

## {יבש 3}

מגישים:

אראג מחיטריאן: 315031203

חוסין עזקי: 213382104

1. במבנה SortedList הגנרי, מה הדרישות ההכרחיות שעל הטיפוס T לקיים? הסבירו במשפט לכל דרישה.

**תשובה:** הדרישות על הטיפוס T בהקשר של המימוש שלנו למחלקה SortedList<T> הן:

1. **בנאי העתקה:** כדי שנוכל לבצע העתקה של הערך ל Data T שמתקבל לתוך חוליה ברשימה ששומרת את הערך ללא תלות בזיכרון המשתמש. כפי שהתבקשנו. במילים אחרות השמירה של ערכי T חייבת להיות by value ולא by reference.
2. **אופרטור בינרי < :** לטעמי מיון בתוך הרשימה הכרחית היכולת לדעת מה הסדר בין שני אברים t1,t2 מטיפוס T.
3. **הורס:** מאחר והמימוש שלנו שומר עותקים של T, בהורס של כל חוליה צריך לקרוא להורס של T (זה קורה ברקע ולא במפורש כמו לקרוא להורס של אובייקט שמצביעים עליו) חייב של T יהיה הורס כי אחרת עלולות להיות זליגות זכרון עקב השימוש ברשימה שמיישנו שהן תלויות בכלל בטיפוס T.

2. נניח כי היינו רוצים לממש איטרטור non-const עבור ה-SortedList. כלומר, עבור איטרטור זה, האופרטור \* היה מחזיר &T. איזו בעיה עלולה להיווצר במימוש?

**תשובה:** הבעיה שעלולה לקרות היא שהמשתמש יכול לשנות את הערכי T במבנה עצמו כרצונו ולהרוס את המיון שהמבנה צריך לשמור עליו. ואין שום דרך למבנה לדעת שאחד מערכיו השתנה בצורה שמצריכה סידור מחדש. למשל עבור T שהוא בפועל int, אם הרשימה המקורית היא {1,2,3}, אם נשנה את ה 2 ל 5, היה מצופה שהרשימה תהפוך להיות {1,3,5}, בפועל זה לא מה שיקרה ונקבל {1,2,3}.

3. סטודנטית בקורס מבוא לתכנות מערכות סיימה לפתור את תרגיל בית 3 והחליטה להשתמש ברשימה הממויינת מהתרגיל לפרוייקט צד שהיא מפתחת בשעות הפנאי. במימוש פרוייקט הצד הסטודנטית נדרשה לסנן רשימה של מספרים שלמים, כך שישארו בתור רק מספרים המתחלקים במספר כלשהו שאינו ידוע בזמן קומפילציה אלא רק בזמן ריצה. הסבירו כיצד ניתן לממש את הפונקציונליות הדרושה בעזרת הפונקציה filter.

**תשובה:** הסטודנטית תצטרך להשתמש ב `SortedList<int>` ולשלוח לפונקציית `filter` פרדיקט עם אופרטור `()` שמקבל `int` ושחזיר `true` אם הקלט לאופרטור מתחלק בערך `k` שמאותחל איתו אותו פרדיקט בזמן ריצה ו `false` אחרת.

דרך המימוש ממש דומה לפונקטור שמימשנו ברטוב לבדיקת `TaskType`.