

מעבדה 2. נושא: מטריצות מיוחדות

תאריך הגשה: 30/03/2023 בשעה 23:00 (בזוגות)

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 איברי האלכסון.

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

- נשים לב כי מספר האלכסונים השונים במעריך הוא $2n - 1$. עבור כל אלכסון נקצה תא אחד במעריך. למשל, עבור המטריצה המופיעה לעיל, נחזיק מעריך 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** הממשת 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 $O(1)$, ו- $getTranspose$ ב- $O(size)$

במקרה שתנאי ה-precondition לא מתקיים (למשל מספר שורה קטן מ-1 או גדול ממספר השורות) יש לזרוק חריגת זמן ריצה מתאימה.

סדר העבודה ופרטים טכניים

- שלפת הפרויקט של lab02 מתוך GITHUB:
 - אם אין לכם גישה לפרויקט שהורדתם מ GITHUB במעבדה הראשונה יש לבצע שליפה מחדש לפי ההוראות במעבדה הראשונה.
<https://github.com/michalHorovitz/DSLlab2023Public>
 - אם יש לכם גישה לפרויקט שהורדתם מ GITHUB במעבדה הראשונה אז בצעו:
 - קליק על שם הפרויקט.
 - עכבר ימני
 - Team-->Pull
 - File-->Import->Git->Projects From Git->Existing Local Repository
- אם אתם עובדים ב VDI, מומלץ לשנות את המיקום המוצע לפרויקט בתיקייה כלשהי בכוון H.

פורמט קובץ ההגשה ובדיקתו:

פורמט: יש להגיש קובץ ZIP בשם

46_lab02_123456789_987654321.zip

(כמובן, יש להחליף את המספרים עם מספרי ת.ז. של המגישים).

על הקובץ להכיל את כל קבצי ה JAVA שכתבתם. שימו לב: הקובץ לא יכיל את התיקיה שבה הקבצים נמצאים, רק את הקבצים עצמם (אם לא ברור מה ההבדל, ראו סרטון הדגמה מטה).

ניתן ליצור את הקובץ בשרת המכללה ע"י הפקודה הבאה:

```
zip ~/46_lab02_123456789_987654321.zip *.java
```

שיש לתת כאשר אתם נמצאים בתיקייה src של הפרויקט. הקובץ יוצר בתיקיית הבית שלכם.

בדיקה: בדקו את הקובץ שיצרתם בתוכנת הבדיקה בקישור:

<https://csweb.telhai.ac.il/>

ראו [סרטון הדגמה](#) של השימוש בתוכנת הבדיקה.

חשוב !!!

בדיקת ההגשות תבוצע ברובה ע"י תוכנית הבדיקה האוטומטית הנ"ל. תוצאת הבדיקה תהייה בעיקרון זהה לתוצאת הבדיקה הנ"ל שאתם אמורים לערוך בעצמכם. כלומר, אם ביצעתם את הבדיקה באתר החוג, לא תקבלו הפתעות בדיעבד. אחרת, ייתכן שתרגיל שעבדתם עליו קשה יפסל בגלל פורמט הגשה שגוי וכו'. דבר שהיה ניתן לתקנו בקלות אם הייתם מבצעים את הבדיקה. היות ואין הפתעות בדיעבד, לא תינתן אפשרות של תיקונים, הגשות חוזרות וכד'.

הגשה שלא מגיעה לשלב הקומפילציה תקבל ציון 0.

הגשה שלא שמתקמפלת תקבל ציון נמוך מ- 40 לפי סוג הבעיה.

הגשה שמתקמפלת תקבל ציון 40 ומעלה בהתאם לתוצאות הריצה, ותוצאת הבדיקה הידנית של הקוד (חוץ ממקרה של העתקה).

תכנית הבדיקה האוטומטית מכילה תוכנה חכמה המגלה העתקות. מקרים של העתקות יטופלו בחומרה.