

שאלה 1 (25 נקודות)

בשאלה זו יש לתאר מבנה נתונים ששומר קבוצות של מספרים שלמים. כל קבוצה יכולה להכיל עד 10 מספרים. הקבוצות לא זרות (כלומר מספר מסוים יכול להופיע במספר קבוצות). השם של קבוצה הוא המספר הסידורי בו הוכנסה הקבוצה למבנה. מבנה הנתונים נדרש לתמוך בפעולות הבאות (n הוא מספר הקבוצות שנמצאות כרגע במבנה):

שם פעולה	תאור פעולה	זמן ריצה במקרה הגרוע
Init()	אתחול מבנה נתונים ריק.	$O(1)$
Add(S)	הוסף את הקבוצה S למבנה. הקבוצה S נתונה ע"י רשימה מקושרת של המספרים בקבוצה.	$O(\log n)$
Delete(i)	מחק את הקבוצה עם שם i. הניחו כי קבוצה עם שם i נמצאת במבנה.	$O(\log n)$
First(x,k)	החזר i מינימלי כך שקבוצה i מכילה את האיבר x, ומתקיים ש-i גדול ממש מ-k. הניחו כי איבר x נמצא במבנה. אם לא קיים i כנדרש יש להחזיר -1.	$O(\log n)$

מבנה נתונים יהיה מורכב מ-

- עץ AVL, T1, המסודר לפי המספר הסידורי של הקבוצה. כל צומת בעץ מכיל כמפתח את מספר הקבוצה, i. בנוסף כל צומת שומר רשימה מקושרת של המספרים ששייכים לקבוצה.
- עץ AVL, T2, ששומר את קבוצת כל האיברים (מספרים) שמופיעים בכל הקבוצות (איבר שמופיע במספר קבוצות נשמר בעץ רק פעם אחת). מפתח של הצומת הוא מספר x. בנוסף, לכל צומת בעץ קיים מצביע לעץ AVL שמכיל את כל מספרי הקבוצות המכילות את x. כיוון שבכל קבוצה יש לכל היותר 10 איברים, אז העץ T2 מכיל לכל היותר 10n צמתים.
- כמו כן, נחזיק מונה בשם counter של מספר הקבוצות שהוכנסו למבנה.

פעולות:

Init()
T1 ← NULL
T2 ← NULL
counter ← 0

זמן ריצה $O(1)$.

Add(S)

$counter \leftarrow counter + 1$

הכנס לעץ T1 צומת בעל מפתח counter, שזה מספר הסידורי של הקבוצה החדשה S. עבור כל $x \in S$, חפש את x בעץ T2.

- אם x נמצא בעץ T2, אז הוסף את מספר הקבוצה S לעץ ה-AVL שבצומת של x.
- אחרת, הוסף את x לעץ T2, וצור בצומת של x עץ AVL שמכיל איבר בודד, שהוא מספר הקבוצה S.

כיוון שב-S יש לכל היותר 10 איברים הזמן הוא $O(\log n)$.

Delete(i)

חפש את קבוצה עם מספר סידורי i בעץ הקבוצות, נקרא לה S. זמן ריצה: $O(\log n)$.
עבור כל $x \in S$ (לפי הרשימה ששמורה בצומת), חפש צומת בעץ T2 שמכיל מפתח x.
מחק את i מעץ ה-AVL שבצומת של x. אם לאחר מחיקה זו העץ הפך להיות ריק, אז מחק את x מעץ T2.
כיוון שכל קבוצה מכילה לכל היותר 10 מספרים, זמן ריצה הכולל הוא $O(\log n)$.

First(x,k)

חפש את x בעץ T2. בעץ AVL ששמור בצומת x בצע חיפוש של i בדרך הבאה: אתחל משתנה בשם min לערך אינסוף. התחל בשורש. כאשר נמצאים בצומת v עם מפתח i, נשווה בין i ו-k.

1. אם $i \leq k$ נרד לבן הימני של v, אם הוא קיים.

2. אם $i > k$ בצע:

a. אם $i < \min$ הצב $i \leftarrow \min$.

b. רד לבן השמאלי של v, אם הוא קיים.

כאשר לא ניתן להמשיך לרדת בעץ, אם הערך של min הוא לא אינסוף, החזר min. אחרת החזר -1.

בס"ד הנדסאים שנה ב' תרגילים נוספים במבנה נתונים לקראת המבחן החיצוני
 2. נכון או לא נכון: לאחר הכנסת איבר x לעץ חיפוש בינרי, האב של x הוא או העוקב או הקודם של x בעץ.

נכון

שאלה 3

נתונה ערימת מקסימום הממומשת ע"י מערך. תאר אלגוריתם שבהינתן ערך x , מדפיס את כל אברי הערימה שגדולים או שווים ל- x . זמן הריצה של האלגוריתם צריך להיות $O(m)$ כאשר m הוא מספר האיברים שהודפסו.

Print_Bigger(A,x,i)
if $i \leq A.size$ and $A[i] \geq x$
Print A[i]
Print_Bigger(A,x,2i)
Print_Bigger(A,x,2i+1)
כאשר הדפסת האיברים שגדולים או שווים ל- x תבצע ע"י קריאה ל- Print_Bigger(A,x,1) .
אם מספר האיברים המודפסים הוא m אז מספר הקריאות ל- Print_Bigger הוא לכל היותר $2m+1$.
לכן זמן הריצה הוא $O(m)$.

ד. נתון מערך A המכיל n מספרים שלמים. בהינתן מספר x המופיע ב- A , נגדיר את הריבוי של x ב- A בתור מספר המופעים של x ב- A . הציעו אלגוריתם שמחזיר מערך ממויין לפי ריבויים B שיכיל את הריבוי של כל אחד מהמספרים שמופיעים ב- A , ויעבוד בזמן צפוי של $O(n)$.
 דוגמא: עבור מערך קלט
 $A = [32, 28, 75, 32, 28, 32, 12, 12]$
 האלגוריתם צריך להחזיר את מערך הפלט:
 $B = [1, 2, 2, 3]$
 (הריבוי של 75 הוא 1, של 12 הוא 2, של 28 הוא 2, ושל 32 הוא 3).

נשתמש בטבלת גיבוב בגודל n , עם שירשור, כך שכל חוליה ברשימה המקושרת תכיל מספר וריבוי, ואם יש שני מספרים זהים רק נגדיל את הריבוי.

ככה נקבל בזמן $O(n)$ צפוי את כל הריבויים של כל n המספרים

נשים לב שריבוי הוא מספר בין 1 ל- n ולכן ניתן לבצע מיון מניה בזמן לינארי

שאלה 1

א. (8 נקודות) נתונה ערימת מינימום הממומשת על-ידי מערך. כל האיברים בערימה שונים זה מזה. נניח שלכל קודקוד בערימה, בנו השמאלי קטן מבנו הימני. האם המערך ממוין?
לא, דוגמא נגדית: 1,2,10,4,5 (במערך)

ו. (8 נקודות) שאלה זו עוסקת בניתוח פחת (amortized analysis). דויד הציע את מבנה הנתונים הבא, המבוסס על עץ חיפוש בינארי: אתחול המבנה מתבצע ע"י בניית עץ חיפוש בינארי ריק (לא עץ AVL). ניתן להניח שבמבנה נתונים זה אין מחיקות. ההכנסה היא הכנסה רגילה של עץ חיפוש בינארי, אך כאשר מבצעים את ההכנסה למבנה יהא n מספר האיברים במבנה, אם n חזקה של 2 (בכל 2^i הכנסות, לכל $i > 3$ שלם), בונים עץ חיפוש

בינארי מושלם (שוב, לא עץ AVL) מכל האיברים שהוכנסו עד עכשיו, ומוחקים את העץ הקודם. מה יהיה זמן הריצה עבור סדרה של n הכנסות למבנה במקרה הגרוע?

$O(n^2)$ התקבל גם Amortized $O(n)$
<u>הסבר</u> : - סידור מחדש של עץ חיפוש בינארי לעץ מושלם $O(n)$.
- מספר הפעמים שסידור זה מתבצע - $O(\log n)$
- נחשב לכל רצף הכנסות $2^i - 1 \dots 2^{i+1} - 1$ את זמן הריצה שלהן.
- הכנסה ל-BST: מתנאי השאלה עץ ה-BST עם $2^i - 1$ איברים מתחיל כעץ מושלם. במקרה הגרוע ביותר גובה העץ יגדל ב-1 בכל הכנסה. למשל: כאשר תמיד נכנס איבר שיותר גדול מכל האיברים בעץ. במקרה כזה, זמן הריצה יהיה: עבור $k = 2^i$
$[\log(k) + 1] + [\log(k) + 2] + \dots + [\log(k) + k] = k * \log k + \frac{k * (k + 1)}{2}$
כלומר, במקרה הגרוע ביותר, עבור $k \approx O(n)$, זמן הריצה להכנסה של איבר אחד לעץ יהיה $O(n)$.
ובסה"כ, לכל האיברים: $O(n^2)$.

ז. (12 נקודות) נתון עץ AVL בעל n מפתחות. מרחק בין שני קודקודים בעץ מוגדר להיות מספר הצלעות במסלול הקצר ביותר ביניהם בעץ.

1. מה המרחק בעץ בין האיבר השני בגודלו לאיבר השלישי בגודלו, במקרה הגרוע ביותר? בסעיף זה, תשובה אסימפטוטית ב- Θ תזכה ב-75% מהניקוד.
האיבר השני בגודלו מכיל בתת העץ השמאלי שלו איבר אחד (האיבר הראשון). מתכונת איזון הגבהים,

האיבר השלישי בגודלו, הנמצא בתת העץ הימני של האיבר השני, יכול להיות לכל היותר במרחק 2.

2. מהו המרחק בעץ בין לאיבר ה- i בגודלו לאיבר ה- $i + 1$ בגודלו, במקרה הגרוע ביותר? בסעיף זה, תשובה אסימפטוטית ב- Θ תזכה ב-100% מהניקוד.

אחד האיברים הוא בהכרח אב קדמון של השני. נניח שהאיבר ה- i הוא האב הקדמון:
בתת העץ השמאלי שלו נמצאים לכל היותר $\Theta(i)$ איברים בגובה של לכל היותר $\Theta(\log(i))$. מתכונת
איזון הגבהים, האיבר ה- $i + 1$ בגודלו, הנמצא בתת העץ הימני של האיבר ה- i , יכול להיות לכל היותר
במרחק $\Theta(\log(i))$.
ניתן לטעון אותו דבר הפוך במידה והאיבר ה- $i + 1$ הוא האב הקדמון.
התשובה $\Theta(\log(i))$ קיבלה בונים 2 נקודות.
התשובה $\Theta(\log(n))$ קיבלה את מלוא הנקודות.

להלן שאלות מרובות בחירה. תיתכן יותר מתשובה אחת נכונה לכל שאלה. יש להקיף בעיגול את כל התשובות הנכונות.

- 1.1. נתון עץ AVL כך שבכל קודקוד v , בנוסף למפתח, נמצא שדה $size$, המכיל את גודל תת העץ ששורשו v . מבצעים סריקת Post-order על העץ, אך במקום להדפיס את המפתח, מדפיסים את ערכי ה- $size$. אילו מהמשפטים הבאים נכונים תמיד? (ניתן להניח כי בעץ יש לפחות שלושה איברים)
- המספר הראשון שיודפס הוא 1.
 - המספר האחרון שיודפס הוא הגדול ביותר.
 - המספרים יודפסו בסדר מונוטוני עולה.
 - סכום המספרים הלפני-אחרון והלפני-לפני-אחרון שיודפסו קטן באחד מהמספר האחרון שיודפס.

- 1.2. נתונה טבלת Hash בגודל m בה פותרים התנגשויות על ידי שרשור. הטבלה תומכת בפעולות: Search ו-Insert. עבור סדרה של N פעולות, המורכבת ממספר שווה של פעולות Insert ופעולות Search, ומתחילה מטבלה ריקה, זמן $amortized\ worst-case$ לפעולה בודדת הוא:

- $\Theta(1)$
- $\Theta(m)$
- $\Theta(m^2)$
- $\Theta(N)$
- $\Theta(N^2)$

בחברה ליבוא אופנים "בייק בורדו" נדרש מבנה לניהול משכורת העובדים אשר תומך בפעולות הבאות :

שם פעולה	תאור פעולה	זמן ריצה במקרה הצפוי	זמן ריצה במקרה הגרוע
$Init(N)$	אתחול מבנה נתונים. N הינו מספר מקסימאלי של עובדים שיכולים להיות במבנה בכל רגע נתון.	ללא דרישה	$O(N)$
$Insert(id, salary)$	הכנסת עובד חדש בעל תעודת זהות id משכורת $salary$ למבנה. ניתן להניח שעובד בעל ת"ז id לא נמצא במבנה ואין צורך לבדוק זאת.	$O(1)$	$O(\log(n))$
$GetSalary(id)$	החזרת משכורת של עובד בעל תעודת זהות id . הערה: מותר להניח שקיים כזה.	$O(1)$	$O(\log(n))$
$SetSalary(id, k)$	הזנת משכורת בעלת ערך k לעובד עם תעודת זהות id . הערה: מותר להניח שקיים כזה.	$O(1)$	$O(\log(n))$
$SetAllSalary(k)$	הזנת משכורת בעלת ערך k לכל העובדים אשר נמצאים במבנה.	ללא דרישה	$O(1)$

• $n \leq N$ – הוא מספר האיבר במבנה ברגע נתון

תארו בקצרה את מבנה הנתונים, ספקו אלגוריתם לכל אחת מהשיטות ונתחו בקצרה את זמן הריצה של הפעולות.

תיאור המבנה:

נשתמש בטבלת גיבוב H בגודל $O(N)$ הפותרת את ההתנגשויות בעזרת שרשר, אך במקום רשימה מקושרת

כל כניסה בטבלה תצביע לשורש של עץ AVL. המפתחות בעץ יהיה מספרי הזהות של העובדים.

בנוסף, נתחזק שלושה משתנים לטובת עדכון המשכורות: מונה $time$, משתנה $global_salary_time$

ומשתנה $global_salary$, כולם יאותחלו ל-0.

כל קודקוד בעץ AVL יכיל את השדות הבאים: $salary_time$, $salary$, id .

תיאור הפעולות:

$Init(N)$ - איתחול טבלת גיבוב H . איחול משתנים $time$, $global_salary_time$ ו- $global_salary$ ל-0.

זמן ריצה הוא $O(N)$

$Insert(id, salary)$ – חישוב אינדקס בטבלת הגיבוב, ולאחר מכן הכנסת איברי לעץ AVL המתאים

בס"ד הנדסאים שנה ב' תרגילים נוספים במבנה נתונים לקראת המבחן החיצוני
(לפי המפתח id). השדה salary_time יעודכן לזמן הנוכחי time, ולאחר מכן נבצע time++.

זמן ריצה: $O(1)$ צפוי (תחת ההנחה של גיבוב אחיד ופשוט מקדם העומס הוא קבוע), $O(\log n)$ במקרה הגרוע (הכנסה לעץ AVL).

$GetSalary(id)$ – חישוב אינדקס בטבלת הגיבוב וחיפוש בעץ המתאים. נבצע בדיקה האם

$salary_time > global_salary_time$. אם כן, נחזיר את ערך שדה salary שבצומת, אחרת נחזיר את $global_salary$.

זמן ריצה: $O(1)$ צפוי (תחת ההנחה של גיבוב אחיד ופשוט מקדם העומס הוא קבוע), $O(\log n)$ במקרה הגרוע (חיפוש בעץ AVL).

$SetSalary(id, k)$ – חיפוש איבר בעל מפתח id בטבלה. השדה salary_time יעודכן לזמן הנוכחי

time, ולאחר מכן נבצע time++.

זמן ריצה: $O(1)$ צפוי (תחת ההנחה של גיבוב אחיד ופשוט מקדם העומס הוא קבוע), $O(\log n)$ במקרה הגרוע (חיפוש בעץ AVL).

$SetAllSalary(k)$ – נעדכן את $global_salary$ לערך החדש.

$Global_salary_time$ יעודכן לזמן הנוכחי time, ולאחר מכן נבצע time++.

זמן ריצה: $O(1)$ במקרה הגרוע

1. נתון מערך A המכיל n מספרים שלמים. רוצים לבדוק האם מערך זה מייצג ערימת מקסימום.
השלימו את התשובה: זמן ריצה של האלגוריתם היעיל ביותר לבדיקה הנ"ל במקרה הגרוע הוא $\Theta(n)$.

2. יהיה C אלפבית בגודל $n > 10$ (כלומר, ישנם n תווים שונים).
טענה: הדחיסה המרבית בקידוד Huffman עבור C תתקבל כאשר שכיחויות של כל התווים זהות.
סמנו את התשובה הנכונה.

א. הטענה נכונה

ב. הטענה לא נכונה

סעיפים ב', ה' לא בחומר

3. נתונים n מספרים. רוצים לבנות מהם מבנה נתונים.

עבור כל אחד מהמבנים הבאים רשמו מהו זמן בניה שלו במקרה הגרוע.

א. טבלת Hash שבה פותרים התנגשויות על ידי רשימה מקושרת $O(n)$

ב. טבלת Hash שבה פותרים התנגשויות בשיטת open addressing $O(n^2)$

ג. ערימת מקסימום $O(n)$

ד. עץ AVL $O(n \log n)$

ה. עץ BB[α] עם $\alpha = 1/3$ $O(n \log n)$

בס"ד הנדסאים שנה ב' תרגילים נוספים במבנה נתונים לקראת המבחן החיצוני

4. יהא T עץ AVL המכיל n מפתחות שונים זה מזה. יהא v הצומת שמכיל את מפתח המינימום בעץ T .

אז גובהו* של v יכול להיות : (סמנו את כל התשובות הנכונות).

א. 0

ב. 1

ג. 2

ד. $\Theta(\log n)$

ה. $\Theta(n)$

* תזכורת: גבהו של צומת מוגדר כאורך המסלול (בקשתות) הארוך ביותר ממנו לעלה. לדוגמה- גובהו של עלה הינו 0.

התבקשתם לממש מבנה נתונים לניהול קבוצת העובדים במפעל כל עובד מזוהה ע"י ת"ז שלו שהוא מספר ייחודי. כמו-כן לכל עובד מצורף מספר המציין את המשכורת החודשית שלו. ייתכנו מספר עובדים בעלי משכורת זהה. **סיבוכיות הזיכרון הנדרשת הינה $O(n)$** , כאשר n הינו מספר העובדים במבנה ברגע נתון. על המבנה לתמוך בפעולות הבאות:

שם פעולה	תאור פעולה	זמן ריצה
Init()	אתחול מבנה נתונים ריק.	$O(1)$ במקרה הגרוע
Insert(e)	הוספת עובד חדש e למבנה. e מכיל שני שדות: ת"ז $e.id$ ומשכורת $e.salary$. הניחו כי e עדיין לא נמצא במבנה ואל תבדקו זאת.	$O(\log n)$ לשיעורין (amortized)
Salary(id)	החזרת מספר המציין משכורת של העובד בעל ת"ז id . הניחו כי עובד בעל ת"ז id נמצא במבנה ואל תבדקו זאת.	$O(1)$ צפוי
Delete(id)	מחיקת עובד בעל ת"ז id מהמבנה. הניחו כי עובד בעל ת"ז id נמצא במבנה ואל תבדקו זאת.	$O(\log n)$ לשיעורין (amortized)
Quarterian()	החזרת משכורת שרק ל- $\lfloor \frac{n}{4} \rfloor$ מהעובדים יש משכורת גדולה או שווה לה. (כלומר, יש להחזיר משכורת כך שרק ל-25% מהעובדים יש משכורת גדולה או שווה לה).	$O(1)$ במקרה הגרוע

- n – מספר העובדים במבנה ברגע נתון
- ת"ז הינו מספר שלם חיובי

תארו בקצרה את מבנה הנתונים, ספקו אלגוריתם לכל אחת מהשיטות ונתחו בקצרה את זמן הריצה של הפעולות.
קיימים מספר פתרונות. להלן אחד מהם.

מבנה נתונים יהיה מורכב מ-

- עץ AVL, T1, בו מפתחות הם משכורות (salary) של העובדים. בכל צומת נוסף שדה num המציין את כמות העובדים בעלי משכורת זו. בעת ההכנסה ל-T נבצע חיפוש ונבדוק האם e.salary כבר נמצאת בעץ ואז רק נוסף 1 לשדה num של הצומת המתאים. אם e.salary עדיין לא נמצאת בעץ, נכניס צומת חדש ונעדכן שדה num שלו להיות 1.
בנוסף, כל צומת בעץ יכיל שדה size – מספר העובדים בתת עץ המושרש בצומת (שדה size סופר גם כפילויות של משכורת, כלומר מחשיב את השדות num בתוכו). נשתמש בשדות אלו על מנת לתמוך בפעולה Quarterian.
- טבלת גיבוב, H, שתכיל את משכורות העובדים לפי ת"ז של העובדים. הטבלה פותרת התנגשויות בעזרת שרשור. המפתחות בטבלה יהיו ת"ז של העובדים.
- עץ AVL, T2, שהמפתחות בו הם id של העובדים. שימוש בעץ זה יהיה בפעולה Delete.
- מצביע, S, לצומת בעץ T1 המצביע לאיבר שיש $\lfloor \frac{n}{4} \rfloor$ משכורות שגדולות או שוות לו.

בס"ד הנדסאים שנה ב' תרגילים נוספים במבנה נתונים לקראת המבחן החיצוני
(5) n מספר המפתחות במבנה.

יש לתאר כל פעולה בפירוט.