גליון יבש 2 – מבני נתונים

<u>מתרגל ממונה על התרגיל:</u> ברק גחטן barakgahtan@cs.technion.ac.il

<u>תאריך ושעת הגשה:</u>

<u>אופן ההגשה:</u> **בזוגות**

<u>הנחיות לפתרון:</u>

- <u>שאלות לגבי תרגיל הבית נא לשאול ב-Piazza של הקורס, מידע נוסף נמצא באתר.</u>
- בפורום הפיאצה ינוהל FAQ ובמידת הצורך יועלו תיקונים כ**הודעות נעוצות** (Pinned Notes). תיקונים אלו **מחייבים**.
- הגישו פתרון מוקלד. הגשה בכתב יד היא באישור המתרגל האחראי על התרגיל בלבד.
- הגשת התרגיל היא <u>אלקטרונית בלבד, באתר הקורס</u>, בקובץ PDF בלבד.
- בקשות להגשה מאוחרת יש לשלוח למתרגלת האחראית של הקורס – סאלי.
- הקפידו לכתוב את פתרונותיכם באופן מסודר ומובנה. התחילו מפתיח המתאר את תשובתכם בקצרה (עד 3 שורות) ולאחריו כתבו פירוט מלא של הפתרון. אי עמידה בכלל זה תגרור הורדת נקודות.
- הקפידו לצרף את כל השאלות והסעיפים לפי הסדר! אי עמידה בכלל זה תגרור הורדת נקודות.
- אין צורך לפרט דברים שנלמדו בהרצאות או בתרגולים. מספיק לצטט או להפנות לחומר הלימוד. עם זאת, יש להוכיח כל טענה שלא נלמדה בהרצאה או בתרגול.

שאלה 1

הוכח/הפרך כל אחד מהסעיפים הבאים.

1. לכל רשימת דילוגים רנדומלית S ולכל x, עבור זוג הפעולות $insert(x), delete(x)$ מתקיים שלכל $x \in S$ פעולת $delete(x)$ ואחריה $insert(x)$ אינן משנות את S.
2. לכל רשימת דילוגים רנדומלית S ולכל x, עבור זוג הפעולות $insert(x), delete(x)$ מתקיים שלכל $x \notin S$ פעולות $insert(x)$ ואחריה $delete(x)$ משנות את S.
3. יהא T עץ AVL, ויהיו T_L, T_R תתי העצים השמאלי והימני של שורש העץ.
   יהיו $|T_L|, |T_R|$ גודל תתי העצים בהתאמה - אזי, $|T_L| = \Theta(|T_R|)$.
4. סיבוכיות הזמן המשוערכת של פעולת העוקב $succ(x)$ בעץ חיפוש בינארי הינה $O(1)$, כאשר מתחילים מהצומת המינימלי בעץ ובכל פעם מפעילים את פעולת העוקב על האיבר האחרון שנמצא.
5. נגדיר את אורך המסלול הפנימי של עץ בינארי מלא להיות סכום עומקי כל הצמתים הפנימיים בעץ .
   נגדיר את אורך המסלול החיצוני של עץ בינארי מלא להיות סכום עומקי כל העלים בעץ.
   בעץ בינארי מלא בעל n צמתים פנימיים ובעל מסלול פנימי i ומסלול חיצוני e מתקיים $e = i + 2n$.

שאלה 2

א. הוסיפו למבנה הנתונים עץ 2-3 את הפעולה הבאה:

$\boldsymbol{Join}(\boldsymbol{T_1}, \boldsymbol{T_2})$: בהינתן שני עצי 2-3 בעלי n_1, n_2 מפתחות, בהתאמה, עבורם כל מפתח ב-T_1 קטן מכל מפתח ב-T_2, יש להחזיר עץ 2-3 שמכיל את איחוד קבוצות המפתחות בעצים T_1, T_2.

<u>סיבוכיות</u>: $O(\max\{\log(n_1), \log(n_2)\})$.

ב. נניח כי ידועים מראש גבהי העצים h_1, h_2, בהתאמה, וכן ידועים ערכי המינימום והמקסימום בכל אחד מהעצים T_1, T_2 (שימו לב: אין להניח כי בידכם מצביעים לאיברים אלו, אלא רק ערכם). הסבירו כיצד ניתן לממש את הפעולה $\boldsymbol{Join}$ מסעיף א'.

<u>סיבוכיות</u>: $O(|h_1 - h_2| + 1)$.

ג. יהא $T_1, T_2, \dots, T_k$ אוסף של k עצי 2-3, עבורם כל מפתח ב-T_i קטן מכל מפתח ב-T_{i+1}. נסמן ב-h_i את הגובה של העץ T_i. נניח כי כל הגבהים ידועים, ובנוסף ידועים לכל עץ ערכי המינימום והמקסימום שלו (לפי הנחות סעיף ב'). בנוסף, נניח כי $h_1 \le h_2 \le \cdots \le h_k$ וכי אין 3 עצים בעלי אותו גובה.

הראו כיצד ניתן לאחד את k העצים לעץ 2-3 יחיד, המכיל את איחוד המפתחות של k העצים.

<u>סיבוכיות</u>: $O(h_k - h_1 + k)$.

ד. הוסיפו למבנה הנתונים עץ 2-3 את הפעולה הבאה:

$\boldsymbol{Split}(\boldsymbol{T}, \boldsymbol{x})$: בהינתן עץ 2-3 אשר יסומן T ומפתח x, הפעולה מפצלת את העץ T לשני עצי 2-3 חוקיים T_1, T_2 כך ש-T_1 יכיל את כל המפתחות בעץ שקטנים או שווים ל-x, והעץ T_2 יכיל את כל המפתחות שגדולים ממש מ-x. ניתן להרוס את העץ T לצורך הפעולה.

<u>סיבוכיות</u>: $O(\log n)$.

שאלה 3

הגדרה – בהינתן שני מפתחות k_1, k_2 נגדיר $S(k_1,k_2)=\sum_{k_1\le k\le k_2} v(k)$, כאשר k עובר על כל הערכים במבנה שהם בין k_1 ל-k_2.

<u>סעיף א</u>:

יש להציג מבנה נתונים שתומך בפעולות הבאות:

$init()$	אתחול מבנה נתונים ריק.	$O(1)$
$insert(k,v)$	הוספת המפתח k בעל ערך v, שהוא מספר שיכול להיות חיובי, שלילי או אפס.	$O(\log n)$
$delete(k)$	מחיקת האיבר בעל מפתח k מהמבנה.	$O(\log n)$
$query(a,b)$	הפונקציה תחזיר את הערך המקסימלי האפשרי של $S(a',b')$ עבור $a\le a'\le b'\le b$, כאשר a',b' הם **מפתחות קיימים במבנה**.	$O(\log n)$

מובטח שקוראים ל-$query(a,b)$ רק עם קלט תקין, כלומר: באמת קיים במבנה מפתח $a\le k\le b$ ולכן המקסימום מוגדר היטב.

<u>סעיף ב</u>: יש לממש את מבנה הנתונים הבא עבור פרמטר k.

$init()$	כמו בסעיף א'	$O(1)$
$insert(k,v)$	כמו בסעיף א'	$O(k\log(k\log(n)))$
$delete(k)$	כמו בסעיף א'	$O(k\log(k\log(n)))$
$query(a,b)$	רוצים להחזיר ערך $S(a',b')$ שהוא k-מקסימלי, כלומר: $a\le a'\le b'\le b$ ושניהם מפתחות קיימים במבנה. וקיימים **בדיוק** $k-1$ זוגות (a'',b''), $a\le a''\le b''\le b$ שהם מפתחות קיימים במבנה, שעבורם $S(a'',b'')>S(a',b')$.	$O(k\log(k\log(n)))$

<u>הבטחות על קריאה ל-query(a,b)</u> (מותר למבנה להשתמש בהן, גם אם הפתרון לא נכון בלעדיהן).

מובטח שמספר זוגות המפתחות $a\le a'\le b'\le b$ הוא לפחות k ואין צורך לבדוק את זה.

מובטח שלכל $a_1\le b_1$ ו-$a_2\le b_2$, כך ש-a_1,b_1,a_2,b_2 כולם מפתחות קיימים במבנה, מתקיים:

$$S(a_1,b_1)=S(a_2,b_2)\ \rightarrow\ a_1=a_2,\quad b_1=b_2$$

ומובטח שלא קיימים במבנה מפתחות a_1,b_1 שעבורם $S(a_1,b_1)=0$.

שאלה 4

בשאלה זו נשמור סטטיסטיקות עבור ליגת הכדורגל הישראלית "ליגת האַל"

המבנה יחזיק שמות שחקני כדורגל ומידע על כמות השארים שהבקיעו במהלך העונה.

א. הציעו מימוש למבנה נתונים התומך בפעולות הבאות:

Init() אתחול מבנה ריק, ללא שחקנים.

סיבוכיות זמן: $O(1)$.

AddPlayer(first,last,id) הוסף למבנה שחקן חדש ששמו הפרטי הוא, first שם משפחתוlast ומספר מזהה id. לאחר ההוספה, לשחקן זה 0 שערים. הניחו כי לא ניתן להכניס למבנה שני שחקנים בעלי אותו שם פרטי ושם משפחה או אותו id (כלומר, אין צורך לבדוק שלא קיים שחקן אחר בעל שם זה או בעל id זה).

סיבוכיות זמן: $O(|first| + |last|)$, כאשר | ... | מציין את אורך המחרוזת.

RemovePlayer(first,last) הוצא מהמבנה את השחקן ששמו הפרטי הוא, first שם משפחתוlast .

סיבוכיות זמן: $O(|first| + |last|)$.

Goal(first,last) השחקן ששמו הפרטי first ושם משפחו last הבקיע שער.

סיבוכיות זמן: $O(|first| + |last|)$.

BestWithFirstName(first) החזר את ה id של המבקיע הטוב ביותר מבין השחקנים ששמם הפרטי first. במקרה של מספר מבקיעים כאלה החזר אחד מהם.

סיבוכיות זמן: $O(|first|)$.

BestWithLastName(last) החזר את ה id של המבקיע הטוב ביותר מבין השחקנים ששם משפחתם last. במקרה של מספר מבקיעים כאלה החזר אחד מהם.

סיבוכיות זמן: $O(|last|)$.

ב. הוסיפו למבנה את הפעולה הבאה מבלי לפגוע בפעולות הקודמות:

TopScorers(k) החזר את מספרם המזהה של k הכובשים המצטיינים (כלומר, בעלי מספר הבקעות הרב ביותר). במקרה של שוויון בין שחקנים, הפעולה תבחר מביניהם באופן שרירותי. למשל, אם במבנה 2 שחקנים בעלי אותו מספר הבקעות והפעולה נקראת עם $k = 1$, אזי ניתן להחזיר כל אחד משני השחקנים.

סיבוכיות זמן: $O(k)$