자료구조론 HW1

김정선 교수님

소프트웨어융합대학 소프트웨어학과

2017012197 여채린

public class ArraySparseMatrix implements SparseMatrix {

    public static final int DEFAULT\_CAPACITY = 1024;

    private int rowCount;

    private int columnCount;

    private int elemCount;

    private Entry[] elements = new Entry[0];

    public ArraySparseMatrix(int rowCount, int columnCount, int capacity) {

        elements = new Entry[capacity];

        this.rowCount = rowCount;

        this.columnCount = columnCount;

        this.elemCount = 0;

    }

    public ArraySparseMatrix(int rowCount, int columnCount) {

        this(rowCount, columnCount, DEFAULT\_CAPACITY);

    }

    /\*

     \* This routine simulates reading from files using two-dimensional matrix.

     \*/

    public static SparseMatrix create(double[][] aMatrix, int rowCount, int columnCount, int elemCount) {

        ArraySparseMatrix matrix = new ArraySparseMatrix(rowCount, columnCount, elemCount);

        int nonZeroCount = 0;

        for (int i = 0; i < aMatrix.length; i++)

            for (int j = 0; j < aMatrix[i].length; j++) {

                if (Double.compare(aMatrix[i][j], 0.0) != 0) {

                    matrix.put(new Entry(i, j, aMatrix[i][j]));

                    nonZeroCount++;

                }

            }

        if (nonZeroCount != elemCount)

            throw new IllegalStateException("Non zero count doesn't match");

        return matrix;

    }

    @Override

    public SparseMatrix transpose() {

        /\*

         \*   Your code goes here

         \*/

        return null;

    }

    @Override

    public SparseMatrix add(SparseMatrix other) {

        if (this.rowCount != other.getRowCount() || this.columnCount != other.getColumnCount()) {

            throw new IllegalArgumentException("Matrix size doesn't match");

        }

    double resultMatrix[][] = new double[this.rowCount][this.columnCount];//결과 matrix

    ArraySparseMatrix m2 = (ArraySparseMatrix) other;//other..m2갖다쓰기!!

    int resultValue = 0;

    for(int i = 0; i < elements.length; i++) {

    resultMatrix[this.elements[i].row][this.elements[i].col] += elements[i].value;

    }

    for(int i = 0; i < m2.elements.length; i++) {

    resultMatrix[m2.elements[i].row][m2.elements[i].col] += m2.elements[i].value;

  }

    for(int i =0; i< this.rowCount; i++) {

    for(int j = 0 ; j< this.columnCount ; j++) {

    if(resultMatrix[i][j] != 0)

    resultValue++;

  }

    }

    SparseMatrix outputSparseMatrix = ArraySparseMatrix.create(resultMatrix ,this.rowCount,this.columnCount, resultValue);

    ArraySparseMatrix outputASparseMatrix = (ArraySparseMatrix) outputSparseMatrix;

    return outputASparseMatrix;

    }

    public void put(Entry entry) {

        elements[elemCount++] = entry;

    }

    @Override

    public SparseMatrix multiply(SparseMatrix other) {

        throw new IllegalStateException("Not implemented");

    }

    @Override

    public int getRowCount() {

        return rowCount;

    }

    @Override

    public int getColumnCount() {

        return columnCount;

    }

    @Override

    public int getElemCount() {

        return elemCount;

    }

//    @Override

//    public Entry getElements(int n) {

//    return elements[n];

//    }

    @Override

    public boolean equals(Object obj) {

        if (this == obj) return true;

        if (!(obj instanceof ArraySparseMatrix)) return false;

        ArraySparseMatrix other = (ArraySparseMatrix) obj;

        if (rowCount != other.rowCount || columnCount != other.columnCount || elemCount != other.elemCount)

            return false;

        for (int i = 0; i < elements.length; i++) {

            if (!elements[i].equals(other.elements[i])) return false;

        }

        return true;

    }

    @Override

    public String toString() {

        StringBuilder builder = new StringBuilder();

        builder.append(rowCount + "\t" + columnCount + "\t" + elemCount + "\n");

        for (int i = 0; i < elemCount; i ++)

            builder.append(elements[i] + "\n");

        return builder.toString();

    }

}