

מבנה נתונים ואלגוריתמים

חומר תשפ"א

תרגיל בית תכנתוני

להגשה עד ה – 29/01/2026 בשעה 23:59.

בצלחה!

תרגיל זה מנוסח בלשון זכר מטעמי נוחות בלבד והוא מיועד לכל המגדרים.

מתרgal אחראי על התרגיל: ים סלונימסקי.

הוראות:

- יש להגיש קובץ `zip` ייחיד המכיל את כל קבצי ה – `java` שהשתמשתם בהם, כאשר השם של הקובץ הוא תעוזות זהות של חברי הקבוצה המופרדים על ידי קו תחתון. למשל אם תעוזות זהותן של שני השותפים הן `ID1`, `ID2` אז השם של הקובץ יהיה `zip2_ID1_ID2`. שימוש לב שבתוך ה – `zip` צריכים להיות **הקבצים עצמם** ולא **תיקיות נוספת**.
- ההגשה תבוצע רק ע"י אחד מבני הזוג למקום הייעודי באתר הקורס במודול.
- עליכם לוודא לפני ההגשה במודול כי הקוד שלכם מתקامل ורץ בשרת Microsoft Azure שהוקצה לכם (הוראות מצורפות בקובץ בנפרד).
- מומלץ ואף רצוי להשוו את הפלט שלכם למופיע בקובץ `master.txt` המצורף לצורך בדיקות נכונות.
- זוג שהתרגיל שלו לא יתאפשר בשרת שהוקצה או יועוף בזמן ריצה ציונו בתרגיל והוא 0.
- יש לכתוב קוד קריית מסודר עם שמות משתנים למשתנים, למתודות ולמחלקות.
- יש להקפיד למלא את כל דרישות התרגיל (שימוש בייצוג נכון, סיבוכיות זמן וכו'). אי עמידה בדרישות התרגיל תגרור ציון 0.

1. כלל

קופת החולים "רפואית" זוקה למערכת פשוטה ויעילה לניהול פעילות המרפאה שלה. במסגרת התרגיל, אתם סטודנטים בקורס מבני נתונים ואלגוריתמים, התבקשتم לפתח מערכת שתסייע בניהול הרופאים הפעילים במרפאה ובמעקב אחר המטופלים הממתינים לכל רופא. המערכת תאפשר הוספה והסרה של רופאים במהלך היום, וכן רישום מטופלים לחדרי המסתנה של הרופאים השונים. מטרתכם היא ליצור פתרון שידע להתמודד עם כמות גדולה של רופאים ומטופלים תוך שימוש על ביצועים אופטימליים.

2. פרטיים

כאשר המרפאה נפתחת (אתחול המערכת) היא ריקה לחלוין רפואיים ומטופלים. רופא מזוהה באמצעות מזהה ייחודי *doctor Id*. כאשר רופא מגע למרפאה נרצה לאפשר כניסה שלו למערכת. מטופל מזוהה גם הוא באמצעות מזהה ייחודי *patient Id*. כאשר מטופל מגע למרפאה, הוא מגע לפגישה עם רופא מסוים, لكن נרצה להכנסו לחדר המסתנה עבור אותו רופא.

יתרן ומטופל יהיה מעוניין לעזוב את המרפאה לפני שנפגש עם הרופא ולן יצא מחדר המסתנה. נרצה לאפשר העברת של מטופל מחדר המסתנה של רופא מסוים אל הרופא עצמו. מטופלים נכנסים לרופא אליו הגיעו באותו הסדר בו נכנסו לחדר המסתנה שלו. כאשר הגיע הזמן של מטופל להיבנס לרופא, הוא למעשה יצא מחדר המסתנה של הרופא ומהמערכת כולה.

בנוסף נרצה לאפשר שליפה של נתונים הקשורים לרופאים, למטופלים ולמספר המטופלים הממתינים ברגע בחדרי המסתנה (פירוט על אילו פעולות בדיק נמצא בדרישות הממשק).

כאשר רופא מסוים את עבודתו, נרצה לאפשר הוצאה שלו מהמערכת. שימוש לב שרווף יכול לעזוב את המערכת אך ורק אם חדר המסתנה שלו ריק.

2.1. סימונים ודרישות

- בתרגיל זה נסמן את מספר הרופאים שנמצאים ברגע המערכת ב – D ואת מספר המטופלים שנמצאים ברגע המערכת ב – P .
- אלא אם נאמר אחרת, אתם לא יכולים להניח שהקלט לפועלות תקין. הגדרתו של קלט לא תקין נמצאת תחת הסביר אודות הפונקציה בהגדרת המשך (3). במקרה והקלט לא תקין, יש לזרוק חריגה מסוג `IllegalArgumentException` בלבד.
- אסור לכם ליבא אף מחלוקת או פונקציה שלא כתבתם, כלומר המילה השמורה `import` לא יכולה להופיע באף קובץ שתיגשו. כמו כן השימוש במחלוקת `System` אסור גם הוא.

3. ממשך

- את מבנה הנתונים שמייצג את המרפאה תבנו בתוך מחלוקת פומבית בשם `ClinicManager` בתוך קובץ `ClinicManager.java`
- למחלוקת חיבר להיות בנאי ברירת מחדל (`ClinicManager()`) שיאתחל את המערכת בסביבות זמן (`O(1)`).
- בקובץ ה – `ClinicManager.java` שמספק לכם עם התרגיל, במחלוקת `ClinicManager`, קיימים שני שדות קבועים לשימושכם:
 - **String MIN_ID**: מזהה ייחודי שUMBOTICH לכם שכל מזהה שיתקבל במערכת יהיה גדול ממנו.
 - **String MAX_ID**: מזהה ייחודי שUMBOTICH לכם שכל מזהה שיתקבל במערכת יהיה קטן ממנו.
- שימושם לב שאם **תחליטו** להשתמש האלו, הם **חייבים להופיע** בקובץ `java`.
ClinicManager.java בבדיקה כי שם מופיעים בקובץ אותו סיפקנו לכם.

כמו כן, על המחלוקת להכיל את הפעולות הבאות:

void doctorEnter(String doctorId)

מכניסה למערכת את הרופא בעל המזהה הייחודי `doctorId`.

סביבות זמן: $O(\log(D))$.

- לא ניתן להכניס למערכת רופא שכבר נמצא בה.

void doctorLeave(String doctorId)

מוציאה מהמערכת את הרופא בעל המזהה הייחודי *doctorId*.
סיבוכיות זמן: $O(\log(D))$.

- לא ניתן להוציא מהמערכת רופא שאינו נמצא בה.
- לא ניתן להוציא מהמערכת רופא שחדר המתנה שלו אינט ריק.

void patientEnter(String patientId, String doctorId)

מכניסה לחדר המתנה של הרופא בעל המזהה הייחודי *doctorId*, את המטופל בעל המזהה הייחודי *patientId*.

סיבוכיות זמן: $O(\log(P) + \log(D))$.

- לא ניתן להכניס למערכת מטופל שכבר נמצא בה.
- לא ניתן להכניס מטופל לחדר המתנה של רופא שאינו נמצא במערכת.

String nextPatientLeave(String doctorId)

מוציאה מחדר המתנה (ומהמערכת כולה) את המטופל הבא מחדר המתנה של הרופא בעל המזהה הייחודי *doctorId* ומחזיר את המזהה הייחודי של אותו המטופל.
סיבוכיות זמן: $O(\log(D) + \log(P))$.

- לא ניתן להוציא מטופל מחדר המתנה של רופא שאינו נמצא במערכת.
- לא ניתן להוציא מטופל מחדר המתנה ריק.

void patientLeaveEarly(String patientId)

מוציאה מחדר המתנה (ומהמערכת כולה) את המטופל בעל המזהה הייחודי *patientId*.
סיבוכיות זמן: $O(\log(D) + \log(P))$.

- לא ניתן להוציא מהמערכת מטופל שאינו נמצא בה.

int numPatients(String doctorId)

מחזירה את מספר המטופלים שנמצאים ברגע בחדר המתנה של הרופא בעל המזהה הייחודי *doctorId*.
סיבוכיות זמן: $O(\log(D))$.

- לא ניתן לבדוק את מספר המטופלים בחדר המתנה של רופא שאינו נמצא במערכת.

String nextPatient(String doctorId)

מחזירה את המזהה הייחודי של המטופל הבא בחדר המתנה של הרופא בעל המזהה הייחודי *doctorId*. סיבוכיות זמן: $O(\log(D))$.

- לא ניתן להחזיר את המזהה הייחודי של המטופל הבא בחדר המתנה של רופא שאינו נמצא במערכת.
- לא ניתן להחזיר את המזהה הייחודי של המטופל הבא בחדר המתנה ריק.

String waitingForDoctor(String patientId)

מחזירה את המזהה הייחודי של הרופא שבחדר המתנה שלו נמצא ברגע המטופל בעל המזהה הייחודי *patientId*. סיבוכיות זמן: $O(\log(P))$.

- לא ניתן להחזיר את המזהה הייחודי של הרופא אליו ממתין מטופל שאינו נמצא במערכת.

int numDoctorsInRange(int a, int b)

מחזירה את מספר הרופאים שכמות המטופלים שמתאימים ברגע בחדר המתנה שלהם, נמצא בטווח $[a, b]$. סיבוכיות זמן: $O(\log(D))$.

int averageLoadWithinRange(int a, int b)

מחזירה את ממוצע מספר המטופלים מבין כל חדרי המתנה שמספר המטופלים שנמצאים בהם ברגע, נמצא בטווח $[a, b]$ – מעוגל כלפי מטה במקרה של שבר. סיבוכיות זמן: $O(\log(D))$.