

מבני נתונים

סמסטר חורף – 22/23 תשפ"ג

תרגיל בית מעשי

תאריך הגשה – 29/12/2022

**אנא קראו את המסמך בשלמותו לפני תחילת העבודה, ייתכנו עדכונים למסמך בימים
הקרובים בהתאם לצורך (הבהרות, שאלות נפוצות וכו')**

10/12/2022

דרישות התרגיל:

- בתרגיל זה עליכם לממש את מבנה הנתונים – "עץ AVL", במימוש זה, אשר עליו להיות תואם למימוש הנלמד בהרצאה ובתרגול, לצמתים בעץ אין מצביע להורה, לכן כלל הפעולות צריכות להיות ממומשות ברקורסיה. כלל המפתחות של הצמתים בעץ יהיו מספרים שלמים ויחודיים. העץ שלכם אמור להיות מתוכנן בכדי לענות על 2 שאלות עיקריות:
1. בהינתן 2 עצי AVL, נדרש לאחד אותם לעץ AVL אחד.
 2. בהינתן 2 ערכי מפתחות x ו y , החזר את סכום המפתחות בעץ שהינם בטווח $[x, y]$.
- a. לדוגמה – עץ עם המפתחות: 1,2,4,5,6,8,9,10, שאילתא Sum(3,7) תחזיר 15 (4+5+6) אנו ממליצים בחום שתחשבו ותתכננו, עוד לפני שתתחילו לתכנת, על פתרון יעיל ל 2 השאלות, בדגש על הפרטים הטכניים השונים.

הפעולות והשאלות של מבנה הנתונים הן:

1. **Insert(x)** – הוסף צומת חדש בעל מפתח x לעץ.
 - a. אם קיים כבר מפתח x בעץ – אל תעשה דבר.
 2. **Search(x)** – אם קיים מפתח x בעץ, החזר מצביע אל הצומת בעלת המפתח x .
 3. **Delete(x)** – מחק את הצומת בעלת המפתח x ושחרר את הזיכרון שהוקצה עבורו.
 - a. אם לא קיים מפתח x בעץ – אל תעשה דבר.
 4. **DeleteAll()** – מחק את כל העץ ושחרר את הזיכרון שהוקצה עבורו.
 5. **LCA(x,y)** – בהינתן 2 מפתחות עם הערכים x ו y , החזר את LCA של הצמתים המחזיקות במפתחות הנ"ל בעץ.
 - a. אם לא קיימים צמתים בעלי המפתחות x ו/או y , החזר את מצביע ל LCA של הצמתים שנמצאים במקום היכן ש x/y היו צריכים להיכנס (ללא תיקונים).
 6. **Sum(x,y)** – בהינתן 2 ערכים x ו y , החזר את סכום המפתחות הנמצאים בעץ אשר בטווח $[x, y]$.
 7. **Merge(T1,T2)** – בהינתן 2 מצביעים לעצי AVL שונים, מצביע לעץ AVL חדש המכיל את איחוד המפתחות של $T1$ ו $T2$.
- סיבוכיות הזמן הדרושה:
- Merge, DeleteAll - $O(n)$
- כל השאר ב $O(\log(n))$.
- שימו לב - פרטים טכניים נוספים בנוגע למימוש נמצאים בהערות בקבצי קוד של התרגיל.

קבצים מצורפים

בקבצי התרגיל, מצורף לכם קובץ avl.h ובו נמצאת הגדרת הרשומה (צומת בעץ) והחתימות של הפונקציות המתאימות לפעולות ושאלות שהגדרנו. קראו את הקובץ היטב והבינו מה הם הפרמטרים וערכי החזרה של כל אחת מהפונקציות. בנוסף, מצורף לכם קובץ avl.c כתוב חלקית, שבו עליכם לממש את הפונקציות הללו (ועוד פונקציות עזר כרצונכם). קראו את ההערות גם בקובץ זה ועקבו אחריהן. בשום מקרה אין להשתמש ברשומה אחרת מזו שנמצאת בקובץ avl.h ואין לשנות אותו.

קומפילציה

לפני שהתחלתם לתכנת, וודאו שבהגדרות סביבת העבודה שלכם (בין אם Visual Studio או Code::Blocks) הקומפילר הוא GCC ושהדגלים הבאים מודלקים :

- -Wall
- -Wextra
- -pedantic-errors
- -std=c11

בדיקת התרגיל

בקובץ avl.c מצויות כמה פונקציות בדיקה בסיסיות. הן בודקות שהעץ שמתקבל הוא אכן עץ AVL תקין ושכל הפעולות והשאלות עובדות כמו שצריך. יהיו פונקציות בדיקה נוספות שעליהן התוכנית שלכם תיבדק אך הן לא יפורסמו, כך שמומלץ לכתוב פונקציות בדיקה משלכם כדי לוודא שאין בתוכנית באגים והן עובדות כצפוי.

ניקוד

ציון ההגשה ייקבע לפי עמידת הקוד בטסטים אוטומטיים בלבד, בהינתן שאלתא/פעולה שנכשלת בטסט, הציון הניתן לאותה שאלתא/פעולה יהיה 0. עליכם לבדוק היטב את הקוד שלכם לפני ההגשה.

- תוכנית שאינה מתקמפלת או שקורסת לא תקבל נקודות.
- תוכנית שרצה בזמן של יותר מ 5 שניות לא תקבל נקודות.
- תוכנית שנכשלת בביצוע הכנסות ומחיקות או שהעץ שמתקבל אינו AVL לא תקבל נקודות. **ניקוד יינתן רק על שאלות שעברו את הבדיקות ולא כאלה שרק מומשו ונכשלו בבדיקות (טסטים אוטומטיים).**
- דליפת זיכרון תגרום להפחתה של 10 נקודות.

מימוש נכון של AVL בצורתו הבסיסית ביותר (Insert, Delete, Search, DeleteAll) הוא תנאי בסיס לקבלת ציון בתרגיל. בהינתן מימוש נכון כנ"ל הציון יתחיל מ-56 נק'. הציון על מימוש נכון של יתרת השאלות (שימו לב שלא ניתן לממש Sum מבלי LCA) הוא :

Merge – 10 pts. LCA – 10 pts. Sum+LCA – 34 pts.

הוראות כתיבה והגשה

- אין להשתמש בספריות נוספות מלבד stdio ו stdlib .
- אין לשנות את הגדרת הרשומה או להשתמש בכל רשומה אחרת מלבד זאת שמופיעה בקובץ avl.h .
- מלאו בראש הקובץ avl.c את השמות ואת תעודות הזהות שלכם בהתאם (ראו תמונה)

```

/****
Student1 name: -----
Student2 name: -----

Student1 ID: -----
Student2 ID: -----
****/

```

- יש להכין קובץ pdf בשם Documentation.pdf המתאר בקצרה כיצד פתרתם כל שאילתא
- יש להגיש אך ורק את avl.c נקי מפונקציות בדיקה/ פונקציית main ואת הקובץ Documentation.pdf בתוך קובץ ZIP/RAR עם הת"ז של המגישים בצורה הזו: HW_AVL_ID1_ID2.Zip. כאשר במקום ID1 ו ID2, הת"ז שלכם.
- הקוד שהוגש צריך להיות מתועד ונקי .
- הגשה שלא תעמוד בדרישות הנ"ל תוביל להורדת ניקוד.

מטרת התרגיל היא ללמוד ולתרגל את החומר מהכיתה, העתקות מכל סוג (חברים, הגשות משניים קודמות, קוד מהאינטרנט וכו') ייבדקו באופן אקטיבי בצורה אוטומטית וידנית - אם יימצאו העתקות הן יטופלו בחומרה!

טיפים מומלצים (לא חובה)

- התחילו מהפונקציות הקלות למימוש, בניית צומת חדשה, חיפוש, רוטציה, הכנסה וכן הלאה.
- ממשו את הפונקציות כך שיהיו רקורסיביות. אם הפרמטרים לא מתאימים לרקורסיה אז הוסיפו פונקציית עזר רקורסיבית .
- כתבו פונקציות עזר רבות. הקפידו שכל פונקציה תהיה קטנה מ 25 שורות .
- השתמשו בכך שהמצביעים לבנים של צומת נמצאים במערך, כדי לחסוך בקוד היכן שיש סימטריה בין ימין ושמאל .

הגשות באיחור והגשה חוזרת

ציוני תרגיל הבית יפורסמו כשבוע-שבועיים לאחר ההגשה,

עבור תוכניות שלא עברו קומפילציה או נכשלו, תינתן אפשרות להגשה חוזרת כשבוע לאחר פרסום הציונים, כאשר הציון המקסימלי של ההגשה החוזרת יהיה כ-80 נק' (כלומר הגשה שקיבלה 100% ניקוד – תקבל 80).

לא יינתנו אישורים להגשה באיחור, אנא תכננו את זמנכם בתבונה.

בונוסים –

3 התרגילים הכי מהירים (זמן ריצה של התוכנית) על הטסטים, יזכו לבונוס לציון של התרגיל (ייתכן ציון מעל 100). **הבונוס יינתן רק לתרגילים שמימשו הכל בהצלחה.**



מקום 1 – בונוס של 20 נק'

מקום 2 – בונוס של 15 נק'

מקום 3 – בונוס של 10 נק'

בהצלחה !