

מעבדה 2. נושא: מטריצות מיוחדות

תאריך הגשה: TBD

יש לקרוא היטב לפני תחילת העבודה !

מבוא:

מטריצה היא מבנה דו-מימדי המכיל אוסף של מספרים.

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

- הפעולות הבסיסיות הדרושות לעבודה עם מטריצה הן:

- $get(i, j)$ המחזירה את האיבר שנמצא במקום (i, j) .
- $set(i, j, x)$ המעדכנת את האיבר הנמצא במקום (i, j) להיות x .

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

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

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

מטרות:

במעבדה זו נכיר מימושים של מטריצה מיוחדת - מטריצה אלכסונית

תיאור: מטריצה ריבועית $n \times n$ בה הערכים בכל אלכסון שווים, נקראת מטריצה אלכסונית. למשל המטריצה הבאה היא מטריצה אלכסונית 4×4 . שים לב שמספר האלכסונים במטריצה זו הוא 7.

2134
5213
6521
7652

סיבוכיות המימוש סטנדרטי:

- $get(i, j)$ בזמן $O(1)$.
- $set(i, j, x)$ בזמן $O(n)$ כי צריך לעדכן את כל אברי האלכסון.

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

- נשים לב כי מספר האלכסונים השונים במערך הוא $2n - 1$. עבור כל אלכסון נקצה תא אחד במערך. למשל, עבור המטריצה המופיעה לעיל, נחזיק מערך בגודל 7 ובו האיברים: 7652134.
- סיבוכיות $get(i, j)$ היא $O(1)$.
- סיבוכיות $set(i, j, x)$ היא $O(1)$.

כך שחסכנו גם במקום וגם בזמן!

המשימה היחידה היא לתרגם נכון את האינדקסים של המטריצה הנתונה אל המטריצה החד-מימדית ובחזרה.

(1) כתבו ממשק בשם Matrix עם הפעולות הבאות והקבוע MAX_SIZE שערכו 100.

double get(int i, int j)

Precondition: $1 \leq i, j \leq n$

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

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

Precondition: $1 \leq i, j \leq n$.

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

void transpose()

Postcondition: replace the current matrix with its transpose.

Matrix getTranspose()

Postcondition: returns a new matrix which equals to the transpose of the current matrix.

(2) כתבו 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

בנוסף, המחלקה מכילה את השיטה הבאה:

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).

יש לממש את השיטה transpose בסיבוכיות זמן של $O(n)$.