

חזרה בנושא רשימה

מיזוג רשימות (ללא כפילויות)

מטרות

- תרגול רעיון המיזוג (מתוך תוכנית הלימודים של יסודות)
- מעבר מקביל על שתי רשימות
- ניתוח סיבוכיות זמן ריצה

רמת השאלה

מתאימה למבחן מסכם ברשימות, אך קלה משאלת בגרות. עטיפת השאלה בסיפור מסגרת יכולה להעלותה לשאלה ברמת גבוהה יותר.

המשימה

נתונות שתי רשימות של מספרים ממשיים, ממוינות בסדר עולה (בכל רשימה אין כפילויות). ברצוננו למזג את שתי הרשימות לרשימה ממוינת אחת, ללא כפילויות.

מה עליכם לעשות?

א. ממשו את הפעולה הבאה:

```
public static List<Double> merge (List<Double> lst1, List< Double>lst2)
```

המקבלת שתי רשימות, lst1 ו-lst2, של מספרים ממשיים, ממוינות בסדר עולה (בכל רשימה אין כפילויות). הפעולה ממזגת את שתי הרשימות לרשימה חדשה, ממוינת בסדר עולה כך שאף מספר לא יופיע ברשימה יותר מפעם אחת.

ב. מהי סיבוכיות זמן ריצה של הפעולה שכתבתם בסעיף א. הסבירו.

הנחיות מיוחדות

1. ערך ההשוואה בין שתי הרשימות הוא פשוט אך במידה וברצונכם לשנות את השאלה ולהפוך את ערך ההשוואה למחרוזתי, יש לבחור בפעולת השוואה מתאימה.
2. המעוניינים לתרגל מיזוג של רשימות, יכולים להדגיש בנוסח השאלה שאין להשתמש באוספים נוספים (פרט לרשימות הנתונות).
3. ניתן להראות פן נוסף ומתקדם של פתרון לשאלה. פתרון יעיל מאד לבעיה יהיה פתרון שישתמש בטיפוס הנתונים Map. מכיוון שאוסף המספרים לא יכיל כפילויות ואנו מעוניינים לקבל את אוסף המספרים באופן ממוין, הרי ניתן להכניס את כל איברי הרשימה הראשונה לתוך Map, ואז להכניס

גם את כל איברי הרשימה השנייה לאותה מפה. פעולת ההכנסה מוודאת שלא יכנסו מספרים כפולים והחזרת אוסף המספרים השמור במפה, תחזיר אותם באופן ממוין בסדר עולה. המפתח (key) במפה יהיה המספר הממשי והערך הוא חסר משמעות מבחינתנו (למשל, null בכל המקרים).

4. השאלה המקורית עוסקת ברשימות של איברים מורכבים יותר מאשר בשאלה הסופית המוצגת כאן. ניתן להרכיב את השאלה ולקרר אותה לנוסח המקורי בו כל איבר ברשימה מכיל מפתח כלשהו הניתן להשוואה, וערך (או ערכים) הקשורים למפתח זה. נוסח השאלה, כמו גם פתרונה, יישארו זהים, אך יש לשים לב להשתמש בפעולות השוואה מתאימות ולדאוג שהערכים המשווים יוכנסו לתוכנית באופן המאפשר השוואה נכונה.

לדוגמה: הכנסת ערכים מחרוזתיים כערכי המפתח, כאשר הם בעלי אורך שונה של תווים תגרום לתוצאות משונות ביותר של ההשוואה ושל היחס הלקסיקוגרפי שיווצר בין האיברים.
נדגים:

```
String s1= "85";
String s2 = "170";
System.out.println (s1.compareTo(s2));
```

ההשוואה תחזיר ערך חיובי המציין ש-s1 גדולה מ-s2. הערך השגוי נובע מכך שבמקום הראשון האפשרי להשוואה (התו השמאלי), אכן 1 קטן מ-8!! לכן יש לדאוג בנוסח השאלה להבהיר שהערכים צריכים להיות באורכים שווים!

מקור השאלה

מעובדת מתוך בגרות עיצוב תוכנה, תשנ"ו – 1996, שאלון 899205, שאלה 13. עובד בראשיתו כפרויקט סיכום השתלמות על ידי בנצי גורפינקל.

נוסח המקור:

בעיר "שממון" היו שתי חברות שידוכים:

אחת – "טובים השניים" בבעלותה של מירי, והשנייה – "קשר אמיתי" בבעלותו של רובי.

כל חברה ניהלה רשימה של לקוחות, הכוללת את הפרטים הבאים: ת"ז, שם משפחה, שם פרטי, כתובת, מספר טלפון. הרשימות מוינו לפי מספר ת"ז בסדר עולה.

כשנישאו מירי ורובי, לאחר שנים של תחרות ורווקות, החליטו למזג את שתי החברות. לשם כך צריך היה למזג את שתי הרשימות של הלקוחות לרשימה ממוינת אחת.

א. כתוב אלגוריתם **מזג-רשימות** המקבל את הרשימות הממוינות של מירי ושל רובי וממזג את שתי הרשימות לרשימה חדשה, ממוינת על פי מספר ת"ז בסדר עולה.

לקוח שהופיע בשתי הרשימות מועתק לרשימה חדשה פעם אחת בלבד.

ב. מה סדר הגודל של סיבוכיות זמן ריצה של האלגוריתם שכתבת בסעיף א'? נמק.

הערה: הפעולות שבהן יכול האלגוריתם להשתמש הן כל הפעולות המוגדרות על טיפוס הנתונים רשימה (LIST), ואין צורך לפרט אותן.

על ההתאמה לשפות מונחות עצמים

1. אילו נעשתה התאמה מלאה לסיפור המסגרת הרי שרשימה של לקוחות בשפות מונחות עצמים פירושה, שכל איבר ברשימה הוא מטיפוס **לקוח Client**. כדי להשתמש בלקוח מספיק לדעת שהממשק שלו כולל פעולה המחזירה את ערך תעודת הזהות ואין לנו צורך בנתונים נוספים. לגבי השוואת ערכי תעודות הזהות יש לשים לב להנחיות המיוחדות, סעיף 4.
2. פעולת המיזוג היא פעולת שירות המוגדרת מחוץ למחלקה List ולכן היא מוכרחה להיות מוגדרת כפעולה סטטית. מקומה של הפעולה בתוכנית הראשית (מכיוון שהיא ספציפית לטיפוס מסוים אין טעם להגדירה בתוכנית שירות לרשימות). נושא זה אמנם אינו עולה בשאלה אך יש להבין שהצורך למזג שתי רשימות נתונות מסביר את הגדרת הפעולה כסטטית.
3. סיפור המסגרת טושטש וצומצם כדי להתמקד בעיקר המטלה שבשאלה. לגבי הרחבת סיפור המסגרת ניתנו הנחיות מפורטות.