

# תרגיל בית מספר 5: מטלת סיכום

**נושא: מיון רב-מימדי גנרי בשירות סימולציה נומרית**

**דדליין: יום ה', 17/01/2019, 23:59**

**הגשה אפשרית בזוגות**

**בהצלחה רבה!**

## תיאור התרגיל

תרגיל זה עוסק בסימולציה של מיון רב-מימדי, ובמימושו אתם מוזמנים להפעיל את כל כלי התכנות שרכשתם עד כה במהלך הקורס. החלק הראשון עוסק בתכנות גנרי של אלגוריתמים למיון, והחלק השני בסימולציה נומרית העושה שימוש באלגוריתמים אלו.

## חלק א': מיון רב-מימדי בקוד תבנית

נגדיר שני מרחבים מתמטיים הנדרשים לתיאור האובייקטים בתרגיל. יהי מרחב המשתנים  $X$  מממד  $n$  ויהי מרחב המטרות  $\Phi$  מממד  $m$ . נניח שלכל וקטור משתנים  $\vec{x}^{(k)} \in X$  קיים מיפוי מטרות  $\vec{f}^{(k)} \in \Phi$ :

$$\vec{f}^{(k)}(\vec{x}^{(k)}) = \{f_1^{(k)}(\vec{x}^{(k)}), \dots, f_m^{(k)}(\vec{x}^{(k)})\}$$

כמו כן, נגדיר *חס סדר חלקי* בין וקטורי מטרה, הנקרא *חס שליטה*, באופן הבא – נאמר כי וקטור מטרה  $\vec{f}^{(1)}$  **שולט** על וקטור מטרה  $\vec{f}^{(2)}$  אם ורק אם מתקיימים שני התנאים הבאים:

$$\forall i f_i^{(1)} \leq f_i^{(2)} \quad \wedge \quad \exists j f_j^{(1)} < f_j^{(2)}$$

בהינתן קבוצה של וקטורי מטרה, **דרגה** של וקטור מטרה בודד מוגדרת כמספר האיברים בקבוצה ששולטים עליו. כלומר, בהינתן קבוצת וקטורים בגודל  $N_p$ , דרגה של וקטור הינה מספר שלם המקיים

$$\text{rank}(\vec{f}^{(k)}) \in \{0, 1, \dots, N_p - 1\}$$

לבסוף, נגדיר אובייקט מטיפוס "אלמנט" באמצעות וקטור משתנים, וקטור המטרות התואם לו, וכן הדרגה שלו ביחס לקבוצת וקטורי המטרה הנוכחית. אתם חופשיים בתכנון ממשק טיפוס "אלמנט".

**בחלק זה של התרגיל עליכם לממש באמצעות קוד תבנית את שלושת האלגוריתמים הגנריים הבאים:**

1. אלגוריתם לדירוג רב-מימדי בשם **ParetoRanking**: אלגוריתם זה מקבל כקלט מערך של וקטורי מטרה ומחזיר כפלט את דרגותיהם ביחס לקבוצה הנוכחית (וקטורי המטרה הינם מטיפוס נתונים כלשהו ובעלי מימד כלשהו).
2. אלגוריתם לסידור מחדש של מערך "אלמנטים" בשם **ElementsReordering**: אלגוריתם זה מקבל כקלט מערך של "אלמנטים", ועליו לסדר אותו מחדש בהנחה שדרגות וקטורי המטרה מחושבות; על הדרגות הנמוכות להופיע ראשונות.
3. אלגוריתם למיון רב-מימדי של מערך "אלמנטים" בשם **ParetoSorting**: הפעלת אלגוריתמים (1) + (2) על מערך "אלמנטים".

עליכם לציין בתיעוד האלגוריתמים כל הנחה אותה אתם מבצעים וכל דרישת ממשק מטיפוס הנתונים עליהם יבוצע החישוב.

בתרגיל זה, מרחב המשתנים החוקיים  $X \subseteq R^n$  יכול להכיל וקטורים כלשהם ממימד  $n$  מעל הממשיים. מרחב המטרות  $\Phi \subseteq R^m$  מוגדר באמצעות  $m$  פונקציות מטרה המבוססות על  $m$  וקטורי יחידה מוגדרים.

כל וקטור יחידה,  $\vec{e}^{(i)}, i = 1 \dots m$ , הוא וקטור בעל ערך קבוע התלוי באינדקס שלו:

$$\vec{e}^{(i)} := (i, i, i, \dots, i)^T$$

כל פונקציית מטרה היא ריבוע המרחק מווקטור היחידה התואם. למשל, כך נראית פונקציית המטרה הראשונה:

$$f^{(1)}(\vec{x}) = \sum_{j=1}^n (x_j - e_j^{(1)})^2$$

בהינתן וקטור משתנים  $\vec{x}$ , מיפוי למרחב המטרות מוגדר היטב באמצעות הפונקציה הווקטורית:

$$\vec{f}(\vec{x}) = \begin{pmatrix} \sum_{j=1}^n (x_j - e_j^{(1)})^2 \\ \sum_{j=1}^n (x_j - e_j^{(2)})^2 \\ \vdots \\ \sum_{j=1}^n (x_j - e_j^{(m)})^2 \end{pmatrix} \quad (1)$$

מטרת התכנית שתממשו היא קבלת קלט מן המשתמש (הכולל פרמטרים להגדרת הבעיה וכן וקטורי משתנים התחלתיים) והרצת סימולציה נומרית. בכל צעד-זמן של הסימולציה קיימת אוכלוסייה של אלמנטים בעלת גודל ידוע מראש, אשר מוכפלת בגודלה באמצעות צעד עדכון, ונעשה מיון של הווקטורים ביחס למרחב המטרה שלאחריו נזרקים מן האוכלוסייה מחצית מן האיברים ע"פ דרגתם.

יהי  $\mu$  גודלה של האוכלוסייה ההתחלתית; ערך זה יישאר קבוע מתחילת הסימולציה ועד סופה. בכל צעד-זמן, גודל האוכלוסייה יוכפל באופן זמני לכדי  $2\mu$  אלמנטים, ע"י יצירת "עותק מוזז אקראי" של כל אלמנט באוכלוסייה.

בהינתן אלמנט באוכלוסייה,  $\vec{\psi}$ , אלמנט חדש  $\vec{\xi}$  יתקבל ע"י הזזה בוקטור אקראי ממימד  $n$  המתפלג נורמאלי (תוחלת 0 וסטיית

תקן 1), אותו נסמן באמצעות  $\vec{z} \sim \vec{N}(0,1)$  (כלומר, נוסף לכל קואורדינטה בוקטור המקורי משתנה אקראי המתפלג נורמאלי):

$$\vec{\xi} = \vec{\psi} + \vec{z} \quad (2)$$

לאחר יצירת עותק מוזז מכל אלמנט, האוכלוסייה תכיל  $2\mu$  אלמנטים, עליהם יבוצע מיון רב-מימדי באמצעות אלגוריתם **ParetoSorting**.

בשלב האחרון של כל צעד-זמן, ייזרקו מן האוכלוסייה מחצית מן האלמנטים המחזיקים בדרגות הגבוהות ביותר במיון הנוכחי. במקרה בו קיימים מספר אלמנטים עם דרגה זהה ממנה יש לנפות במיון זה, בחירת "השורדים" תהיה אקראית. הסימולציה תסתיים כאשר יושלם מספר נתון של צעדי-זמן, אשר יוגדר מראש ע"י המשתמש.

## פורמט קלט והודעות שגיאה

התכנית תקבל כארגומנטים שם קובץ קלט ושם קובץ פלט. לאחר קומפילציה של התכנית, למשל לכדי קובץ הרצה בשם `simulateKugel`, כך ניתן יהיה להריץ אותה במקרה של עבודה דרך טרמינל:

```
>> ./simulateKugel init.dat output.dat
```

על קובץ הקלט להכיל בשורה הראשונה את גודל האוכלוסייה  $\mu$ , את מימד מרחב המשתנים  $n$ , את מימד מרחב המטרות  $m$ , ואת מספר צעדי הזמן שעל הסימולציה לרוץ, מופרדים ברווח בודד. בשורות לאחר מכן הקובץ צריך להכיל  $\mu$  וקטורי משתנים (האוכלוסייה ההתחלתית) – כל שורה תכיל  $n$  קואורדינטות, מופרדות ברווחים. להלן דוגמא של קובץ קלט חוקי:

# תכנות בשפת C++, סתיו 2018-19

```
10 6 3 5000
0.5 0.5 0.5 0.5 0.5 0.5
0.1 0.1 0.2 0.5 0.5 0.5
0.0 0.0 0.0 0.0 0.0 0.1
0.1 0.0 0.2 0.5 0.5 1.0
1.0 0.0 0.0 0.0 0.0 0.0
0.0 1.0 0.0 0.0 0.0 0.0
0.0 0.0 1.0 0.0 0.0 0.0
0.0 0.0 0.0 1.0 0.0 0.0
0.0 0.0 0.0 0.0 1.0 0.0
0.0 0.0 0.0 0.0 0.0 1.0
```

במקרה של קלט לא חוקי (למשל, שגיאת קבצים, הזנת תווים לא חוקיים, חוסר-עקביות במימדי הוקטורים וכיוצא באלה), מנגנון חריגות (Exceptions), שעליכם לממש, יופעל. בעקבותיו התכנית תסיים את ריצתה בשלב זה ותדפיס הודעת שגיאה מתאימה לערוץ השגיאות הסטנדרטי:

1. באם לא קיים קובץ שניתן לתכנית באמצעות המחרוזת `<filename>`, או פתיחתו לקריאה/כתיבה נכשלת, על הודעת השגיאה לציין את שם הקובץ באופן הבא (במקרה ששני הקבצים נכשלים, יש להדפיס רק את כישלון הקלט):  
`ERROR: <filename> does not exist or cannot be opened\n`
2. באם קיימת שגיאה/סטירה בהגדרת הסימולציה בשורת הפתיחה של קובץ הקלט `<input>`, על הודעת השגיאה לציין זאת באופן הבא:

`ERROR: simulation definition in <input> is invalid\n`

3. באם קיימת שגיאה/סטירה בהגדרת האוכלוסייה ההתחלתית בקובץ הקלט `<input>` בשורה מסוימת `<n>`, על הודעת השגיאה לציין זאת באופן הבא:

`ERROR: population definition in <input> at line <n> is invalid\n`

- בהנחה שהקלט של המשתמש חוקי, התכנית תתחיל לסמלץ את מעברי הזמן עד שמספר המעברים המתבקש יחלוף.

## פורמט פלט

בסיום החישובים במעברי הזמן על התכנית לרשום לקובץ הפלט את וקטורי המטרה של האוכלוסייה הסופית. ערכי וקטורי המטרה ייכתבו לקובץ הפלט עם דיוק של לכל היותר 6 ספרות אחרי הנקודה, כאשר  $m$  הקואורדינטות יופרדו ברווח בודד. למשל, כך יכול להיראות קובץ הפלט `output.dat` עבור הדוגמא שלעיל:

```
1.9957 0.000535 2.02972
2.00814 2.00074 0.000138
0.000001 2.00233 2.00255
0.850634 0.548029 0.677972
0.345762 0.810637 1.07955
1.50143 1.04018 0.157897
1.34533 0.187674 0.997386
0.453395 1.36354 0.560787
0.095883 1.54489 1.23409
0.985761 1.4277 0.183395
```

## דרישות מפורשות למימוש

- אתם נדרשים להגדיר אובייקט מטיפוס "אלמנט"; אובייקט זה יכיל לכל הפחות וקטור משתנים, וקטור המטרות הממופה אליו ודרגתו הנוכחית ביחס לאוכלוסיית הוקטורים.
- אתם רשאים לאחסן את האובייקטים מטיפוס "אלמנט" ב-`container` כלשהו מספריית STL לבחירתכם, אך אין חובה לעשות זאת (ספריית STL תילמד בשבוע האחרון של הקורס).
- הניחו כי  $m < n$
- אתם רשאים להניח שגודל האוכלוסייה לא יעלה על  $\mu \leq 1000$  איברים, וכי מימד מרחב המשתנים לא יעלה על  $n \leq 100$ .

# תכנות בשפת ++C, סתיו 2018-19

## דגשים

- יש לבדוק תקינות קלטים; במקרה של אי-תקינות, יש להפעיל מנגנון חריגות ולהציג הודעות שגיאה מתאימות.
- כאמור, בתרגיל זה ניתן להשתמש בספריית STL **במלואה בחלק ב' בלבד** (בחלק א' ניתן לעבוד עם *container* שלה).
- עליכם לתעד את הקוד באמצעות הערות המתארות בקצרה את המחלקות והפונקציות השונות.
- עצה: מומלץ להשתמש בפונקציה המגרילה משתנה אקראי בהתפלגות אחידה (אשר מימשתם בתרגיל בית 3) לצורך הגרלה אקראית המתפלגת באופן נורמאלי במימוש משוואה (2). **פתרון המגריל הזזה המתפלגת באופן אחיד גם כן יתקבל.**
- עליכם לוודא כי התכנית עוברת קומפילציית ++g התואמת את הקומפיילר שעל שרת המכללה ללא כל שגיאות או אזהרות כלשהן, ורצה בהצלחה.
- יש להריץ את הבודק האוטומטי על שרת החוג בטרם ההגשה בכדי לוודא תאימות ונכונות של ההגשה: התחברו לשרת החוג והריצו **hwcheck** על הארכיב שלכם, או לחילופין העלו את הארכיב תוך שימוש בפרוטוקול HTML בקישור <http://cs.telhai.ac.il/homework>

## הגשה

- עליכם להגיש במערכת Moodle קובץ ארכיב מטיפוס zip בלבד, ששמו כולל את קוד הקורס ('11'), שם התרגיל ('ex5') ותעודת הזהות של הסטודנט/ית המגיש/ה, מופרדים בקו תחתי באחד מן הפורמטים הבא:  
**11\_ex5\_studID1\_studID2.zip** או **11\_ex5\_studID.zip** במקרה של הגשת צמד.
- על ארכיב zip זה להכיל את כל קבצי המקור (ממשק/מימוש) הנדרשים לקומפילציה, והוא רשאי להכיל תיעוד טקסטואלי; מבחינת טיפוס קבצים, עליו לכלול רק קבצים עם סיומות **\*.cpp \*.h \*.txt**.
- לדוגמה: על סטודנט שמספר הזיהוי שלו הינו 012345678 להגיש ארכיב בשם **11\_ex5\_012345678.zip** הכולל את כל קבצי המקור של הפרוייקט, ללא תיקיות כלשהן, ורשאי להכיל קובץ טקסטואלי לתיעוד.

אי-הקפדה על ההנחיות, כולל פורמט ההגשה הדיגיטלי, תגרור הורדה בציון התרגיל.  
לא תתקבלנה הגשות באיחור!