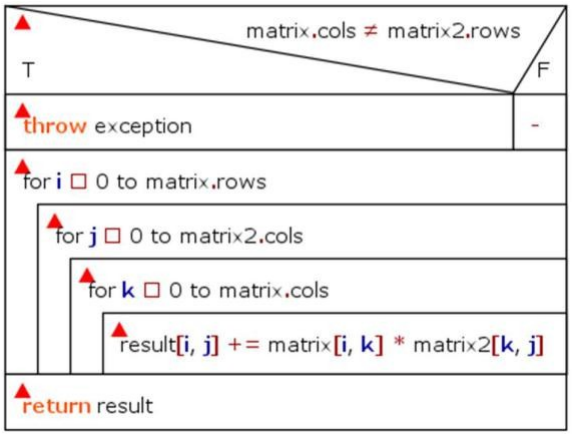
Addition




### 3. Multiplication

**The product of matrices a and b (represented by chessboard matrices matrix and matrix2) goes to chessboardmatrix result, where a's column and b's row is same.**

Multiplication



## Testing

**Testing the operations (black box testing)**

1) **Creating, printing, and checking matrices of different size.**
   a) **Creating a 3x3 -size matrix**
   b) **Printing a 3x3 -size matrix**
   c) **Reading the printed matrix and checking whether it's the same one or not**

2) **Creating and getting an entry**
   a) **Creating a 4x4 -size matrix**
   b) **Getting an entry by indexing the row and column number**

3) **Sum of two matrices, command c := a+b.**
   a) **With matrices of same size**
   b) **With matrices of different size**

4) **Multiplication of two matrices, command c := a*b.**
   a) **With matrices where column number of the first matrix is equal to the row number of the second matrix.**
   b) **With matrices where column number of the first matrix doesn't equal to the row number of the second matrix.**

**Testing based on the code (white box testing)**

1. **Generating and catching exceptions.**