דף סיכום בחינה

מזהה קורס: 20407   שם קורס: מבני נתונים ומבוא לאלגוריתמים

מספר שאלה	ציון מירבי	ציון שאלה סופי
1.1	9.00	8.00
1.2	8.00	8.00
1.3	8.00	8.00
2.1	8.00	8.00
2.2	8.00	8.00
2.3	9.00	9.00
3.1	5.00	
3.2	8.00	
3.3	12.00	
4.1	5.00	5.00
4.2	5.00	5.00
4.3	5.00	3.00
4.4	5.00	5.00
4.5	5.00	5.00
5	25.00	25.00

ציון בחינה סופי : 97.00

הבחינה הבדוקה בעמודים הבאים

מבחן במועד׳ : 16/07/2020

שאלה 1:

סעיף א׳:

נפתור את נוסחת הנסיגה בעזרת שיטת האיטרציה:

$$T(n) = T(n-1) + \log n = \log n + \log(n-1) + T(n-2) =$$

חוקי לוגריתמים

$$= T(n-2) + \log(n(n-1)) = \dots =$$

חוקי לוגריתמים

$$= T(n-i) + \log(n(n-1)\cdot\ldots\cdot(n-k)) = \log(n!) = \Theta(n\lg n)$$

מדוע? לא השלמת את הפיתוח.

משאלה ב-5 בעמ׳ 34 במדריך הלמידה

(1.1)

סעיף ב׳:

נקבע את היחס האסימפטוטי בין $n!$ לבין $(2n)!$:

נגדיר $f(n)=(2n)!$ ו- $g(n)=n!$ ונחשב:

$$\lim_{n\to\infty}\frac{f(n)}{g(n)} = \lim_{n\to\infty}\frac{(2n)!}{n!} = \lim_{n\to\infty} 2n\cdot(2n-1)\cdot\ldots\cdot(n+1) = \infty$$

(1.2)

לפיכך קיבלנו שהיחס האסימפטוטי הוא $f(n)=\omega(g(n))$, כלומר:

$$(2n)! = \omega(n!)$$

סעיף ג׳:

נוכיח לפי ההגדרה כי: $f(n)+O(f(n)) = \Theta(f(n))$:

נמצא קבועים c_1, c_2 ו- n_0 כך שלכל $n>n_0$: $0 \le c_1 f(n) \le f(n)+O(f(n)) \le c_2 f(n)$ ⊛⊕

נגדיר את הקבועים $c_1=\frac{1}{2}$, $c_2=2$ ו- n_0 המוגדר ב-⊛ ונקבל:

$$0 \le \frac{1}{2}f(n) \le f(n) \le f(n)+O(f(n)) \le f(n)+f(n) = 2f(n)$$

⊛ ההגדרה של O, בעמ׳ 41 בספר, לכל $c>0$ קיים קבוע $n_0>0$. נבחר $c=1$

לכן קיים קבוע n_0

כלומר הראינו שעבור הקבועים שנתנו מתקיים ⊛⊕ ולכן מההגדרה נסיק כי:

$$f(n)+O(f(n)) = \Theta(f(n))$$

סעיף ג׳ מופיע בדף הבא

סעיף ג:

נוכיח ע"פ ההגדרה כי: $f(n)+o(f(n))=\Theta(f(n))$:

נגדיר $g(n)=o(f(n))$, לכן מההגדרה נקבל:

לכל $c>0$ קיים $n_0>0$ כך ש- $g(n)<cf(n)$.

לפיכך נבחר $c=1$, לכן קיים $n_0>0$ כך ש- $g(n)<f(n)$, כלומר:

$$g(n)=o(f(n)) \Rightarrow g(n)<f(n) \Rightarrow f(n)+f(n)\le 2f(n)$$

לכן מהגדרת O נקבל כי: $f(n)+o(f(n))=O(f(n))$ (עבור n_0 וקבוע 2)

מצד שני מתקיים כי $0<o(f(n))$, לכן $f(n)<f(n)+o(f(n))$

כלומר $f(n)+o(f(n))=\Omega(f(n))$ לכן ממשפט 3.1 נסיק כי:

$$f(n)+o(f(n))=\Theta(f(n))$$

(1.3)

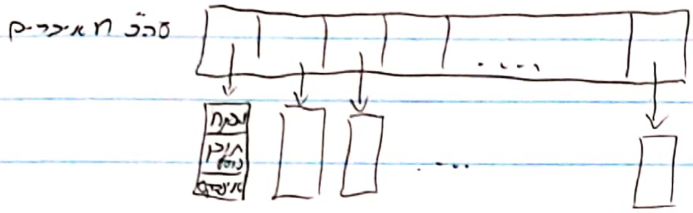
שאלה 2:

סעיף א':

נסביר כיצד ניתן לשנות כל אלגוריתם מיון כך שיהיה יציב:

מיון יציב נקרא כך אם הוא אינו משנה את סדר היחסי של איברים הזהים זה לזה. לפיכך כדי לשמור על תכונה זו נצטרך למנוע שינוי בסדר האיברים הזהים בעת ביצוע המיון.

כדי לפתור בעיה זו נעבור על הקלט שהתקבל (נניח מערך) ונמיר אותו באופן הבא: כל איבר יהפוך לרשימה שבה יוחזק המפתח, הערכים הנלווים למפתח, והאינדקס של האיבר לפני ביצוע השינוי של הקלט. לדוגמה:



הסיכת הקלט שהתקבל לצורה שכתבנו לעיל דורשת מעבר על הקלט, לכן סיבוכית $O(n)$.

כעת במהלך המיון שנבצע (אלגוריתם מיון נתון) נבצע אותו באופן זהה פרט למצב בו האיברים זהים, כאשר מכניסים שני איברים זהים אחד ליד השני, נבצע בדיקה של האינדקסים (של השדה הנוסף שהוספנו לכל איבר) המקוריים ובמידה והסדר השתבש במהלך המיון נחליף בין שני האיברים, כך שהאינדקס המקורי הקטן יופיע לפני האינדקס המקורי הגדול. פעולה זו מתבצעת ב-$O(1)$ (החלפת שני איברים סמוכים).

לסיכום נקבל שהשינוי יוסיף לאלגוריתם הנתון מעבר בתחילת המיון ב-$O(n)$, אך מיון השוואות הוא $O(n\lg n)$ ומיון לא מבוסס השוואות הוא $O(n)$ ולכן בעצם לא נשנה את סיבוכיות המיון והיא תישאר זהה.

8 (2.1)

סעיף ב':

נתון מערך של n מספרים שלמים ושונים זה מזה בתחום $[-n...n^2]$. נכתוב אלגוריתם למיון המערך בזמן ליניארי:

תחילה נעבור על המערך ולכל איבר נוסיף את הערך n. לפיכך לאחר המעבר על המערך נקבל שכל האיברים במערך בתחום $[0...n^2+n]$. מעבר זה מתבצע ב-$O(n)$ שכן עוברים על כל איבר במערך.

כעת נעזר בשאלה 8-3 שבמדריך הלמידה, כלומר נבצע מיון בסיס על n מספרים שלמים השונים זה מזה בתחום $[0...n^2+n]$ (כלומר זהו מקרה פרטי יותר של $K=3$, כלומר $[0...n^3-1]$). לפיכך נקבל שהמערך כעת ממוין בעזרת מיון בסיס (כלומר $O(3n)$ ולכן זמן הריצה הוא $O(n)$. (סיבוכיות $O(n)$)

לבסוף נעבור שוב על כל האיברים שמערך ונחסיר את הערך n מכל איבר. ונקבל שכל האיברים שוב בתחום $[-n...n^2]$. לסיכום קיבלנו מערך ממוין בתחום $[-n...n^2]$

8 (2.2)

המשך סעיף ב:

כאשר סיבוכיות זמן הריצה הכוללת היא $O(n)$.

סעיף ג:

יהי A מערך ממוין מהסעיף הקודם. נכתוב פסאודוקוד למציאת אינדקס i המקיים $A[i]=i$. אם אין אינדקס כזה נחזיר -1. זמן ריצה $O(\lg n)$:

האלגוריתם שנכתוב הוא מבוסס על חיפוש בינארי עם שינויים קלים, רק שנבדוק בכל איטרציה האם $A[i]=i$. אם לא נמצא אינדקס שכזה, נחזיר -1:

```
FINDINDEX(A[1..n])
1)  start ← 1
2)  end ← length[A]
3)  while start < end
4)      mid ← ⌊(start+end)/2⌋
5)      if A[mid] = mid
6)          return mid
7)      else
8)          if A[mid] > mid
9)              end ← mid-1
10)         else
11)             start ← mid+1
12) return -1
```

9 (2.3)

האלגוריתם מבצע חיפוש על המערך A. ידוע כי A ממוין, לכן לכל אינדקס i, כל האיברים שמימינו גדולים מ-$A[i]$ וכל האיברים שמשמאלו קטנים מ-$A[i]$ (נזכור כי ידוע שהאיברים במערך שונים זה מזה ושלמים). לכן כדי למצוא את i $A[i]=i$ (נבצע חיפוש בינארי כך שבמקום שנחפש ערך כלשהו במערך, נחפש ערך i המקיים $A[i]=i$ (כלומר רק שורה 5 שונה). בכל פעם נחלק את המערך ל-2 (בעזרת המצביעים end, start ו-mid) כך שאם מתקיים $A[mid]>mid$ אז בהכרח אם קיים i כך ש-$A[i]=i$ אז הוא נמצא בחלק השמאלי של המערך, כלומר בין start ל-mid-1 וההפך.

הסיבוכיות היא $\Theta(\lg n)$ בדומה לחיפוש בינארי, מכיוון שבכל איטרציה מחלקים את כמות האיברים במערך בחצי בעזרת האינדקסים start, mid ו-end עד אשר נמצא את האינדקס הרצוי (או במקרה הגרוע לא נמצא ונצטרך לעבור על כל המערך). לכן נקבל את החלוקה הבאה (עד האיטרציה ה-x): $n \to \frac{n}{2} \to \frac{n}{4} \to \dots \to \frac{n}{2^x}$ כלומר זמן ריצה של $\Theta(\lg n)$.

שאלה 4:

עבור כל אחת מהטענות הבאות נסביר בקצרה מדוע נכון/לא נכון:

סעיף א':

יהי T עץ חיפוש בינארי. טענה: בהינתן ערך מפתח הקיים בעץ, ניתן להגיע לצומת בעל ערך המפתח הנתון בזמן $O(lgn)$. 5 (4.1)

הטענה אינה נכונה, בהינתן T עץ חיפוש בינארי בצורת שרוך ימני, כלומר קיימים n איברים כך שהמינימלי הוא השורש והמקסימלי הוא העלה (הצומת הכי ימני בעץ). לפיכך כדי למצוא את הערך הגדול ביותר בעץ (העלה הימני), נצטרך בחיפוש בינארי (שסיבוכיותו $O(h)$ כאשר h גובה העץ) לרדת עד לאיבר האחרון. כלומר במצב זה הסיבוכיות תהיה $O(n)$, מכיוון שגובה העץ הוא n.

סעיף ב':

T עץ בינארי (n צמתים) ו-A מערך בעל n מספרים שונים. הטענה הנתונה נכונה, נוכיח: 5 (4.2)
תחילה נמיין את המערך A. כעת נתחיל למלא את העץ T באופן הבא:
עבור כל צומת x בעץ T נבדוק לפי המיקום של הצומת בעץ כמה צמתים בהכרח אמורים להיות קטנים מהצומת x בכך שנספור את כל הבנים השמאליים של x, את הסבא של x ואת בניו השמאליים של הסבא של x. נשמור את ערך זה ב-index ונציב ב-x את הערך שב-$A[x]$. נבצע זאת על כל צומת בעץ ולבסוף נקבל כי T עץ חיפוש בינארי מבלי ששינינו את מבנה העץ.

סעיף ג':

הטענה נכונה, נוכיח בקצרה:
המערך A הינו ממוין, לכן נבצע את הבנייה באופן הבא:
בכל צומת בעץ T נכניס את החציון שבאותו תת מערך, כדי לשמור על איזון מלא של העץ.
כלומר עבור המערך $A[1...n]$ נכניס לשורש העץ T את הערך שב-$A[\frac{n}{2}]$ ונשלח את שאר המערך מצד שמאל $A[1...\frac{n}{2}-1]$ ומצד ימין $A[\frac{n}{2}+1...n]$ ונשוב על החלק השמאלי נכניס לבן השמאלי של השורש את הערך שב- $A[\frac{\frac{n}{2}-1}{2}]$ ובחלק הימני את הערך שב- $A[\frac{n}{2}+\frac{\frac{n}{2}-1}{2}]$ ונבצע את תהליך זה עד שנעבור על כל הערכים במערך A.
-2
לבסוף נקבל עץ אדום שחור מאוזן בסיבוכיות $O(n)$.
(4.3)

סעיף ה':

ומה עם צביעת הצמתים? איך זה מסתדר?

נתון מערך A בגודל n. הטענה אינה נכונה. נניח בשלילה כי ניתן לבנות תור קדימויות (מינימום) מ-n איברי המערך A בזמן $O(n)$, כך שהתור תומך בפעולת Extract-Min() בזמן $O(n)$, 5 (4.5)
לכן יהיה ניתן לבצע מיון מערך A המכיל n מספרים בלי הגבלות כלשהן על הקלט

המשך שאלה 4 סעיף ב:

בסיבוכיות $O(n)$ באופן הבא: נבנה תור קדימויות (סיבוכיות $O(n)$) ונבצע n פעמים
Extract-min() (סה"כ $O(n)$) ובכל הוצאה נכניס את הערך שיצא למערך B לפי סדר היציאה, לבסוף המערך
B שהתקבל יהיה ממוין בסיבוכיות $O(n)$ ולכן זו סתירה למשפט 8.1 בספר.

סעיף ג:

הטענה לא נכונה. ידוע שפונקציית הגיבוב היא פשוטה ואחידה, לכן ל- $\frac{m}{2}$ האיברים שהוכנסו
היתה הסתברות זהה להיכנס לכל אחד מהתאים. לא בהכרח כל איבר נכנס לתא שונה, יכול
להיות כי מספר איברים שווה ל- $\frac{m}{2}$ נכנסו לאותו תא כך שהתבצעה התנגשות לכן יכולים להיות
קיימים יותר מ- $\frac{m}{2}$ תאים ריקים. לכן ההסתברות להתנגשות בזמן ההכנסה של האיבר ה-$(\frac{m}{2}+1)$
יהיו לא יותר מ- $\frac{m}{2}$ תאים פנויים, לכן לא בהכרח ניתן לדעת, שכן יכול להיות שההסתברות
היא $\frac{1}{2}$ או יותר קטנה מ-$\frac{1}{2}$

5
(4.4)

שאלה 5:

נציע מבנה נתונים S התומך בפעולות הנדרשות, כך ש-n מציין את מספר האיברים במבנה.

מבנה הנתונים S ישתמש בערימת מקסימום M, בערימת מינימום H ובמערך ממוין A, ומשתנה min.

ערימת המקסימום M- תשמור את $\lceil \frac{n}{2} \rceil$ האיברים המינימלים במבנה S.

ערימת המינימום H- תשמור את $\lfloor \frac{n}{2} \rfloor - 6$ האיברים המקסימלים במבנה S.

מערך ממוין A- ישמור את 6 האיברים האמצעים בסדר ממוין.

min - משתנה שישמור את האיבר המינימלי שב-S.

כעת נראה את אופן הפעולה של השגרות: סיבוכיות $O(n)$

⊖ BUILD(S) - בניית המבנה S מניח סדרה של n מפתחות באופן הבא: תחילה נמצא מינימום ונכניס אותו למשתנה min

נבצע על n האיברים את השגרה SELECT עם הפרמטר $\lceil \frac{n}{2} \rceil$ כדי למצוא את ערך המיקום $\lceil \frac{n}{2} \rceil$. נכניס את כל האיברים שקטנים/שווים לערימה M (סיבוכיות בניית ערימה $O(n)$).

(סיבוכיות $O(6\lfloor \frac{n}{2} \rfloor)$)

נמצא מתוך ה- $\lfloor \frac{n}{2} \rfloor$ איברים הנותרים את ששת האיברים המינימלים ונכניס אותם בצורה ממוינת למערך (מספר קבוע של איברים לכן סיבוכיות $O(n)$).

כעת נכניס את כל ה- $\lfloor \frac{n}{2} \rfloor - 6$ האיברים הנותרים לערימה H (סיבוכיות בניית ערימה $O(n)$).

סה"כ סיבוכיות של שגרה זו היא $O(n)$.

⊖ MIN(S)- נחזיר את הערך שבמשתנה min, אשר מחושב בעת בניית המבנה S.

נשים לב שאין הכנסה של איברים חדשים, וההוצאה היא של החציון, לכן אין צורך לשנות את הערך שב-min.

סה"כ סיבוכיות של שגרה זו היא $O(1)$.

⊖ DEL-MEDIAN(S) - מחיקת החציון המפתחות של S באופן הבא:

נבצע את השגרה Extract-max על הערימה M, כדי להוציא את המקסימלי מהערימה M. נשים לב כי בערימה M קיימים $\lceil \frac{n}{2} \rceil$ איברים, לכן המקסימלי שבערימה זו (ערימה המכילה את כל האיברים המינימלים) הוא החציון של S. (פעולה זו של הוצאת המקסימום היא $O(\log n)$).

25
(5)

כעת נצטרך לשמור על האיזון של המבנים שהגדרנו באופן הבא:

נכניס את הערך $A[1]$ (המינימלי במערך) לערימה M. סיבוכיות $O(\log n)$.

נוציא את הערך המינימלי שבערימה H (סיבוכיות $O(\log n)$) ונכניס את ערך זה למערך A במקום $A[1]$ ונמיין את המערך (מספר קבוע של איברים, לכן $O(1)$). סה"כ סיבוכיות $O(\log n)$.

⊖ OS-MED(S)- נחזיר את הערך שנמצא בשורש הערימה H (המכילה $\lfloor \frac{n}{2} \rfloor - 6$ איברים). סיבוכיות $O(1)$.