

מעבדה 4. נושא: מטריצות דלילות

הגשה

קבוצה של יום ב': 23.11.20

קבוצה של יום ד': 25.11.20

(הגשה בזוגות)

עבודה נעימה!

מבוא:

במעבדה זו נלמד סוג נוסף של מטריצות מיוחדות - מטריצות דלילות ונראה כיצד ניתן לממש אותן בעזרת רשימה מקושרת חד-כיוונית.

תיאור:

1. מטריצה ריבועית $n \times n$ בה רוב האיברים שווים לאפס, נקראת **מטריצה דלילה**. למשל:

0 1 0 4

0 0 0 0

0 0 0 0

0 6 0 0

זו מטריצה דלילה בגודל 4×4 . במטריצה זאת מספר האיברים השונים מאפס שווה ל-3.

לאפס נקרא האיבר הנפוץ.

מקובל להגדיר מטריצה דלילה ע"י כך שמספר האיברים השונים מאפס r הרבה יותר קטן מ $n \times n$, כלומר

$$r \ll n \times n$$

2. גם מטריצה כזאת ניתן לראות כמטריצה דלילה:

3 1 3 4

3 3 3 3

3 3 3 3

3 6 3 3

במקרה זה רוב האיברים של המטריצה הנ"ל שווים ל-3 ויש מעט איברים השונים מ-3

כלומר האיבר הנפוץ כאן הוא 3.

1). **מימוש סטנדרטי** , באמצעות מערך דו-מימדי , מאפשר ביצוע פעולות בסיבוכיות:

- $O(1)$ היא $get(i,j)$

- $O(1)$ היא $put(i,j,x)$

- אך סיבוכיות המקום הנה גבוהה $O(n^2)$.

2). נציע מימוש חלופי באמצעות **רשימה מקושרת חד-כיוונית**:

לכל איבר השונה מהאיבר הנפוץ, נחזיק צומת בעל 3 שדות: אינדקס i , אינדקס j , וערך האיבר.

לדוגמא, עבור המטריצה המופיעה בסעיף 1, ניתן להחזיק רשימה מקושרת הבאה

$[1,2,1] \rightarrow [1,4,4] \rightarrow [4,2,6]$

ולמטריצה כללית המופיעה בסעיף 2, נציע רשימה:

$[-1,-1,3] \rightarrow [1,2,1] \rightarrow [1,4,4] \rightarrow [4,2,6]$

כאשר האיבר הראשון ברשימה מחזיק את האיבר הכללי שהוא 3 ושאר האיברים מאורגנים כמו בדוגמא הקודמת.

במימוש זה, **הסיבוכיות** תהיה כדלקמן:

סיבוכיות מקום: $O(r)$

- סיבוכיות $get(i,j)$ היא $O(r)$ (במקרה הגרוע יש לעבור על כל הרשימה).

- סיבוכיות $put(i,j,x)$ היא $O(r)$ (יש לבדוק האם האיבר כבר קיים ברשימה).

המימוש המוצע חוסך הרבה מקום.

שלב 1.

ראשית, נצור מחלקה של איברי המטריצה (ENTRY). כל איבר יכיל שני אינדקסים (i,j) של המיקום והערך (value). להלן המחלקה:

```
public class SparseMatrixEntry {
    private double value; // Sparse Matrix element
    private int i;        // Index i
    private int j;        // Index j

    public SparseMatrixEntry(double val, int i, int j) { }

    public int getI() { } //
    public void setI(int i) { }

    public int getJ() { }
    public void setJ(int j) { }

    public double getValue() { }
    public void setValue(double newVal) { }
}
```

ממש את המחלקה SparseMatrixEntry

שלב 2.

צור מנשק בשם **Matrix** (או השתמש בזה של מעבדה 2) התומך בפעולות הבאות

int getSize()

Precondition: None

Postcondition: returns the number of rows (=number of columns=size) of the matrix.

double get(int i, int j)

Precondition: $0 \leq i, j < N$

Postcondition: returns the value of the element at the place (i,j).

void put(int i, int j, double x)

Precondition: $0 \leq i, j < N$

Postcondition: update the value of the element at the place (i,j) to be equal to x.

void transpose()

Precondition: None.

Postcondition: configure the matrix to be transpose matrix.

Think about a "smart" implementation of $O(1)$

void multByConstant (int C)

Precondition: None.

Postcondition: returns the matrix, where every element is multiplied by integer constant C. **Do it in $O(1)$!**

שלב 3.

כתוב מחלקה בשם **SparseMatrix** המממשת את המנשק הנ"ל בעזרת רשימה מקושרת חד-כיוונית שמימשת במעבדה 3 ומכילה:

1). **בנאי**

SparseMatrix (int size, double defaultValue);

הבונה מטריצה שבה האיבר הנפוץ הוא defaultValue, וגודל המטריצה (מספר השורות והעמודות) הוא sizeXsize.

2). מממשת **toString** משלה (עבור המטריצה בצורתה הטבעית בגודל sizeXsize).

שלב 4. בדיקות

א). צור מחלקה חדשה בשם **TestSparseMatrix.java** בה מטריצות דלילות שונות בגדלים שונים ובדוק את עבודתה של המחלקה SparseMatrix.

שלב 5. ועוד שתי פעולות לסיום

ב). תוסיף למחלקה SparseMatrix פעולות אריתמטיות לחיבור וחסור של שתי מטריצות דלילות. יש לבצע את הפעולות באופן יעיל!!! ולציין את הסיבוכיות.

הגשה:

יש להגיש

SparseMatrixEntry.java, SparseMatrix.java, TestSparseMatrix.java

ערעורים:

יש להפנות לד"ר תמר צמח בלבד ע"י שליחת מייל tamar.zemach@yahoo.com **תוך שבוע** ממועד פרסום הציונים. בכותרת המייל יש לציין: "ערעור מעבדה מס' X ע"י +ת.ז. של הסטודנטים". יש לקחת בחשבון שבעת הערעור העבודה נבדקת מחדש וכתוצאה מהערעור ציון העבודה עלול להשתנות (יכול לעלות או לרדת). התשובה לערעור תהיה סופית ולא ניתנת לערעור נוסף.