

Dry

שאלה 1:

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

1. אנחנו חייבים בנאי העתקה לייצר עותקים מהאובייקטים מסוג T. כשמגדירים איבר חדש ורוצים לתת לו ערך של אובייקט אחר נשתמש בו.
2. חייבים הורס למחוק את האיברים מטיפוס T.
3. חייבים אופרטור > כדי שנוכל לעשות מיון לאיבייקטים ברשימה הממוינת.

שאלה 2:

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

הבעיה במימוש איטרטור non-const עבור SortedList היא שהגישה לאיבר ברשימה עכשיו מאפשרת למשתמש קריאה וכתובה ולכן יכול להיות שהמשתמש משנה איבייקט ברשימה ונקבל רשימה לא ממוינת לפי ההגדרה.

שאלה 3:

הרעיון הכללי היא להגדיר מחלקה גינרית בתוך מחלקת SortedList בשם mod כך שתקבל בזמן הריצה את המספר שלפיו נחזיר את הרשימה החדשה. והפונקציה filter משתמשת באופרטור () שמחזיר את תוצאת החילוק והפתרון בכללי יראה כך:

```
template < class T >
class mod {
    T mod_number;
public:
    mod(T number) : mod_number(number) {}
```

```
bool operator()(T number) const {  
    return (number % mod_number) == 0;  
}  
};
```

```
SortedList<int> filter_list(SortedList<int>& t, mod<int> number){  
  
    return filter(mod<int>(number));  
  
}
```

