

תוכנה 1 – סתיו תשע"ד

תרגיל מספר 6

הנחיות כלליות:

- קראו בעיון את קובץ נהלי הגשת התרגילים אשר נמצא באתר הקורס.
- הגשת התרגיל תעשה במערכת ה-moodle בלבד (<http://moodle.tau.ac.il/>).
- יש להגיש קובץ zip יחיד הנושא את שם המשתמש ומספר התרגיל (לדוגמא, עבור המשתמש aviv יקרא הקובץ aviv_hw6.zip). קובץ ה-zip יכול:
 - קובץ פרטים אישיים בשם details.txt המכיל את שמכם ומספר ת.ז.
 - קבצי ה-java של התוכניות אותם התבקשתם לממש, כולל תיקיות החבילה.

חלק א' (80%) – כתיבת מחלקות לפי מפרט

בחלק זה של התרגיל נממש מחלקה בשם `il.ac.tau.cs.sw1.disjointsets.DisjointSets` המייצגת קבוצה S של קבוצות זרות: $S = \{S_1, S_2, \dots, S_n\}$ כך שכל S_i הינה קבוצה של מספרים שלמים אי-שליליים, והחיתוך בין הקבוצות ריק (http://en.wikipedia.org/wiki/Disjoint_sets). המחלקה תתמוך בשירותים הציבוריים הבאים:

```
/**
 * Create a singleton set containing x (i.e. {x}) and add it to this object.
 * @pre x is not in any of the sets Si
 * @post this == $prev(this) ∪ {{x}}
 */
public void makeSet(int x) {...}

/**
 * Return true if and only if x and y belong to the same set in this object.
 * @pre inASet(x) && inASet(y) && x in Si && y in Sj
 * @return true iff i == j
 */
public boolean equiv(int x, int y) {...}

/**
 * Find the different sets that x and y belongs to. Remove the sets from this
 * object and add their union
 * @pre inASet(x) && inASet(y) && x in Si && y in Sj && i != j
 * @post this = $prev(this) - {Si} - {Sj} ∪ { Si ∪ Sj }
 */
public void joinSets(int x, int y) {...}

/**
 * Return true if and only if x is in a set of this object
 * @pre true ("no precondition")
 * @return true iff exists i such that x is in Si
 */
public boolean inASet(int x) {...}
```

```

/**
 * Return the number of sets in S (each with a different root)
 *
 * @return |S|
 */
public int getNumSets() {...}

/**
 * Return the number of distinct values in S
 *
 * @return |S1|+|S2|+...+|Sn| (n == getNumSets())
 */
public int getNumValues() {...}

```

כמו כן תתמוך המחלקה בבנאי ללא ארגומנטים המייצג קבוצה ריקה של קבוצות זרות.

אפשרות אחת לממש את המחלקה היא ע"י ייצוג כל קבוצה S_i של S בתור עץ. כל צומת בעץ תייצג מספר שלם אי שלילי x , ותצביע לצומת הורה. בדרך זו, שורש של עץ מייצג בצורה ייחודית את הסט S_i .

מבנה הנתונים הנ"ל ניתן למימוש באמצעות מערך בשם parent בו כל תא מייצג צומת בעץ. האינדקס של כל תא במערך מייצג את המספר המאוחסן בצומת (x) , וערך התא $(parent[x])$ משמש כמצביע לאינדקס צומת ההורה. תא במערך שערכו -1 מייצג ערך שלא שייך לאף סט. שורש כל עץ מצביע לעצמו. אורך המערך צריך להיות גדול יותר מכל מספר x שנמצא כרגע בסט S_i כלשהוא. לפיכך, אם נוסף מספר גדול יותר מהמקסימאלי עד כה, נצטרך להחליף את המערך במערך חדש גדול יותר.

כל אובייקט של המחלקה DisjointSets מייצג קבוצה של קבוצות זרות, של שלמים אי-שליליים $\{S_1, S_2, \dots, S_n\}$. ייצוג זה מוגדר פורמאלית ע"י פונקצית העזר $root(x)$:

```

if parent[x] = -1 then root(x) = -1
else if parent[x] == x then root(x) = x
else root(x) = root(parent[x]) //recursive expression

```

כעת, נוכל להגדיר את המיפוי מ-parent, שדה של האובייקט, לקבוצה של קבוצות $S = \{S_1, S_2, \dots, S_n\}$

- איבר שאינו נמצא ב-S כלל:

For all x , [for all i , $1 \leq i \leq n$, $x \notin S_i$] iff $[x \geq parent.length \text{ or } parent[x] == -1]$

- שני איברים באותו סט S_i :

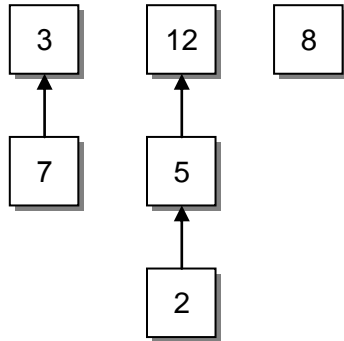
For all $0 \leq x, y \leq parent.length$, $x, y \in S_i$ (x and y are in the same set)

iff $root(x) == root(y)$ and $root(x) != -1$

למשל, לאובייקט של DisjointSets כאשר מערך ה- parent הוא:

-1	-1	5	3	-1	12	-1	3	8	-1	-1	-1	12
0	1	2	3	4	5	6	7	8	9	10	11	12

העצים יהיו:

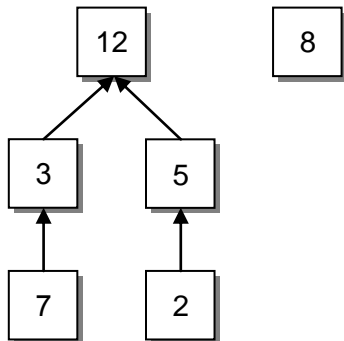


ו- $S = \{ \{3,7\} \{12,5,2\} \{8\} \}$

אם נקרא ל- $equiv(3,5)$ נקבל false. אם נפעיל $joinSets(7,5)$ נקבל:

-1	-1	5	12	-1	12	-1	3	8	-1	-1	-1	12
0	1	2	3	4	5	6	7	8	9	10	11	12

העצים יהיו:



(הערה: ייתכן מבנה אחר לעץ השמאלי, אך הוא חייב להיות מורכב מאותם צמתים)

ו- $S = \{ \{3,7,12,5,2\} \{8\} \}$

אם נפעיל כעת $makeSet(4)$ נקבל ש- $S = \{ \{3,7,12,5,2\} \{8\} \{4\} \}$

המשימה:

- השלימו את מימוש המחלקה DisjointSets הנתונה באתר הקורס. שימו לב שבמחלקה נתונה לכם פונקציית main שיכולה לשמש לבדיקה, וכן שניתן להגדיר שירותי עזר. מומלץ להגדיר:
- פונקציה שמוודאת שמערך ה-parent ארוך מספיק להכיל את כל האיברים, ואם לא, מגדילה אותו).
 - פונקציה $getRoot(int x)$, אשר תחזיר, עבור כל ערך x, את השורש של העץ של x.

חלק ב' (20%) – חידות ג'אווא

בכל סעיף של חלק זה תקבלו חבילה ובה מספר מחלקות. חבילות אלה מצורפות לתרגיל באתר הקורס. עליכם לשנות את הקוד בהתאם להנחיות, כדי לקבל את התוצאה הנדרשת. יש להגיש את כל המחלקות (כולל אלה שלא שיניתם בהם דבר, וכמובן, הקוד המתוקן).

1. החבילה `il.ac.tau.cs.sw1.riddle.a` מכילה שתי מחלקות, A ו-B. B היא תכנית המקבלת כארגומנט מס' שלם. עליכם לשנות את הקוד בתוך `printA()` בלבד כך שבהרצת פונקציית ה-`main` ב-B יודפס המספר שניתן כקלט בין A1 ל-A2. לדוגמא, אם הקלט הוא 15, יודפס:

```
B
A1
15
A2
```

- **מותר:** לשנות את הקוד בתוך `printA`.
- **אסור:** לשנות את B, את חתימת `printA()`, וקוד ב-A שנמצא מחוץ ל-`printA()`.

2. החבילה `il.ac.tau.cs.sw1.riddle.b` מכילה שלוש מחלקות, A, B ו-C. C היא תכנית המקבלת כארגומנטים שלוש מחרוזות. עליכם להשלים את מימוש המתודות `printA`, `printA2` ו-`printA3` ב-A כך שבהרצת פונקציית ה-`main` ב-C יודפסו 3 המחרוזות בין הכוכביות. לדוגמא, אם הקלט לתכנית הוא `hello world bye`, יודפס:

```
hello
***
world
***
bye
```

- **מותר:** לשנות את הקוד בתוך `printA`, `printA2` ו-`printA3`.
- **אסור:** לשנות את B ו-C, את חתימות המתודות `printA`, `printA2` ו-`printA3`, וקוד ב-A שנמצא מחוץ למתודות הנ"ל.

3. החבילה `il.ac.tau.cs.sw1.riddle.c` מכילה שתי מחלקות, A ו-B. B היא תכנית. עליכם לשנות את חתימות המתודות והשדות ב-A כך ש-(א') הקוד יתקמפל ללא שגיאות, ו-(ב') B תדפיס, כשורה אחרונה, **success!**. ייתכן שיודפסו שורות נוספות לפני שורה זו, המשמשות לבקרה בלבד (כל עוד מודפס **success!** הפתרון נכון).

- **מותר:** לשנות את חתימות המתודות ואת השדות, כולל: שינוי נראות (public ל-private ולהפך), הוספת והורדת `static`, שינוי טיפוס ההחזרה של מתודה, ושינוי הארגומנטים למתודה.
- **אסור:** לשנות את B, לשנות קוד בתוך מתודות A, לשנות את שמות המתודות ב-A ולשנות את ערך השדה `k`.

4. החבילה `il.ac.tau.cs.sw1.riddle.d` מכילה שתי מחלקות, `A` ו-`B`.
`A` היא תכנית המדפיסה מס' שלם. בתוך קוד `A` מופיעות 4 קריאות לפונקציה `setI` של מחלקה `B`, עם הארגומנטים `k`, `B.I`, `j`, ו-`1` בהתאמה. עליכם לשנות את הארגומנטים של `setI` כך שהקוד יתקמפל ללא שגיאות, ובסופו של דבר יודפס המס' **210**. עדיין, הארגומנט של כל קריאה חייב להיות `I`, `j`, `k` או `1`.

- **מותר:** לשנות את הארגומנטים של `setI`, למשל במקום בו הופיע `setI(j)` אפשר לשנות ל-`setI(k)`. אם יש צורך ניתן להוסיף שם מחלקה או מופע לפני שדה, למשל `B.I` או `this.1`.
- **אסור:** להשתמש במשתנים\שדות מלבד הארבעה הנ"ל, או במס' טבעיים. אין לשנות את `B` או כל קוד ב-`A` מלבד הארגומנטים המועברים ל-`setI`.

בהצלחה!