

## מעבדה 2. נושא: מטריצות מיוחדות

קבוצות יום ב - 09.11.20

קבוצה יום ד - 11.11.20

(הגשה בזוגות)

יש לקרוא היטב לפני תחילת העבודה!

### מבוא:

מטריצה היא מבנה המכיל אוסף של מספרים. בדרך כלל כדי להחזיק  $m \times n$  מספרים, נדרשת מטריצה בעלת  $m$  שורות ו  $n$  עמודות.

- המימוש הקלאסי של מטריצה הוא מערך דו-מימדי בגודל  $m \times n$ .

- הפעולות הבסיסיות הדרושות לעבודה עם מטריצה הן:

1.  $get(i,j)$  המחזירה איבר שנמצא במקום  $(i, j)$ .

2.  $put(i,j,x)$  המעדכנת את האיבר הנמצא במקום  $(i, j)$  להיות  $x$ .

- סיבוכיות הזמן של שתי פעולות אלה  $O(1)$ .

סיבוכיות המקום של המבנה  $O(m \times n)$ .

אולם, כאשר מטריצה היא בעלת מבנה מיוחד, ניתן להציע מימוש חכם יותר וכך לשפר את הסיבוכיות.

### מטרות:

במעבדה זו נכיר מימושים של מטריצה מיוחדת - מטריצה אלכסונית מסוג 2

**תיאור:** מטריצה ריבועית  $n \times n$  בה הערכים בכל אלכסון מסוג 2 שווים, נקראת מטריצה אלכסונית מסוג 2. למשל:

6	5	3	4
5	3	4	2
3	4	2	7
4	2	7	1

זו מטריצה אלכסונית מסוג 2 בגודל  $4 \times 4$ . שים לב שמספר האלכסונים מסוג 2 במטריצה זו שווה ל 7.

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

-  $get(i,j)$  היא  $O(1)$ .

-  $put(i,j,x)$  היא  $O(n)$ , כי צריך לעדכן את כל אברי האלכסון.

2). נציע מימוש חלופי באמצעות מערך חד-מימדי בגודל  $2n-1$ .

נשים לב כי מספר האלכסונים השונים במערך עומד על  $2n-1$ . לכל אלכסון נגדיר תא אחד במערך. למשל, עבור המטריצה המופיעה לעיל, נחזיק מערך בגודל 7 ובו האיברים:

6 5 3 4 2 7 1

– סיבוכיות  $get(i,j)$  היא  $O(1)$ .

– סיבוכיות  $put(i,j,x)$  היא  $O(1)$ .

כך שחסכנו גם במקום וגם בזמן! המשימה היחידה היא לתרגם נכון את האינדקסים של המטריצה הנתונה אל המטריצה החד-מימדית ובחזרה.

רמז: חשוב מה מאפיין כל אלכסון מסוג 2 (התבונן באינדקסים :  $i$  ו  $j$ ).

## משימות המעבדה

### משימה 1

כתוב ממשק (interface) בשם **Matrix** עם הפעולות הבאות והקבוע  $MAX\_SIZE$  שערכו 100.

**int getSize()**

return the number of rows (=number of columns= $N$ ) of the matrix.

**double get (int i, int j)**

*Precondition:*  $0 \leq i, j \leq N$ .

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

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

*Precondition:*  $0 \leq i, j \leq n$

*Postcondition:* updates 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. **Do it in  $O(1)$ !**

**void multByConstant (int C )**

*Precondition:*  $C > 0$ .

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

### משימה 2

כתוב מחלקה בשם **DiagonalMatrix** המממשת את הממשק הנ"ל כמטריצה אלכסונית (כלומר, אם משנים איבר במטריצה, אז כל האיברים באותו אלכסון גם משתנים) ומכילה את הבנאים הבאים:

**DiagonalMatrix(int size)**

*Precondition:*  $size > 0$

*Postcondition:* Initializes a  $size * size$  diagonal matrix using an array of length  $2 * size - 1$ .

## DiagonalMatrix( )

Initializes an MAX\_SIZE \* MAX\_SIZE diagonal matrix

## String toString( )

Postcondition: Returns the matrix in its natural N\*N form as a string (with \t between entries of the same row and \n between rows).

## משימה 3

השתמש ב tokenizer וכתוב מחלקה בשם **TestDiagMatrix** התומכת בפעולות הבאות ובודקת את עבודתה של DiagonalMatrix:

פקודה	פעולה
Put i j x	מעדכן איבר במקום ( i, j ) להיות x.
Get i j	מחזיר ומדפיס איבר הנמצא במקום ( i, j ).
PrintM	מדפיס את המטריצה בתצוגה הרגילה
MultCons C	הכפל את כל אברי המטריצה ב C
TransM	בצע transpose של מטריצה
Quit	סיים את התוכנית

## הגשת המעבדה:

יש להגיש 3 קבצים **Matrix.java, DiagonalMatrix.java, TestDiagMatrix.java**

## ערעורים:

יש להפנות לד"ר תמר צמח בלבד ע"י שליחת מייל [tamar.zemach@yahoo.com](mailto: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.