תרגיל בית 3- חלק יבש

מספר קורס : 234124 תאריך : 27.7.2024

שם : לינוי גבע ת.ז : 213518194

שם : עדי וולפמן ת.ז :211402789

3.1 שאלות על SortedList

1. במבנה שבנינו הדרישות ההכרחיות שעל הטיפוס T לקיים הם :
בנאי העתקה בהוספת אובייקט לרשימה עלינו ליצור עותק של האובייקט שאנו רוצות להוסיף על מנת ליצור מבנה עצמאי מבחינת זיכרון ומידע, ועל כן עליו להיות בעל בנאי העתקה שיאפשר זאת.

אופרטור < האובייקט שאנחנו מייצרים ממיין את הטיפוס T לפי האופרטור <, ולכן לכל אובייקט שנרצה ליצור לו רשימה האופרטור < חייב להיות ממומש.

הורס- אנחנו משתמשים באובייקט גנרי. בקוד שלנו אנו לא משתמשים בהקצאות דינאמיות של T, ועדיין על האובייקט להיות אחראי על הזיכרון שלו ולכן יש צורך בהורס בין אם דיפולטי ובין אם מוגדר.

*במימוש הספציפי שלנו לא נדרש אופרטור השמה על T (גם לא default) כיוון שבאופרטור ההשמה של list אנו משתמשים ב-copy c'tore של T.

2. במידה והיינו מממשים במחלקה שלנו non-const iterator בלבד, היינו נתקלים בבעיות הבאות :

- במקרה בו היינו עובדים עם const SortedList< T> ומנסים לעבור באיטרציה על מבנה הנתונים, הקוד היה עובר קומפילצי כיוון שהמתודות begin, end מוגדרות כ-const כלומר לא משנות את הרשימה.
אך במידה ובמהלך האיטרציה היינו מפעילים פונקציה שמשנה את ערכי T,
למשל אם קיים אופרטור השמה :

```
for (T& value: list) {
    value= T(other);
}
```

היינו משנים את אברי הרשימה על אף שהיא מוגדרת כקבועה
T אינו מוגדר בהכרח כקבוע, בהגדרת רשימה קבועה אנו מבטיחים כי השדות של הרשימה יהיו קבועות במקרה שלנו הכתובת של ראש הרשימה אינה תשתנה, אין שום הגבלה על ערכה וזו כאמור התנהגות בלתי רצוייה ולכן נקבל שגיאת זמן ריצה.

- שגיאה נוספת שהייתה עלולה להתקבל במקרה שכזה היא קבלת מבנה נתונים שאינו ממויין.
מבנה הנתונים שלנו רוצה להבטיח מונוטוניות במידע. דבר זה מובטח ומתבצע בכל הכנסת איבר חדש, ואינו נבדק בפעולות אחרות, מנקודת הנחה שאם איבר ממוקם לפני השני הוא ישאר ככה.
במקרה ובמהלך איטרציה נוכל לשנות את הערך של איברי הרשימה, נוכל לפגוע במונוטוניות של המידע ולגרום להתנהגות בלתי רצוייה בהמשך התוכנית.

3. בהינתן הפרמטרים הבאים:

```
int n;  //the modulo parameter
SortedList<int> list; // the list with all the data
```

נשתמש ב-lambda בקריאה לפונקציית filter באמצעת הפקודה הבאה:

```
SortedList<int> newList = list.filter([=](int i) { return (i % n == 0); });
```

כאמור הפקודה filter מסננת רשימה עבור פונקציה המחזירה ערך בוליאני עבור כל איבר שלה.

יצרנו כפרמטר לפונקציה, מתודה המקבלת int (טיפוס הרשימה) ובודקת את שארית החלוקה במספר n (מספר שיכל להתקבל ולהתעדכן בזמן ריצה.)

הפונקציה filter תיצור רשימה חדשה בה ישארו רק המספרים שיתחלקו ב-n כנדרש.