

### חלק יבש – תרגיל בית 3

#### 3.1 שאלות על SortedList

- הדרישות ההכרחיות שעל הטיפוס T לקיים:
  1. `> Operator`: למשל בפונקציה `insert` השוונו בין הערכים של האלמנטים כדי למיין את הרשימה בסדר יורד ולכן יש צורך שיהיה לאלמנט מטיפוס T אופרטור `<`.
  2. `Copy constructor`: כדי לאפשר אתחול של אלמנט חדש מטיפוס T על פי העתקת אלמנט שכבר קיים, למשל במימוש של `operator=` כאשר יצרנו רשימה ממוינת חדשה המכילה את השדות של `SortedList` קיימת.
  3. `Destructor`: כדי לאפשר שחרור זיכרון תקין של משאבים המוקצים עבור הטיפוס T.
- אם נפעיל את האיטרטור `non-const` על הרשימה ואופרטור `()*` יחזיר `T&` זה יאפשר גישה ושינוי של האיבר שנמצא ברשימה, וזה עלול להשפיע על סדר האברים ברשימה הממוינת ועל מבנה הנתונים שעשינו לכן השתמשנו ב `const` `iterator` שלא מאפשר שינוי ישיר של האברים אלא משנים רק דרך פקודות כמו `insert` ששומרות על הסדר.
- הפונקציה `filter` היא פונקציה טמפלטית שמקבלת פרדיקט ומפעילה אותו על אברי הרשימה. לכן הסטודנטית יכולה להשתמש בביטוי `lambda` ולכתוב פונקציה שבודקת אם המספר מתחלק במספר כלשהו. כך היא יכולה להעביר לפונקציה `filter` ביטוי `lambda` ואז ניתן לקבל ערך בזמן ריצה ולהעביר אותו לביטוי `lambda` דרך `capture_list`. אם מספר זה הוא 0 צריך לזרוק חריגה. כך נוכל לסנן את הרשימה לפי ערך שאינו ידוע מראש בזמן קומפילציה.