

DRY HW3

- 1 :** לא ניתן להסתפק בלהגדיר את פעולת האיטרטור הרגיל כקבועה מכיוון שתור יכול להיות קבוע או לא קבוע, ואם נרצה לעבור על תור רגיל (לא קבוע) ולשנות את איבריו לא נוכל לעשות זאת באמצעות `constIterator`.
- 2 :** אנחנו מניחים על הטיפוס ה `template` בבנאי, בבנאי העתקה, ב-`destructor`, באופרטור השמה ב-`pushback` ובכל הפונקציות שקוראות ל-`pushback`.
- כאשר אנו יוצרים `NODE` חדש עם `m_data` ב-`NODE` אנו מניחים שיש ל-`T` בנאי חסר ארגומנטים, וכל פונקציה ושדות שיאפשרו לנו להוסיף אותו, להחסיר אותו או להשוות אותו ל-`NODE` אחר.
- בנוסף באופרטור השמה אנו מניחים כי קיים אחד כזה עבור `T` גם מכיוון שאנחנו עושים השמה בין חוליות מטיפוס זה.
- בנוסף נשים לב שבפונקציות `filter`, `transform` אנחנו מניחים שלפונקציות הטמפלייטיות המתקבלות יש אופרטור סוגריים שמגדיר את הפעולה שלהן.
- 3 :** השגיאה מתרחשת בשלב ה `linking` והיא תהיה `undefined reference` כי הקומפיילר צריך להחזיק את כל המידע והמימוש של `templat` באותו קובץ בזמן הקומפילציה, ולכן הקומפיילר לא יכיר את הטיפוס `T` שבקובץ `CPP`. למשל אם יש לנו פונקציית גנרית מוגדרת ב-`Queue.h` וממומשת ב-`Queue.cpp` ושנקראת רק ב-`main.cpp`. כאשר `Queue.cpp` מתקמפל, זה לא ידוע בהכרח שהקומפילציה של `main.cpp` ידרוש מופע של `template`. אנחנו יכולים לשים מימוש של `template` ב-`cpp` רק אם השתמשנו בו רק באותו קובץ.
- 4 :** אם המספר ידוע רק בזמן ריצה. עלינו להכריז על `template` חדש, כי `templat` יקבל את הערך של `int S` רק בזמן הריצה ויעשה את הפעולה הנכונה. ככה :

```
template<int S>
bool isEven(int n)
{
    return (n % S) == 0;
}
```

Ruben Timsit
Itay Elkeslassy