

פיתוח תוכנה מתקדם 2 – סמסטר ב' מועד ב'

תזכורת: כתובת מערכת הבדיקות: <https://cktest.cs.colman.ac.il/>. שם הקורס PTM2, מועד ב'. לאחר הורדת המבחן ממערכת הבדיקות. העתיקו את כל קובצי ה Java לתוך הפרויקט ב package בשם test. במבחן זה 3 שאלות, חובה לענות על כל 3 השאלות ולהגיש למערכת הבדיקות במוד הגשה סופית לפני סוף המבחן.

שאלה 1 - תכנות מקבילי באמצעות ת'רדים (35 נק')

בקובץ Par.java עליכם לממש את המחלקה Par באופן הבא:

- Par היא Thread Pool שתומך בשתי פעולות חישוב map ו fold, ופעולת סגירה close.
 - בבנאי Par תקבל את מס' הת'רדים המקסימלי שמותר לפתוח
 - המתודה fold
 - תקבל מערך של טיפוס פרמטרי V, ואופרטור בינארי.
 - המתודה "תקפל" את כל הערכים לערך בודד באמצעות אופרטור זה.
 - על החישוב להתבצע בת'רד נפרד כמשימה של thread pool,
 - ולהחזיר מיד אובייקט Future מתאים (לא חיקוי, אלא java.util.concurrent.Future)
 - לדוגמה ההפעלה הבאה סוכמת את כל איברי buff (מערך של Integer) ברקע ומחזירה מיד Future<Integer> שיכיל את התוצאה לכשתחושב.
- ```
Future<Integer> f=p.fold(buff, (x,y)->x+y);
```

- המתודה map
    - תקבל מערך מטיפוס פרמטרי V, ופונקציה מ V לטיפוס פרמטרי אחר כלשהו R.
    - זה לא חייב להיות אותו V של המתודה fold
    - המתודה תמפה ברקע באמצעות הפונקציה כל איבר V במערך לאיבר מסוג R
    - המתודה תחזיר Future<List<R>> (גם כאן, האמתי, לא החיקוי)
    - לדוגמה ההפעלה הבאה הופכת כל ערך ב buff (מערך של Integer) למחרוזת המכילה את הערך עם "\$". היא עושה זאת ברקע, ומחזירה מיד Future<List<String>> שיכיל את התוצאות לכשתחושבנה.
- ```
Future<List<String>> f2=p.map(buff, i->i+"$");
```

- המתודה close תשלים משימות שכבר ניתנו אך לא תקבל משימות חדשות ותסגור את ה thread pool על כל הת'רדים שפתח.

הבדיקה כוללת בדיקה שנפתחו מס' נכון של תרדים, קבלה של תוצאות נכונות, ושכל הת'רדים נסגרו לאחר הקריאה ל close. מוד האימון ניתן ב MainTrain1.java. שאלה זו תיבדק בצורה אוטומטית בלבד, ולכן חובה שתתקמפל ותרוץ ללא שגיאות. מוד ההגשה דומה למוד האימון.

שאלה 2 - ForkJoinPool (35 נק')

פונקציית ה GCD (Greatest Common Divider) מוגדרת באופן הרקורסיבי הבא:

```
int gcd(int a, int b){  
    if(b==0) return a;  
    return gcd(b,a%b);  
}
```

עליכם לממש את המחלקה FJ_GCD (בקובץ FJ_GCD.java) המבצעת את חישוב ה GCD כך שהקריאה הרקורסיבית מתבצעת ברקע ע"י ה ForkJoinPool. בפרט, עליכם לדאוג שהתוצאה הנכונה תגיע כאשר נקרא למתודה invoke של ה ForkJoinPool.

קריאה לדוגמה מתוך MainTrain2:

```
ForkJoinPool fj=new ForkJoinPool();  
if(fj.invoke(new FJ_GCD(18, 27))!=9)  
    System.out.println("you got a wrong result (-11)");
```

ה GCD של 18 ו 27 הוא כמובן 9. זו התוצאה שצפויה לחזור מהקריאה ל invoke של ה ForkJoinPool.

שאלה זו אף תיבדק ידנית. עם זאת משקל קוד מתקפל ורץ ללא שגיאות ריצה = 10 נק'.

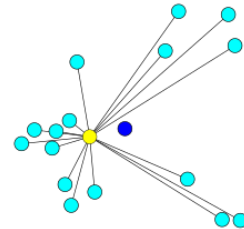
מוד ההגשה זהה למוד האימון.

שאלה 3 - אופטימיזציות קוד (30 נק')

חציון גיאומטרי זו הנקודה שסכום המרחקים האוקלידיים ממנה לכל שאר הנק' הוא מינימאלי.

לדוגמה זו הנק' הצהובה בתרשים שבצד.

פורמלית:
$$\arg \min_{y \in \mathbb{R}^n} \sum_{i=1}^m \|x_i - y\|_2$$
 כלומר יש למצוא את הנק' y מתוך m הנק' $x_1 \dots x_m$ כך שסכום המרחקים האוקלידיים מנק' y לכל השאר הוא מינימאלי.



בקובץ BadCode.java מצויה הפונקציה geoMedian אשר בהינתן מערך של נקודות (אובייקטי Point) היא מחזירה את הנקודה המהווה את החציון הגיאומטרי.

האימפלמנטציה בקובץ זה היא לא יעילה. עליכם לערוך את הקוד בקובץ GoodCode.java כך שהאימפלמנטציה תהיה לפחות פי 4 יותר מהירה. טיפ: תחתנו חישובים מיותרים.

כמובן, יש להחזיר את התוצאה הנכונה.

מוד האימון זהה למוד ההגשה והוא ניתן בקובץ MainTrain3.java.

תחילה מיצרים קלט של מערך אקראי.

לאחר מכן מתבצעת מדידת זמן של BadCode ושל המימוש שלכם ב GoodCode.

קוד שהוא לפחות פי 4 יותר מהיר יקבל את מלוא 30 הנק'.

קוד שהוא פחות מפי 4 יותר מהיר יקבל את החלק היחסי (למשל קוד שהוא רק פי 2 מהיר יותר יקבל 15 מתוך 30 הנק').

שאלה זו אף תיבדק ידנית. עם זאת הקנס על קוד עם שגיאת ריצה או קומפילציה הוא 5 נק'.

הגשה

עליכם להיכנס למערכת הבדיקות בכתובת: <https://cktest.cs.colman.ac.il/> ולהגיש ל PTM2 ומועד ב' את הקבצים GoodCode.java , FJ_GCD.java, Par.java ,

בכל הגשה יש להגיש את כל הקבצים (ולוהתייחס לפלט רק של השאלות שעניתם עליהן)

ניתן להגיש במוד אימון ובמוד הגשה כמה פעמים שתמצאו עד לסוף המבחן.

בסוף המבחן יש להגיש במוד הגשה ואז במוד הגשה סופית. אחריה תקבלו מס' אסמכתא בין 4 ספרות. לאחר הגשה במוד זה לא תוכלו להגיש יותר.

בהצלחה!