מבני נתונים ומבוא לאלגוריתמים
20407

ממן 11

מוגש ע"י אורנית כהן גינדי

חלק מהפתרונות שלי סרוקים וחלק מוקלדים.

עימך הסליחה. פשוט התאים ככה יותר מבחינת אופי התרגיל

ממן 11 שאלה 1 א

1.2-2

נתונים: מיון הכנסה: $8n^2$

מיון מיזוג: $64 n\lg n$

עבור אילו ערכים של n מתקיים $8n^2 < 64 n\lg n$?

נעביר אגפים ונצמצם:

$$\frac{n}{\lg n} < 8 \Rightarrow n < 8\lg n$$

n מייצג גודל קלט ולכן חייב להיות $0 < n \in \mathbb{N}$

נציב ערכי n:

$n=1$: $\frac{1}{\lg 1} < 8 \Rightarrow$ לא חוקי $\Rightarrow n > 1$

$n=2$: $\frac{2}{\lg 2} = 2 < 8 \Rightarrow$ מתקיים $\Rightarrow n \geq 2$

$n=64$: $\frac{64}{\lg 64} = \frac{64}{6} < 8 \Rightarrow$ לא מתקיים

$n=32$: $\frac{32}{\lg 32} = \frac{64}{5} < 8 \Rightarrow$ מתקיים

מכאן ש: $5 < \lg n < 6$

נכפיל ב-8: $8\cdot 5 < 8\lg n < 8\cdot 6$

כדי למצוא את ה-n הקרוב נציב $n = 8\lg n$

$$8\cdot 5 < n < 8\cdot 6$$
$$\Downarrow$$
$$40 < n < 48$$

מניסוי וטעיה בטווח הזה מצאתי שה-n המקסימלי שמקיים את המשוואה של מהירות הריצה הינו 43.

$n=43 \Rightarrow 43 \overset{?}{>} 2^{5.375} = 41$ ✓ מתקיים

$n=44 \Rightarrow 44 \overset{?}{>} 2^{5.5}$ ✗ לא מתקיים.

לסיכום ערכי n המקיימים את המשוואה: $\boxed{2 \le n \le 43}$

1.2-3

מהו n הקטן ביותר עבורו: $100n^2 < 2^n$

$$\lg[100n^2] < \lg[2^n]$$

$$\lg 100 + \lg n^2 < \lg 2^n$$

$$\lg 100 + 2\lg n < n\lg 2$$

$$2\lg 10 + 2\lg n < n\lg 2$$

$$2\cdot \lg 10n < n\underbrace{\lg 2}_{=1}$$

$$\Downarrow$$

$$n > 2\lg 10n$$

$$\frac{n}{\lg 10n} > 2$$

$n \in \mathbb{N}$ וחיובי לכן $n>0$

(בדוק עבור $n=1$

$$\frac{1}{\lg 10} \overset{?}{>} 2$$

לא מתקיים

$n=16$

$$\frac{16}{\underbrace{\lg 10}_{\sim 3.2}+\underbrace{\lg 16}_{4}} \overset{?}{>} 2$$

מתקיים

(בין 1 - $n=16$ כדי למצוא n מינימלי :

$n=15$

$$\frac{15}{\underbrace{\underbrace{\lg 10}_{\sim 3.32}+\underbrace{\lg 15}_{\sim 3.9}}_{>7}} \overset{?}{>} 2$$

מתקיים.

הערך במכנה יהיה גדול מ7 גם עבור n=14
אבל אז כבר לא יתקיים שערך השבר גדול מ2
לכן הn המינימלי עבורו התנאי מתקיים הוא :

$$\boxed{n=15}$$

ממן 11 שאלה 1 ב

$f, g > 0$

נתון: $f(n) = O\left(\sum_{k=1}^{n} g(k)\right) \implies f(n) = Sum(g(n))$

צ"ל: $n = Sum(\log n)$

$$\Downarrow$$

$$n = O\left(\sum_{k=1}^{n} \log k\right)$$

כלומר הוא n חסום מלמעלה ע"י סכום של $\log n$.

פונקציה חוזרת: $\sum_{k=1}^{n} \log k = \log 1 + \log 2 + \dots + \log\frac{n}{2} + \dots + \log n$

אנו רואים: האיברים היו $\log n$ אלא הפונקציה החוסמת מלמעלה היא $n\log n$ וגם מחצית מהאיברים היו 0 ומקדימים השנייה: $\log\frac{n}{2}$ אנו הסכום הכולל חסום: $\frac{n}{2}\cdot\log\frac{n}{2}$

מכאן ש:

$$\frac{n}{2}\log\frac{n}{2} < \sum_{k=1}^{n} \log k < n\log n$$

נפתח את הסימון:

$$\sum_{k=1}^{n} \log k = \log 1 + \log 2 + \dots + \log n =$$
$$= \log(1 \cdot 2 \cdot \ldots \cdot n) = \log n!$$

$$\frac{n}{2}\log\frac{n}{2} < \log n! < n\log n$$

נמצא ש $\sum_{k=1}^{n} \log k$ הסכום עליה (מלמעלה) ע"י סדר גודל של $n\log n$.

וכמו כן ש:

$$\implies n = O(n\log n) \implies \boxed{n = O\left(\sum_{k=1}^{n} \log k\right)}$$ מש"ל

n חסום עליה ע"י $n\log n$

ממן 11 שאלה 1 ג

ממן 11 שאלה 1ג

<u>זוג ראשון:</u>

$f(n) = \log n$

$g(n) = n^2$

נראה ש- $f(n) = O(g(n))$

צ"ל שקיימים $c, n_0 > 0$ כך ש- $n \geq n_0$: $\log n \leq c \cdot n^2$

נחלק ב- n^2: $\frac{\log n}{n^2} \leq c$

נאמר: $n_0 = 2$ ו- $c = 1$

התנאי מתקיים ולכן $f(n) = O(g(n))$

נראה ש- $g(f(n)) = \omega(f(g(n)))$

$g(f(n)) = g(\log n) = \log^2 n$

$f(g(n)) = f(n^2) = \log n^2 = 2\log n$

נמצא את היחס בעזרת בדיקת הגבול:

$$\lim_{n\to\infty} \frac{g(f)}{f(g)} = \lim_{n\to\infty} \frac{\log^2 n}{2\log n} = \infty \Rightarrow g(f) = \omega(f(g))$$

זוג שני של F ו G בעמוד הבא

ממן 11 שאלה 1 ג

המשך 1ג

זוג שני:

$f(n)=\log n$

$g(n)=\sqrt{n}$

נראה ש- $f(n)=O(g(n))$

$$\log n \leq c\cdot\sqrt{n} \quad /:\sqrt{n}$$

$$\frac{\log n}{\sqrt{n}} \leq c$$

נבחר $n_0=4$ ו- $c=1$

$\sqrt{n}$ גדל מ- $\log n$ כך שככל ש n יגדל, השבר יקטן

כך שעבור כל $n\geq 4$ יתקיים - $\frac{\log n}{\sqrt{n}} \leq 1$

נראה כעת ש- $g(f)=o(f(g(n)))$

$$g(f(n)) = g(\log n) = \sqrt{\log n}$$

$$f(g(n)) = f(\sqrt{n}) = \log\sqrt{n} = \log n^{1/2} = \frac{1}{2}\log n$$

נבדוק את הגבול:

$$\lim_{n\to\infty}\frac{g(f)}{f(g)} = \lim_{n\to\infty}\frac{\sqrt{\log n}}{\frac{1}{2}\log n} = 0 \Rightarrow g(f)=o(f(g))$$

ממן 11 שאלה 2א.

אם מסתכלים על הלולאה while , רואים שפעולת הFLIP מתבצעת בדילוגים של i, מאותו i ועד סוף המערך.
הדילוגים הללו מביאים לכך שהפעולה FLIP תקרה בכל איטרציה שבה האינדקס של המערך יהיה מכפלה של i, וייתכנו אינדקסים שפעולת FLIP תתקיים עבורם מספר פעמים. המספר הזה יהיה כמספר המחלקים שיש לאותו אינדקס.
בתחילת התהליך כל המערך מאותחל לערכי F. ומשם והלאה, יישארו במצב T אלו שיש להם מספר אי זוגי של מחלקים (עליהם פעולת FLIP תבוצע מספר אי זוגי של פעמים ותסתיים בT) והאינדקסים עם מספר מחלקים זוגי, יתהפכו מספר זוגי של פעמים וישארו עם הערך F :

$$A[i]=\begin{cases} T & i=2k \\ F & i=2k+1 \end{cases}$$ כאשר K >=0

אם נסתכל על מחלקיו של מספר כלשהו, הרי תמיד מתקיים שאם i = k1 * k2 אז גם i= k2 * k1.
ואם k1≠k2 אז כמות המחלקים היא זוגית. אבל אם קיים מצב שבו k1 = k2 אז כמות המחלקים היא אי זוגית . ומכך שקיים k1=k2 כך ש i= k1*k2, אז i הוא מספר ריבועי. מכאן שאם ורק אם i הוא ריבועי , מספר פעולות הFLIP תהיינה אי זוגיות ובאינדקס i ישאר הערך T.

שמורת לולאה: בסוף איטרציה הi של לולאת הfor השניה (שורה 3) מתקיים שבתת מערך A[1..i] מתקיים שלכל 1 ≤ j ≤ i , A[j] =T אם ורק אם j הוא ריבוע שלם.
ז"א i = k^2 כאשר k מספר שלם.

אתחול: עבור i=1 , מתקיים A[1] =T ו-1 הוא ריבוע שלם.

תחזוקה: נניח שהטענה נכונה עבור i, צ"ל נכונות עבור i+1
נניח שבסוף איטרציה i, בתת המערך A[1..i] מתקיים שאם ורק אם i ריבוע שלם אז A[i] =T.
באיטרציה הi +1 פעולת הFLIP תופעל רק על האיבר הi+1 וממנו והלאה ז"א עבור כל k שמקיים i ≤ k ≤ n ובדילוגים של i+1 , מכאן ש תת המערך A[1..i] נותר ללא שינוי. וA[i+1] יקבל את ערכו בהתאם להיותו ריבוע שלם כמוסבר קודם. ערך זה יקובע עבור האיטרציה הבאה, כפי שהערך בו הi התקבע עבור איטרציה זו.

סיום: בסוף האיטרציה הn יתקבל שבכל המערך A, אם ורק אם i הוא ריבוע שלם, A[i] =T.

ממן 11 שאלה 2ב

בלולאת הfor בשורה 1, עוברים על כל המערך מi=1 עד n ← זמן ריצה מסדר O(n)

בלולאת הfor משורה 3 והלאה עוברים על כל המערך מi=1 עד n כאשר בכל איטרציה מופעלת לולאה על איבר אחד פחות מהאיטרציה הקודמת:

i=1 → n iterations
i=2 → (n-1)/2 iterations
i=3 →(n-2)/3 iterations
....
i=n → (n-i)/i iterations

$\sum_{i=1}^{n}(n-i)/i = \sum_{i=1}^{n}\ (n/i\ -1) = \sum_{i=1}^{n}\frac{n}{i}\ - n = n\sum_{i=1}^{n}\frac{1}{i}\ - n =$

$= n\sum_{i=1}^{n}\frac{1}{i}\ - n = n\ (\frac{1}{1}+\frac{1}{2}+\frac{1}{3}+\ldots+1/n)$ - n = nln n –n =n(ln n -1)

לפעולה FLIP זמן ריצה מסדר קבוע O(1)
אז זמן הריצה הטוטאלי: T(n) = O(n) + O(n lnn) + O(1)

אם נסתכל על הסדר הגבוה ביותר, זמן הריצה הוא O(n * ln n)

ממן 11 שאלה 3א.

שאלה 2-4 מהספר

יהי A[1..n] מערך של n מספרים **שונים**. אם i<j וגם A[i] > A]j] אז הזוג (I,j) נקרא היפוך בA

א. סה"כ 5 היפוכים:

A[1] > A[5]
A[2] > A[5]
A[3] > A[4]
A[3] > A[5]
A[4] >A[5]

ב. עבור {1,2....n} , מערך אשר ממוין בסדר יורד יהיה בעל מספר היפוכים הרב ביותר. מספר ההיפוכים מחושב כסכום של סדרה חשבונית ללא האיבר האחרון שלא קיים עבורו שום היפוך כיון שאין עוד אינדקס אחריו שיכיל מספר שיכול להיות קטן ממנו:

$$(n-1) + (n-2) + ...+ 2 +1 = (n-1)* (n-1-1)/2 = \boxed{(n-1)(n/2)}$$

ג. זמן הריצה הגרוע ביותר במיון הכנסה הוא במערך בו מספר ההיפוכים הוא מקסימלי או כפי שכתבתי קודם , מערך ממוין בסדר יורד.
הלולאה while מתבצעת כל עוד A[i] > k (ו- k הוא A[j] עבור j>i)
מספר הריצות המקסימלי של while הוא כאשר k נמוך יותר מכל A[i]

ד. בעמוד הבא

ממן 11 שאלה 3א.

ד. 2-4 אלגוריתם לחישוב מספר היפוכים בתמורה כלשהי של n איברים, ב- $\Theta(n\lg n)$ (במקרה הגרוע)

השתמשתי ברמז לשנות את Merge sort:

```
Count-Merge (A, p, q, r)
  n1 ← q-p+1
  n2 ← r-q
  for i ← 1 to n1
      L[i] ← A[p+i-1]
  for j ← 1 to n2
      R[j] ← A[q+j]
  i ← 1
  j ← 1
  opp ← 0
  for k ← p to r do
      if L[i] ≤ R[j]
      then A[k] ← L[i]
           i ← i+1
      else A[k] ← R[j]
           opp ← opp + n - i + 1
           j ← j+1
  return opp
```

סכימת ההיפוכים עי ספירה של כל הפעמים בהם R היה קטן מ L פלוס כל הערכים שנותרו בL מנקודת ההשוואה i

```
Opposites-Count (A, p, r)
  if p < r
  then q ← (p+r)/2
       countL ← Opposites-Count (A, p, q)
       countR ← Opposites-Count (A, q+1, r)
       return countL + countR + Count-Merge (A, p, q, r)
  return 0
```

ממן 11 שאלה 3ב.

A^R מערך הפוך ל- A, בגודל n.

נתון: ב- A ישנן k שגיאות

צ"ל - כמה שגיאות ב- A^R

במערך בגודל n ישנם $\frac{n(n-1)}{2}$ זוגות.

כאשר יש זוג שהוא שגוי (מסודר בסדר הפוך), אז במערך ההפוך הוא יהפוך לסדר נכון ולהיפך. כך שאם ב- A יש k שגיאות אזי במערך A^R אותן k שגיאות יהיו k זוגות מסודרים ומספר השגיאות יהיה מספר הזוגות שהיו מסודרים ב- A (והפכו לשגיאות) שהם בעצם המשלים ל- k ממספר הזוגות הכולל:

$$\boxed{\frac{(n-1)n}{2} - k}$$

ממן 11 שאלה 4

א. נתון A[1…n], השגרה <u>ISV(A)</u> מקבלת את מערך A ומחזיר true אם קיים v המקיים A[1] < v < A[n] (וfalse אחרת) :

זמן ריצה Θ(1)

<u>ISV(A)</u>

```
len ← length[A]
If A[length] - A[1] +1  >  len
   then return true
Else return false
```

ב. בהנחה ש ISV(A) החזירה true , השגרה <u>FINDV(A)</u> מוצאת v שחסר בסדרה.

האלגוריתם מתבסס על הנחה זו ועל גודל ההפרשים בין קצוות המערך למרכזו. אם ההפרשים זהים אזי ההנחה הראשונה מאשרת שניתן להמשיך בכל זאת ולבחור בצד אקראי להמשיך בו את החיפוש כיוון שיהיה ניתן למצוא v בכל צד שנבחר. אם לא זהים, ברור שנמשיך את החיפוש בצד בו ההפרש גדול יותר.

ה v שיוחזר כמובן לא קיים במערך וערכו יהיה **<u>בין</u>** 2 ערכים צמודים שההפרש ביניהם גדול מ1.

זמן ריצה Θ(log n)

<u>FINDV(A)</u>

```
High ← A[n]
low ← A[1]
While low<high do
    mid ← (high + low) /2
    if A[mid+1] - A[mid] > 1
    then  return A[mid]+1
    if (high – low +1) is even
    then rightDelta ← A[high] - A[mid+1]
    else rightDelta ← A[high] - A[mid]
    leftDelta ← A[mid] - A[low]
    if rightDelta >= leftDelta
     // comment:  can find missing v in both sides.  arbitrarily choosing rightDelta
    then low = mid + 1
    else if rightDelta < leftDelta
       then high ← mid
```

ממן 11 שאלה 5א.

$$\begin{cases} T(1) = c > 0 \\ T(n) = 16T\left(\frac{n}{4}\right) + \underbrace{n^{\alpha}\lg^{\alpha+1} n}_{f(n)} \end{cases} \qquad 0<\alpha\in\mathbb{R}$$

$a=16 \qquad b=4$

לפי משפט האב:

התשובה תלוי ב- $f(n)$ ו- $f(n)$ תלויה ב- α.

$f(n)$ עונה על התנאי של משפט 2 המורחב:

אם $\underline{\alpha=2}$

$$\log_b a = \log_4 16 = 2$$

$$f(n) = \Theta(n^{\log_b a}\log^k n) = \Theta(n^2\log^k n)$$

אם:

$k=\alpha+1$ ו- $\alpha=2$ $\Leftarrow$ $k=3$ ומתקיים $k\geq 0$

אז פתרון הנוסחה הוא

$$\boxed{T(n)=\Theta(n^2\lg^4 n)}$$

אם $\underline{0<\alpha<2}$ אז מקיים $n^{2-\varepsilon}$ כמו מקרה 1:

$$f(n) \overset{?}{=} O(n^{2-\varepsilon})$$

האם קיים c, n_0 ו-ε כך ש:

$$n^{\alpha}\lg^{\alpha+1}n \leq c\cdot n^{2-\varepsilon}$$

$$\frac{n^{\alpha}\lg^{\alpha+1}n}{n^{2-\varepsilon}} \leq c$$

נסמן $2-\varepsilon=t$ $\Leftarrow$ $t<2$

t ו- α קטנים מ- 2 אך הם יכולים להיות שונים

כך ש- $t>\alpha$ יתקיים c

כך ש:

$$\frac{n^{\alpha}\lg^{\alpha+1}n}{n^{t}} \leq c$$

$$\Rightarrow \boxed{T(n)=\Theta(n^2)}$$

ממן 11 שאלה 5א המשך.

אם $\boxed{\alpha>2}$ יתקיים $n^{2+\varepsilon}$ כמו דוגמה 3 של משפט האב.

נבדוק האם קיים c, n_0 ו- ε כך ש-

$$f(n) \stackrel{?}{=} \Omega(n^{2+\varepsilon})$$

$$n^\alpha \lg^{\alpha+1} n \ge c\cdot n^{2+\varepsilon} \quad /: n^{2+\varepsilon}$$

$$\frac{n^\alpha \lg^{\alpha+1} n}{n^{2+\varepsilon}} \ge c$$

כדי שמצב זה יתקיים צריך ש- $\alpha > 2+\varepsilon$ והרי אמרנו ש- $\alpha>2$, וניתן לבחור ε כך ש- $\alpha>2+\varepsilon$ כך שהמונה גדל מהר מהמכנה כך שקיים קבוע c מספיק קטן כדי שאי השוויון יתקיים.

<u>תנאי הרגולריות</u>: נראה שקיים c כך ש: $af(\frac{n}{b}) \le c\cdot f(n)$

$$\Downarrow$$

$$16 f\left(\frac n4\right) \le c\cdot n^\alpha \lg^{\alpha+1} n$$

$$16\left(\frac n4\right)^\alpha \lg^{\alpha+1}\left(\frac n4\right) \le c\cdot n^\alpha\lg^{\alpha+1} n \quad /: n^\alpha \lg^{\alpha+1} n$$

$$\Longleftrightarrow \frac{\frac{16 n^\alpha}{4^\alpha}\lg^{\alpha+1}\left(\frac n4\right)}{n^\alpha \lg^{\alpha+1} n} \le c \qquad \frac{16}{4^\alpha}=\frac{4^2}{4^\alpha}=4^{2-\alpha}$$

$$\Rightarrow \underbrace{4^{(2-\alpha)}}_{\text{קבוע}}\cdot \frac{n^\alpha\lg^{\alpha+1}\left(\frac n4\right)}{n^\alpha \lg^{\alpha+1} n}\le c \qquad \lg n > \lg\frac n4$$

$$\Downarrow$$

מכאן המכנה גדל יותר מהר מהמונה כך שהכל שואף ל-0 וקיים קבוע c שיהיה גדול יותר. תנאי הרגולריות מתקיים ולכן:

$$\boxed{T(n)=\Theta(n^\alpha \lg^{\alpha+1} n) \quad \text{עבור } \alpha>2}$$

ממן 11 שאלה 5ב

ממן 11 שאלה 5ב

נתונות הפונקציות f_1 ו- f_2:

$$f_1(n) = \max\{\sqrt{n^3}\lg n, \sqrt[3]{n^4}\lg^5 n\}$$

$$f_2(n) = \begin{cases} n\cdot\lg^3 n & n=2k \\ n^3\lg n & n=2k+1 \end{cases}$$

נסתכל על הביטויים בפונקציה f_1:

$$\sqrt{n^3}\lg n = n^{\frac{3}{2}}\lg n$$

$$\sqrt[3]{n^4}\lg^5 n = n^{\frac{4}{3}}\lg^5 n$$

$$n^{4/3} > n^{3/2}$$

ואילו

$$\lg^5 n > \lg n$$

אולם בהיררכית גודל פונקציות n בחזקת קבוע חיובי גדלה מהר יותר מפולילוג ולכן קיים n שממנו והלאה $f(n)$ מתייצבת על הביטוי השמאלי.

נסתכל על f_2: שוב שני ביטויים אולם לנצח נצחים תהיה החלפה ביניהם כיון ש- n זוגי ואי זוגי באים לסירוגין עד אינסוף. אז לא ניתן להגיד שקיימת התנהגות יציבה החל מ- n כלשהו.

אם נשווה את סדרי הגודל מול הביטוי השמאלי של $f_1(n)$ נמצא ש- $n^{3/2}$ נמצא בין n ל- n^3

כך ש- f_2 חוסמת את f_1 ונחסמת ע"י f_1 עבור ערכי n זוגיים ואי זוגיים בהתאמה

ומכאן שלא מתקיים יחס חסימה כלשהו בין f_1 ל- f_2.

ממן 11 שאלה 5ב

דוגמה נגדית אחרת:

$$f_1(n) = \max\{\underbrace{\sqrt{n^3}\lg n}_{g_1}, \underbrace{\sqrt[3]{n^4}\lg^5 n}_{g_2}\}$$

$$g_1 = \sqrt{n^3}\lg n = n^{\frac{3}{2}}\lg n \qquad g_2 = \sqrt[3]{n^4}\lg^5 n = n^{\frac{4}{3}}\lg^5 n$$

$$\exists n_0 , \forall n > n_0 \quad g_1 > g_2 \quad \lim_{n\to\infty} \frac{g_1(n)}{g_2(n)} = \infty$$

נסמן:

$$f_2(n) = \begin{cases} n\lg^3 n & n=2k \leftarrow h_1 \\ n^3\lg n & n=2k+1 \leftarrow h_2 \end{cases}$$

$$\exists n_1 , \forall n > n_1 \quad \lim_{n\to\infty} \frac{g_1(n)}{h_1(n)} = \infty$$

ושם מתקיים ש:

$$\exists n_2 , \forall n > n_2 \quad \lim \frac{g_1(n)}{h_2(n)} = 0$$

עבור המקסימום בין n_0 ו-n_1, אפשר להגיד ש-g_1 (של n_0 ומעלה חסום מלמטה ע"י f_1) וחסומה מלמעלה ע"י h_1

ועבור ה-n המקסימלי: בין n_0 ו-n_2, g_1 חסומה מלמעלה ע"י h_2.

כלומר f_2 מורכבת מ-h_1 ו-h_2. אבל h_1 חסומה מלמטה עבור n כלשהו, ונמצא אינסוף ערכי n מעליו. שעבורם h_2 חסומה מלמעלה ולהיפך.

ומכאן (על דרך השלילה), שאין שום יחס אסימפטוטי בין f_1 ו-f_2.