**מימוש עץ מחווט**

*\*נספח א' – הדפסה של קובץ התוכנית\**

*ידוע שבעח"ב בן n צמתים קיימים מצביעי left ו- right שערכם*

*במילים אחרות, חצי מהזיכרון המכיל את הקישורים מבוזבז.*

*נבצע את השינוי הבא לכל צומת z בעץ :*

*אם נותנים ל את הערך של .*

*אם נותנים ל את הערך של .*

*עץ זה נקרא עץ מחווט והקישורים החדשים נקראים חוטים.*

1. *ניתן להבדיל בין החוטים לבין מצביעים לבנים אמיתיים בשתי דרכים:*
2. ***דגלים*** *- הוספת 2 משתנים בוליאניים לכל צומת,*

*אשר מכילים אם מדובר במצביע לבן אמיתי, או אחרת.*

*יש לשים לב כי בדרך זו אנו נדרשים בזיכרון נוסף (הנוגד את עצם קיומו של עץ מחווט)*

*הגידול בזיכרון אומנם לינארי אך בפועל, זהו זיכרון של משתנה בוליאני ומספר הביטים שהוא דורש הינו קטן משמעותית מ"מצביע".*

1. ***שאלת "האם אני האבא של הבן שלי"***

*במקום לקרוא ישירות למשתנים ו- נייצר שגרות חדשות ו- .*

*שגרות אלו יבדקו אם מתקיים*

*או בהתאמה*

*אם מתקיים הנ"ל, אזי השגרה תחזיר או בהתאמה.*

*אחרת תחזיר . (משמע "אין לי בן ימני/שמאלי")*

1. *נממש את מבנה הנתונים הנ"ל באמצעות השיטה השנייה המוזכרת בסעיף א' (האם אני האבא של הבן שלי)*

*נייצר שגרות חדשות ו- שיעזרו לנו לקבל את הבנים האמתיים של כל צומת*

*ובנוסף נתחזק את החוטים כפי שמפורט בהמשך.*

*מבנה הנתונים תומך באפשרויות הנדרשות בשאלה ולהלן עיקרון האלגוריתם של כל אחת:*

* ***הכנסה*** 
  + *תתבצע הכנסה רגילה לעץ חיפוש בינארי*
  + *צומת שנכנסה בהכרח תהיה עלה, לכן "נחווט" את בניה לעוקב והקודם בהתאמה*
  + *נשים לב גם שעלה זה אינו יכול להיות עוקב/קודם של צומת אותה יש לחווט מחדש,*

*מכיוון שאם העלה שנכנס יהיה עוקב לצומת זו אז הוא יכנס כבן ימני, ואם יהיה קודם אז הוא יכנס כבן שמאלי, ולכן לא נדרש חיווט (כי לצומת יש בן אמיתי)*

* ***הוצאה***
  + *נבצע הוצאה רגילה (ע"פ הספר) של הצומת ובמקביל נתקן את החיווט ע"פ מקרים,*
    - *אם לצומת שהוצאנו אין בנים:*
      * *אם הוא בן ימני, אבא שלו יחווט לעוקב שלו*
      * *אם הוא בן שמאלי, אבא שלו יחווט לקודם שלו*
    - *אם לצומת שהוצאנו יש בן אחד:*
      * *האבא לא נדרש בחיווט חדש.*
    - *אם לצומת יש שני בנים*
      * *נבצע הוצאה של העוקב ע"פ שהאלגוריתם מוגדר בספר ולאחר מכן בדיקה של המקרים לעיל.*
  + *לאחר ההוצאה נבדוק האם היה צומת בעץ שחווט לצומת שהוצאנו, זאת ע"י בדיקה פשוטה אם הבן השמאלי של העוקב או הבן הימני של הקודם הינם בנים אמתיים.*

*אם לא, כלומר היה חיווט, נתקן אותו לצומת המתאימה לאחר ההוצאה.*

* ***חיפוש***
  + *ע"פ האלגוריתם הרגיל שמופיע בספר, אך עם שימוש בפונקציות העזר שהגדרנו בתחילת הסעיף כדי לקבל את הבן האמיתי.*
* ***החזרת עוקב והחזרת קודם***
  + *ע"פ האלגוריתם הרגיל שמופיע בספר, אך עם שימוש בפונקציות העזר שהגדרנו בתחילת הסעיף כדי לקבל את הבן האמיתי.*
  + *במקרה שהבן לא אמיתי ננצל את החוט ונקבל את מבוקשנו בזמן קבוע.*
* ***החזרת מינימום והחזרת מקסימום***
  + *ע"פ האלגוריתם הרגיל שמופיע בספר, אך עם שימוש בפונקציות העזר שהגדרנו בתחילת הסעיף כדי לדעת מתי אנו מגיעים לקצה העץ (מינימום בקצה השמאלי, מקסימום בקצה הימני)*

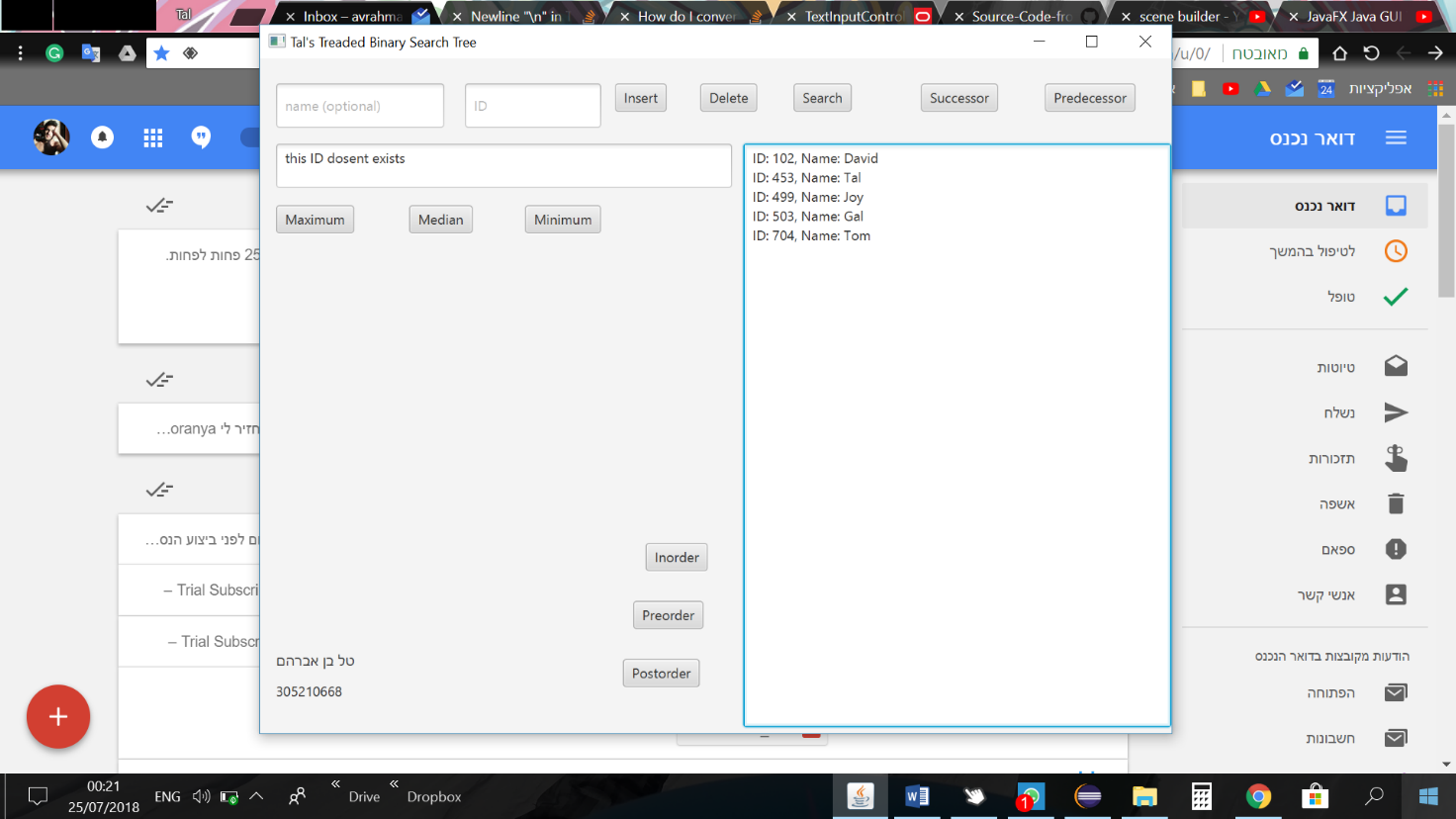
***בכל שגרה אנו מוסיפים לאלגוריתם המוכר של עח"ב מספר קבוע של פעולות, לכן זמן ריצה נשאר לינארי בגובה העץ .***

1. ***פעולות הסריקה***
   * *סריקה תחילית וסופית תבוצע ע"פ האלגוריתם הרקורסיבי הרגיל של עח"ב בזמן*
   * *בסריקה תוכית ננצל את חוטי העץ, נמצא את המינימום ולאחר מכן נדפיס את העוקב עד שנגיע לאיבר האחרון. אנו נעבור על כל איבר לכל היותר פעמים ולכן זמן הריצה נשאר*
2. ***החזרת חציון בזמן קבוע***
   * *נניח כי המפתחות שונים זה מזה וכאשר יש מספר זוגי של איברים נחזיר את החציון התחתון.*
   * *נשמור בעץ משתנה המצביע תמיד על החציון ומשתנה המונה את מספר הצמתים בעץ.*
   * ***תחזוקה בהכנסה***
     + *לאחר הכנסה נבצע תיקון של המצביע לחציון ע"י הפרדה למקרים*
       1. *אם מספר הצמתים זוגי והכנסנו איבר גדול מהחציון – החציון "יזוז למעלה"*
       2. *אם מספר הצמתים אי-זוגי והכנסנו איבר קטן מהחציון – החציון "יזוז מטה"*
     + *שאר המקרים האפשריים אינם גורמים להזזה של החציון.*
   * ***תחזוקה בהוצאה***
     + *מכיוון שבאלגוריתם ההוצאה שלנו תתכן העתקה של נתונים (מקרה של 2 בנים) הצבעה לחציון אינה בהכרח תהיה בתוקף לאחר מחיקה, לכן:*
     + *נשמור לפני מחיקה את המפתח של החציון ואת מפתחות העוקב שלו והקודם שלו.*
     + *לאחר מחיקה נדע מי החציון המעודכן ע"י הפרדה למקרים*
       1. *מספר הצמתים זוגי ואנו מוחקים את החציון או מתחתיו – החציון "יזוז מעלה".*
       2. *מספר הצמתים אי-זוגי ואנו מוחקים את החציון או מעליו – החציון "יזוז "מטה".*
     + *שאר המקרים אינם מזיזים את החציון.*
     + *כך נדע מיהו המפתח המהווה את החציון, נבצע את שגרת ה"חיפוש" למציאתו והצבעה מחדש.*
   * *פעולת התחזוקה מוסיפה במספר קבוע של פעולת מציאת עוקב/קודם - (ע"פ סעיף ב')*

*לכן פעולת ההכנסה וההוצאה נשארות בזמן ריצה לינארי בגובה העץ.*

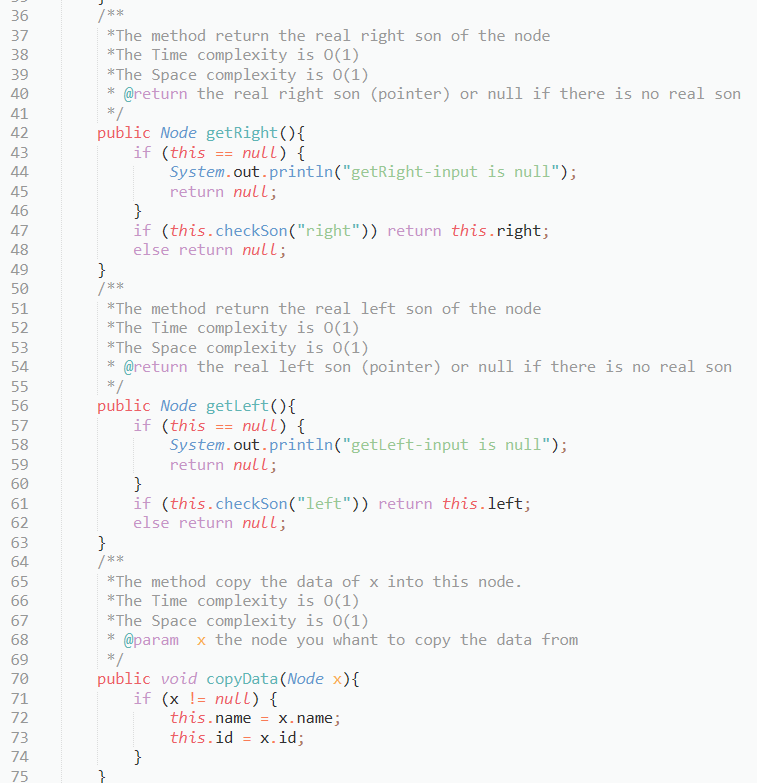
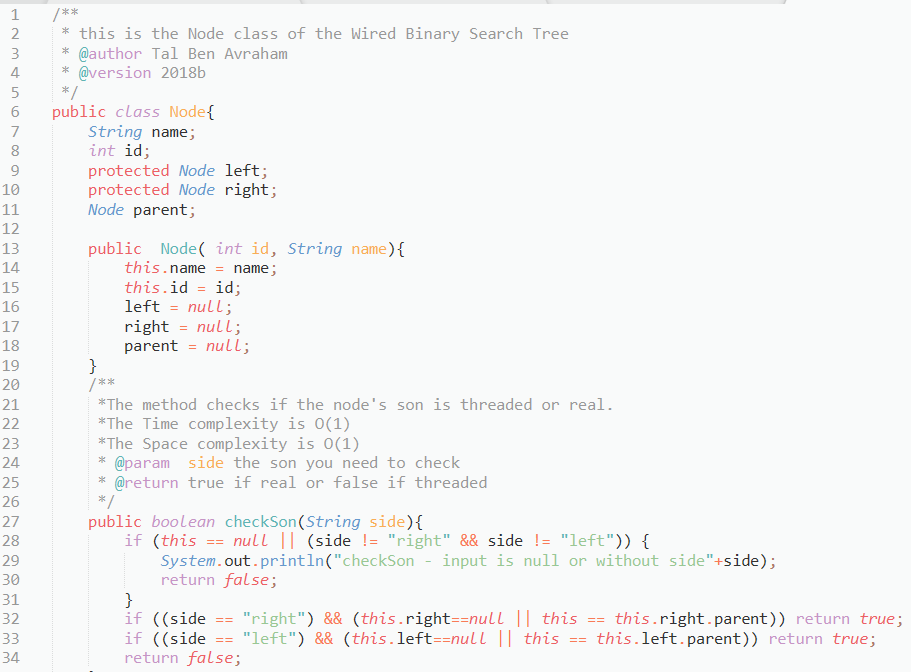
***ממשק GUI***

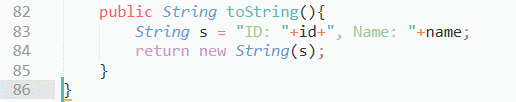
* *יש להריץ את RunMe על מנת שהממשק יופיע.*
* *בחלון הגדול יופיע פלט של סריקות העץ (כפי שניתן לראות את הסריקה התוכית בתמונה המצורפת)*
* *בחלון הצר יופיעו שאר ההודעות והפלטים (בתמונה ניתן לראות פלט של חיפוש מפתח שלא קיים בעץ)*
* *תעודת זהות היא חובה (המפתח) וניתן להזין "צמתים" ללא שם (name).*



*נספח א' – הדפסת קובץ התוכנית*

***מחלקת***





***מחלקת***

