

## מטלה - עץ בינארי

כיתבו מחלקה בשם Tree, המייצגת עץ-חיפוש בינארי לשמירת מספרים לפי סדר עולה, כפי שלמדתם בקורס מבני נתונים (לא חייב להיות עץ מאוזן). המחלקה צריכה לאפשר את הפעולות הבאות:

- insert - מקבלת כקלט מספר i ומכניסה את i לעץ במקום המתאים.
- remove - מקבלת כקלט מספר i ומסירה את i מהעץ.
- size - מחזירה את מספר המספרים בעץ.
- parent - מקבלת כקלט מספר i ומחזירה את המספר שנמצא מעל i בעץ.
- left - מקבלת כקלט מספר i ומחזירה את המספר שהוא הבן השמאלי של i בעץ.
- right - מקבלת כקלט מספר i ומחזירה את המספר שהוא הבן הימני של i בעץ.
- print - הדפסת העץ, לצורך ניפוי שגיאות. פורמט ההדפסה - לשיקולכם.

יש לזרוק חריגות במקרים הבאים:

- insert - כשמנסים להכניס מספר שכבר נמצא בעץ.
- delete - כשמנסים למחוק מספר שלא נמצא בעץ.
- בשאר הפקודות - לזרוק חריגה בכל מצב שנראה לכם חריג.

מצורף לתרגיל זה קובץ **main.cpp** עם תוכנית ראשית ובה כמה בדיקות - חלקן ידניות ( manual\_tests) וחלקן אוטומטיות (automatic\_tests). התוכנית הראשית חייבת להתקמפל ולרוץ כמו שהיא עם המחלקה שתכתבו.

**הגשה:** צרו מאגר (repository) חדש בגיטהאב והעלו לשם את הקבצים בתיקיה הראשית. הגישו בטופס-ההגשה קישור-שיבוט למאגר - הקישור שרואים כשלוחצים על הכפתור clone בגיטהאב.

**בדיקה אוטומטית:** אנחנו נבצע את הפקודות הבאות ממחשב עם לינוקס:

1. `git clone <הקישור שלכם>`
2. נעתיק לתוך התיקיה שלכם תוכנית ראשית משלנו, הכוללת בדיקות אוטומטיות נוספות וחישוב ציון.
3. `clang++-5.0 -std=c++17 main.cpp Tree.cpp`
4. `./a.out`

אתם יכולים לפתור את התרגיל בכל סביבת-פיתוח שאתם רוצים, אבל לפני ההגשה, וודאו שהפקודות האלו רצות בלי שגיאות על מחשב לינוקס אחר כלשהו. אם אין לכם clang תשתמשו ב-g++ במקום.

### דגשים (לבדיקה בתירגולים):

- יש לחזור על החומר של ההרצאות לפני שמתחילים לכתוב, ולהשתמש בו לפי הצורך.
- יש לתכנת בהתאם לכללים בסיסיים של הנדסת תוכנה, בפרט: חלוקה לקבצים, תיעוד בעברית או באנגלית, שמות משמעותיים למשתנים, בדיקות תקינות, בדיקות-יחידה (unit - test), ומניעת דליפת-זיכרון.
- לצורך כתיבת בדיקות-יחידה, ניתן להשתמש במחלקה TestCase המצורפת (כמודגם בתוכנית הראשית main.cpp), או בכל מערכת אחרת לבחירתכם.
- אם אתם לא מספיקים לממש את כל הפונקציות - ממשו חלק מהפונקציות בצורה נכונה ומלאה, כולל תיעוד ובדיקות, ובשאר הפונקציות שימו את הקוד המינימלי שיאפשר למערכת להתקמפל (זה נקרא "stub"). העדיפו מימוש בדוק לחלק מהפונקציות, על-פני מימוש לא-בדוק לכל הפונקציות.
- מותר להשתמש בתכונות מתקדמות של שפת C++ גם אם עדיין לא נלמדו בהרצאות.
- אין להעתיק תרגילים שלמים מסטודנטים אחרים. מותר להיעזר בקטעי קוד מהאינטרנט, אולם **יש לציין בבירור את המקור**, לוודא שהקוד עובד, ולוודא שאתם מבינים למה הוא עובד.