

Mohamed Hamed. Assignment 1. task 4

DG1EWF

Group 11

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 b_1 and b_2 , where $1 \leq b_1, b_2 \leq n-1$ and $b_1 + b_2 = n$ (in the example, $b_1 = 2$ and $b_2 = 4$). 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).

Set of values

$BM(n, b_1, b_2) = \{ a \in \mathbb{Z}^{n \times n} \mid i, j \in [1..n]: (i, j) \text{ not within } B1 \text{ or } B2 \rightarrow a[i, j] = 0 \}$
with $B1$ the block matrix with values on the $b_1 \times b_1$ block starting from $(0, 0)$
with $B2$ the block matrix with values on the $b_2 \times b_2$ block ending in (n, n)

Operations:

1. Creating an empty matrix

$A : BM \times \mathbb{Z} \times \mathbb{Z}$

$a \quad b_1 \quad b_2$

$Pre = (a = a' \text{ and } b_1 = b_1' \text{ and } b_2 = b_2')$

$Post = (Pre \text{ and } a(b_1, b_2))$

2. Creating a Custom Filled Matrix

$A : BM \times \mathbb{Z} \times \mathbb{Z} \times \mathbb{Z}^*$

$a \quad b_1 \quad b_2 \quad list$

$Pre = (a = a' \text{ and } b_1 = b_1' \text{ and } b_2 = b_2' \text{ and } list = list')$

$Post = (Pre \text{ and } a(b_1, b_2, list))$

3. Summation

$A : BM \times BM \times BM$

$a \quad b \quad c$

$Pre = (a = a' \text{ and } b = b')$

$Post = (Pre \text{ and } c = a + b)$

4. Multiplication

$A : BM \times BM \times BM$

$a \quad b \quad c$

$Pre = (a = a' \text{ and } b = b')$

$Post = (Pre \text{ and } c = a * b)$

Representation

size, $b_1_size : \mathbb{Z}$

block1, block2, matrix : Matrix ($\mathbb{Z}^{n \times n}$)

Implementation

1. Creating an Empty Matrix

if ($b_1 \leq 0 \parallel b_2 \leq 0$) then error

```

else
    size := b1 + b2
    block1 := new Matrix(b1);
    block2 := new Matrix(b2);
    matrix := new Matrix(size);
endif

```

2. Creating a Custom Filled Matrix

```

if (b1 <= 0 || b2 <= 0 || b1 * b1 + b2 * b2 != list.Count) then error
else
    block1 := new Matrix(list.GetRange(0, b1 * b1));
    block2 := new Matrix(list.GetRange(b1 * b1, b2 * b2));

    matrix := new Matrix(size);
    matrix := Matrix.Add(matrix, block1);
    matrix := Matrix.Add(matrix, block2, b1);
endif

```

3. Summation

```

if (m1.size != m2.size) then error
else if (m1.b1_size != m2.b1_size) then error
else
    BlockMatrix result = new BlockMatrix(m1.b1_size, m1.size - m1.b1_size);

    Matrix temp := Matrix.Add(m1.block1, m2.block1);
    result._matrix := Matrix.Add(result._matrix, temp);

    temp := Matrix.Add(m1.block2, m2.block2);
    result.matrix := Matrix.Add(result.matrix, temp, result.b1_size);
endif

```

4. Multiplication

```

if (m1.size != m2.size) then error
else if (m1.b1_size != m2.b1_size) then error
else
    BlockMatrix result = new BlockMatrix(m1.b1_size, m1.size - m1.b1_size);

    Matrix temp := Matrix.Multiply(m1.block1, m2.block1);
    result._matrix := Matrix.Add(result._matrix, temp);

    temp := Matrix.Multiply(m1.block2, m2.block2);
    result.matrix := Matrix.Add(result.matrix, temp, result.b1_size);
endif

```

Testing

Testing the operations (black box testing)

1. **Multiplying Matrices**

- a) Multiplying 2 Block Matrices and comparing their result against the correct result

2. **Adding Matrices**

- a) Adding 2 Block Matrices and comparing their result against the correct result
- b) Illegally adding 2 Block Matrices with different dimensions and awaiting exception to be thrown
- c) Illegally adding 2 Block Matrices with different distribution and awaiting exception to be thrown

3. **Creating Custom Filled Matrix**

- a) Creating Custom Filled Matrix Illegally with the number of elements in the list not enough to satisfy the block distribution
- b) Creating Custom Filled Matrix Illegally with the block distribution more not matching a valid list of elements
- c) Creating Custom Filled Matrix Illegally with a null block distribution and a valid list of elements