מבני נתונים ואלגוריתמים – תרגיל 1

יש לכתוב **שם מתרגלת\ת** ו**תעודת זהות** על הפתרון.

יש להגיש את התרגיל **בתחילת** התרגול בשבוע שמתחיל ב-13.11. [ניתן להגיש בקבוצת תרגול שאינה שלכם; לא ניתן להגיש לתאים כי אין תאים השנה.]

שאלה 1

דרגו את הפונקציות הבאות לפי קצב הגידול שלהן. הוכיחו את קביעתכם.

1. $(n-\sqrt{n}\ln n)\sqrt{n}$
2. $e^{(\ln\ln n)^2}$
3. $123n^2+456n+789$
4. $(\ln n)^5$
5. $(\ln n)^{\ln n}$
6. $\sqrt[n]{11n^3+4n^4}$

פיתרון

$\sqrt[n]{11n^3+4n^4} \ll (\ln n)^5 \ll e^{(\ln\ln n)^2} \ll (n-\sqrt{n}\ln n)\sqrt{n} \ll 123n^2+456n+789 \ll (\ln n)^{\ln n}$

כאשר $f(n) \ll g(n)$ אומר $f(n)=o\big(g(n)\big)$.

הוכחה $\sqrt[n]{11n^3+4n^4} \ll (\ln n)^5$: מתקיים

$$\sqrt[n]{11n^3+4n^4} \le \sqrt[n]{11n^4+4n^4} = \sqrt[n]{15}\left(\sqrt[n]{n}\right)^4 \xrightarrow{n\to\infty} 1\cdot 1 = 0$$

אבל $(\ln n)^5 \to \infty$ ולכן $\sqrt[n]{11n^3+4n^4} \ll (\ln n)^5$.

הוכחה $(\ln n)^5 \ll e^{(\ln\ln n)^2}$ נשים לב ש-$(\ln n)^5 = e^{5\ln\ln n}$. לכן:

$$\lim_{n\to\infty}\frac{\ln^5 n}{\exp((ln\,ln\,n)^2)} = \lim_{n\to\infty}\frac{\exp(5\,ln\,ln\,n)}{\exp((ln\,ln\,n)^2)} = \lim_{n\to\infty}\exp\big(\ln\ln n\,(5-\ln\ln n)\big)$$

הביטוי בתוך ה-exp שואף ל-$-\infty$ ולכן הגבול הוא 0.

הוכחה $e^{(\ln\ln n)^2} \ll (n-\sqrt{n}\ln n)\sqrt{n}$: נשים לב ש-

$$(n-\sqrt{n}\ln n)\sqrt{n} = n^{1.5}-n\ln n = n^{1.5}-o(n^{1.5}) = \Theta(n^{1.5})$$

לכן מספיק להראות $\lim_{n\to\infty}\frac{\exp((ln\,ln\,n)^2)}{n^{1.5}}=0$. נשים לב כי $n^{1.5}=e^{1.5\ln n}$. לכן:

$$\lim_{n\to\infty}\frac{\exp((ln\,ln\,n)^2)}{n^{1.5}} = \lim_{n\to\infty}\frac{\exp((ln\,ln\,n)^2)}{\exp(1.5\,ln\,n)} = \lim_{n\to\infty}\exp((ln\,ln\,n)^2-1.5\,ln\,n)$$

היות ו-$(\ln\ln n)^2 = o(\ln n)$ (זה כלל שציינו בכיתה, וגם אפשר לבדוק ע"י לופיטל) אז הביטוי באקספוננט שואף ל-$-\infty$ והאקספוננט שואף ל-0.

הוכחת $123n^2 + 456n + 789 \ll (n - \sqrt{n}\ln n)\sqrt{n}$: הראינו מקודם ש-$(n - \sqrt{n}\ln n)\sqrt{n} = \Theta(n^{1.5})$. בנוסף, לפי הכללים בכיתה $123n^2 + 456n + 789 = \Theta(n^2)$. לכן, הטענה נובעת מכך ש-$n^2 \ll n^{1.5}$. זה נובע מהכללים שראינו בכיתה או מבדיקה ישירה בעזרת גבול המנות.

הוכחת $(\ln n)^{\ln n} \ll 123n^2 + 456n + 789$: לפי הפיתרון הקודם מספיק להראות $\lim_{n\to\infty} \frac{n^2}{(\ln n)^{\ln n}} = 0$. באמת:

$$\lim_{n\to\infty} \frac{n^2}{(\ln n)^{\ln n}} = \lim_{n\to\infty} \frac{\exp(2\ln n)}{\exp(\ln n \ln\ln n)} = \lim_{n\to\infty} \exp\big(\ln n\,(2 - \ln\ln n)\big)$$

הביטוי באקספוננט שואף ל-$-\infty$ ולכן הגבול הוא 0.

שאלה 2

תהיינה $f, g: \mathbb{N} \to \mathbb{R}$ ו-$0 < \alpha, \beta \in \mathbb{R}$. הוכיחו $\alpha f(n) + \beta g(n) = \Theta(f(n) + g(n))$.

הוכחה

בה"כ $\alpha < \beta$, אחרת נחליף בין f, g. מתקיים $\alpha\big(f(n) + g(n)\big) \le \alpha f(n) + \beta g(n) \le \beta\big(f(n) + g(n)\big)$. מכאן נובע $\alpha f(n) + \beta g(n) = \Theta(f(n) + g(n))$. מש"ל.

שאלה 3

תהי $f: \mathbb{R}_{\ge 0} \to \mathbb{R}_{\ge 0}$.

א. נניח ש-f **<u>עולה</u>**. הוכיחו $f(1) + \dots + f(n-1) \le \int_0^n f(x)dx \le f(1) + \dots + f(n)$. [רמז: התבוננו בפונקציות $f(\lfloor x \rfloor), f(\lceil x \rceil)$.]

ב. בהנחות של סעיף א, הוכיחו כי $f(1) + \dots + f(n) = \Theta\left(\int_0^n f(x)dx + f(n)\right)$.

ג. הוכיחו כי כאשר f מונוטונית **<u>יורדת</u>** מתקיים $f(1) + \dots + f(n) = \Theta\big(\int_0^n f(x)dx + f(1)\big)$.

ד. מצאו בעזרת הסעיפים הקודמים או בכל דרך אחרת פונקציה מפורשת (ללא סכום או אינטגרל) $f: \mathbb{N} \to \mathbb{R}$ כך ש-$1\ln 1 + 2\ln 2 + \dots + n\ln n = \Theta(f(n))$.

פיתרון

הוכחת א: היות ו-$f(x)$ מונוטונית עולה מתקיים $f(\lfloor x \rfloor) \le f(x) \le f(\lceil x \rceil)$ (כי $\lfloor x \rfloor \le x \le \lceil x \rceil$). לכן מתקיים $\int_0^n f(\lfloor x \rfloor)dx \le \int_0^n f(x)dx \le \int_0^n f(\lceil x \rceil)dx$. אבל:

$$\int_0^n f(\lfloor x \rfloor)dx = \sum_{i=0}^{n-1} \int_i^{i+1} f(\lfloor x \rfloor)dx = \sum_{i=0}^{n-1} f(i) = f(0) + f(1) + \dots + f(n-1)$$

$$\int_0^n f(\lceil x \rceil)dx = \sum_{i=0}^{n-1} \int_i^{i+1} f(\lceil x \rceil)dx = \sum_{i=0}^{n-1} f(i) = f(1) + f(2) + \dots + f(n)$$

א נובע באופן מיידי מכך ש-$f(0) \ge 0$. מש"ל.

הוכחת ב: ב-א הראינו ש-$f(1) + \dots + f(n-1) \le \int_0^n f(x)dx$ ולכן

$f(1)+\cdots+f(n-1)+f(n) \le \int_0^n f(x)dx + f(n)$

מצד שני הראינו גם ש-$\int_0^n f(x)dx \le f(1)+\cdots+f(n)$ ולכן:

$\int_0^n f(x)dx + f(n) \le f(1)+\cdots+f(n)+f(n) \le 2\big(f(1)+\cdots+f(n)\big)$

מכאן נובע $\int_0^n f(x)dx + f(n) = \Theta\big(f(1)+f(2)+\cdots+f(n)\big)$. היות ו-$\Theta$ הוא יחס סימטרי, אז גמרנו. מש"ל.

הוכחת ג: דומה מאוד לא' וב'. לא נפרט.

פיתרון ד: נגדיר את $g(n)$ להיות $n\ln n$ בקטע $[1,\infty)$ ו-0 בקטע $[0,1)$. בדקו ש-g לא יורדת. לכן, לפי סעיף ב, $1\ln 1 + 2\ln 2 + \cdots + n\ln n = g(1)+g(2)+\cdots+g(n) = \Theta\left(\int_0^n g(x)dx + gn\right)$. מתקיים:

$\int_0^n g(x)dx = \int_1^n x\ln x\, dx = \left[\frac{x^2\ln x}{2}\right]_1^n - \int_1^n \frac{x^2}{2}\cdot\frac{1}{x}dx = \frac{1}{2}n^2\ln n - \left[\frac{x^2}{4}\right]_1^n = \frac{1}{2}n^2\ln n - \frac{n^2}{4} + \frac{1}{4}$

ולכן

$1\ln 1 + 2\ln 2 + \cdots + n\ln n = \Theta\left(\frac{1}{2}n^2\ln n - \frac{n^2}{4} + \frac{1}{4} + n\ln n\right) = \Theta(n^2\ln n)$

לסיכום, נבחר $f(n) = n^2\ln n$.

שאלה 4

מצאו ביטוי מפורש (ללא סכום) לסיבוכיות של נוסחאות הנסיגה הבאות:

1. $T(n) = 5T\left(\frac{n}{4}\right) + 2n$
2. $T(n) = 3T\left(\frac{n}{4}\right) + n$
3. $T(n) = 2T\left(\frac{n}{4}\right) + T\left(\frac{n}{2}\right) + n$

פיתרון

הערה: מניחים $T(n) = 1$ עבור $n \le 1$.

פיתרון 1: נשתמש במשפט המאסטר. מתקיים $2n = O\left(n^{\log_4 5-\epsilon}\right)$ עבור ϵ מספיק קטן ולכן $T(n) = \Theta\left(n^{\log_4 5}\right)$.

פיתרון 2: נשתמש במשפט המאסטר. מתקיים $2n = \Omega\left(n^{\log_4 3+\epsilon}\right)$ עבור ϵ מספיק קטן ולכן $T(n) = \Theta(n)$.

פיתרון 3: האינטואיציה אומרת שכנראה $T(n) = \Theta(n\lg n)$. ננסה להוכיח באינדוקציה[1]

$0.1n\lg n \le T(n) \le 10n\lg n + 1$.

[1] אם אתם תוהים איך נבחרו הקבועים 10 ו-0.1, אז התשובה היא שפשוט ניסינו לקחת קבועים שהם כנראה מספיק גדולים וזה עבד.

זה נכון עבור $n \le 1$ כי $T(n) = 1$. עבור $n > 1$ נקבל:

$T(n) = 2T\left(\frac{n}{4}\right) + T\left(\frac{n}{2}\right) + n \le 10 \cdot 2\left(\frac{n}{4}\right)\lg\left(\frac{n}{4}\right) + 2 + 10 \cdot \left(\frac{n}{2}\right)\lg\left(\frac{n}{2}\right) + 1 + n = 5n(\lg n - \lg 4) + 5n(\lg n - \lg 2) + 1 + n = 10n\lg n - 10n - 5n + 1 + n = 10n\lg n - 14n + 1 \le 10n\lg n + 1$

$T(n) = 2T\left(\frac{n}{4}\right) + T\left(\frac{n}{2}\right) + n \ge 0.2\left(\frac{n}{4}\right)\lg\left(\frac{n}{4}\right) + 0.1\left(\frac{n}{2}\right)\lg\left(\frac{n}{2}\right) + n = 0.05n(\lg n - \lg 4) + 0.05n(\lg n - \lg 2) + n = 0.1n\lg n - 0.1n - 0.05n + n \ge 0.1n\lg n$

ולכן גמרנו. (באי-שוויונים האדומים משתמשים בהנחת האינדוקציה.)

שאלה 5

מצאו אלגוריתם שמקבל כקלט מספרים שלמים a, n ומחזיר את a^n. על האלגוריתם לעבוד ב-$O(\lg n)$ פעולות [הראו זאת]. מה סיבוכיות הזיכרון של האלגוריתם?

פיתרון

```
int power(a,n):
      res = 1      // the output
      a_power = a // holds a^(2^k) after k-th iteration of the loop
      while n > 0:
            if n % 2 == 1:
                  res = res * a
            end if
            a_power = a_power * a_power
            n = int(n/2) // n/2 is rounded down
      end while
      return res
```

הסבר: אפשר לבדוק בקלות שהאלגוריתם נכון כאשר $n = 0$. אחרת, אפשר לכתוב $n = b_r 2^r + b_{r-1}2^{r-1} + \cdots + b_0 2^0$ כאשר $b_i \in \{0,1\}$ ו-$b_r = 1$. קל לראות (באינדוקציה) שלאחר k איטרציות של לולאת ה-while ערכו של res יהיה $a^{n \,\%\, 2^k} = a^{b_{(k-1)}2^{k-1} + \cdots + b_0 2^k}$, ערכו של a_power יהיה a^{2^k} וערכו של n יהיה $\lfloor n/2^k \rfloor = b_r 2^{r-k} + b_{r-1}2^{r-1-k} + \cdots + b_k 2^0$. לכן, אחרי $r+1$ איטרציות ערכו של res יהיה a^n וערכו של n יהיה 0. לכן אנו נצא מהלולאה ונחזיר את res, שמכיל את התשובה הנכונה.

מחוץ ללולאת ה-while מתבצעות $\Theta(1)$ פעולות. לולאת ה-while מתבצעת $r + 1 = \lceil \lg n \rceil$ פעמים וכל איטרציה שלה מבצעת $\Theta(1)$ פעולות. לכן, סיבוכיות הזמן תהייה $\Theta(1) + \Theta(\lceil \lg n \rceil) = \Theta(\lg n)$.

סיבוכיות הזיכרון היא $\Theta(1)$ כי היינו צריכים מספר קבוע של משתני עזר.

שאלה 6

נתון האלגוריתם הבא המקבל כקלט מספר n:

```
void Algo(int n):
      A = int array of size n
      for i = 2 to n-1:
```

```
        A[i] = 0
    end for
    for i = 2 to n-1:
        j = 2 * i
        while j < n:
            A[j] = 1
            j = j + i
        end while
    end for
    for i = 2 to n-1:
        if A[i] == 0:
            print i
        end if
    end for
```

1. מה עושה האלגוריתם?
2. מה סיבוכיות הזיכרון שלו?
3. מה סיבוכיות הזמן שלו?
4. *מצאו אלגוריתם מהיר יותר אסימפטוטית שמבצע אותו דבר. הוכיחו כי הוא מהיר יותר אסימפטוטית.

פיתרון

תשובה ל-1: האלגוריתם מדפיס את המספרים הראשוניים מ-1 עד n (לא כולל n).

הסבר: לאחר $k-1$ איטרציות של של לולאת ה-for השנייה האיברים במערך A שערכם 0 הם בדיוק אלה שהאינדקס שלהן הוא מספר ראשוני קטן או שווה ל-k או שהאינקס שלהם לא מתחלק באף מספר קטן או שווה ל-k (או שהאינדקס הוא 0 או 1, אבל נתעלם מזה). [ניתן להראות זאת באינדוקציה.] לכן, המספרים שיודפסו בלולאת ה-for השלישית יהיה בדיוק הראשוניים בין 1 ל-n (לא כולל).

תשובה ל-2: מקצים מערך מספרים באורך n ועוד מספר קבוע של משתנים. לכן סיבוכיות הזיכרון היא $\Theta(n)+\Theta(1)=\Theta(n)$.

תשובה ל-3: נתעלם ממה שמחוץ ללולאות כי זה $\Theta(1)$ ולכן זניח.

לולאות ה-for הראשונה והשלישית מתבצעות $n-2$ פעמים ובכל אחת מתבצעות $\Theta(1)$ פעולות. לכן הזמן שלהן הוא $(n-2)\Theta(1)=\Theta(n)$. באופן דומה, הפקודות בלולאת ה-for האמצעית שאינן ב-while עולות גם $\Theta(n)$. נשאר להבין כמה זמן לוקחות הפקודות בלולאת ה-while.

ראשית נשים לב שלולאת ה-while מתבצעת לכל היותר $\frac{n}{i}$ פעמים ולכל הפחות $\frac{n}{i}-2$ פעמים (באשר i הוא המשתנה בלולאת ה-for האמצעית). לכן, הפקודות בלולאת ה-while מתבצעות בין $\sum_{i=2}^{n}\left(\frac{n}{i}-2\right)=\sum_{i=2}^{n}\left(\frac{n}{i}\right)-2(n-1)$ ל-$\sum_{i=2}^{n}\left(\frac{n}{i}\right)$ פעמים.

ננסה להעריך את $\sum_{i=2}^{n}\left(\frac{n}{i}\right)=n\cdot\sum_{i=2}^{n}\frac{1}{i}$. לשם כך נעזר בתרגיל 3 סעיף ג ממנו נובע כי

$$\sum_{i=1}^{n-1}\frac{1}{i+1}=\Theta\left(\int_0^{n-1}\frac{dx}{x+1}\right)=\Theta([\ln(x+1)]_1^{n-1})=\Theta(\ln n-\ln 2)=\Theta(\ln n)$$

לכן, $\Theta(n\ln n) = \sum_{i=2}^{n}\frac{1}{i} \cdot n = \sum_{i=2}^{n}\left(\frac{n}{i}\right)$. זה אומר שלולאת ה-while רצה בין $\Theta(n\ln n) - 2(n-1)$ ל-$\Theta(n\ln n)$ פעמים ולכן היא רצה $\Theta(n\ln n)$ פעמים. זמן הריצה של הפקודות בה הוא $\Theta(1)$ ולכן נקבל כי זמן הריצה הכולל של לולאת ה-while הוא $\Theta(n\lg n) = \Theta(1) \cdot \Theta(n\ln n)$.

סה"כ קיבלנו שזמן הריצה הוא $\Theta(n\lg n) = \Theta(n) + \Theta(n\lg n)$.

תשובה ל-4: אפשר לייעל את האלגוריתם באופן הבא: במקום לבצע את לולאת ה-while עבור כל i נבצע אותה רק אם A[i]==0 (שקול לכך ש-i ראשוני). האלגוריתם עדיין יעבוד והסיבוכיות שלו תהייה $\Theta\left(n\sum_{p<n}\frac{1}{p}\right)$ באשר הסכום נלקח על כל הראשוניים הקטנים מ-n. אפשר להוכיח ש-$\sum_{p<n}\frac{1}{p} = \Theta(\ln\ln n)$ ולכן נקבל סיבוכיות של $\Theta(n\ln\ln n)$.

אפשר להוכיח $\sum_{p<n}\frac{1}{p} = \Theta(\ln\ln n)$ בעזרת משפט המספרים הראשוניים. קיימות דרכים אלמנטריות נוספות להראות זאת.