ממ"ן 11 – פתרונות נבחרים

(לקוח מעבודות סטודנטים; לאו דווקא מלאים/מושלמים אך בהחלט טובים)

שאלה 1

סעיף א

1.2-2

$$8n^2 < 64nlog(n)$$
$$8n < 64\log(n)$$
$$n < 8\log(n)$$

כיוון ששתי הפונקציות עולות ממש, נחפש את הנקודה שבה עבור n מתקיים $n < 8\log(n)$
אבל עבור n+1 מתקיים $n+1 \geq 8\log(n+1)$:
נראה כי עבור $n = 43$ מתקיים $43 < 8\log(43) = 43.41$ אבל עבור $n = 44$ מתקיים

$44 \geq 8\log(44) = 43.675$ כלומר, עבור $1 \leq n \leq 43$ מיון הכנסה יהיה יותר מהיר ממיון מיזוג.

1.2-3

$$100n^2 < 2^n$$

כיוון ששתי הפונקציות עולות ממש, נחפש את הn הראשון שבו התנאי יתקיים והוא יהיה התשובה.

נראה כי עבור$n = 14$ מתקיים $16384 = 2^{14} \geq 100 \cdot 14^2 = 19600$ אבל עבור $n = 15$ מתקיים $32768 = 2^{15} < 100 \cdot 15^2 = 22500$, כלומר החל מ n=15 אלגוריתם שזמן ריצתו הוא $100n^2$ יהיה יותר מהיר מאלגוריתם שזמן ריצתו הוא 2^n.

סעיף ב

א) הטענה נכונה:
$f(n) = Sum\big(g(n)\big)$ אם $f(n) = O\big(\Sigma_{k=1}^{n} g(k)\big)$. ומכאן שכדי לבדוק האם $n =$ $Sum(\log n)$ נבדוק האם מתקיים $n = O(\Sigma_{k=1}^{n} \log n)$. נראה שהטענה מתקיימת:
נסתכל על המשוואה בצורה הבאה: $\Sigma_{k=1}^{n} 1 = O(\Sigma_{k=1}^{n} \log n)$
יהי $c = 1$ ויהי $n_0 = 3$, נראה כי עבור $n = 4 > 3 = n_0$ מתקיים
$\Sigma_{k=1}^{4} 1 = 4 < 1 \cdot \Sigma_{k=1}^{4} \log n = 4.585$. בנוסף, לכל $n > 4$ מתקיים $\log n > 1$ ולכן ההוספה שלו לסכום תשמור על האי שוויון לעומת n.

הראנו כי קיים c ממשי וקיים n_0 טבעי כך שלכל $n > n_0$ מתקיים $n \leq 1 \cdot \Sigma_{k=1}^{n} \log n$ ומכאן שמתקיים $n = O(\Sigma_{k=1}^{n} \log n)$ ולכן מתקיים $n = Sum(\log n)$.

סעיף ג

זוג א':

$f_1(n) = \sqrt{n}$

$g_1(n) = 4^n$

זוג ב':

$f_2(n) = \log n$

$g_2(n) = n^2$

נראה ש-$f_1 = O(g_1)$. יהיו $c = 1, n_0 = 0$. לכל $n > n_0$ מתקיים:

$$f_1(n) = \log n < \frac{1}{2} < \log 2 < \log n \ \Rightarrow f_1 \leq cg_1$$

נראה ש-$f_2 = O(g_2)$. יהיו $c = 1, n_0 = 0$. לכל $n > n_0$ מתקיים:

$$f_2(n) = \log n \leq n < n^2 \ \ \Rightarrow f_2 \leq cg_2$$

בכך הראינו שהיחס O מתקיים עבור שני הזוגות.

נראה שמתקיים $g_1{}^\circ f_1 = o(f_1{}^\circ g_1)$:

$$g_1{}^\circ f_1 = 4^{\sqrt{n}}, \qquad f_1{}^\circ g_1 = \sqrt{4^n}$$

$$\lim \frac{g_1{}^\circ f_1}{f_1{}^\circ g_1} = \lim \frac{4^{\sqrt{n}}}{\sqrt{4^n}} = \lim \frac{4^{\mathrm{n}^{0.5}}}{4^{0.5n}} = \lim \frac{1}{4^{0.5n-\sqrt{n}}} = 0$$

$$\Rightarrow g_1{}^\circ f_1 = o(f_1{}^\circ g_1)$$

נראה שמתקיים $g_2{}^\circ f_2 = \omega(f_2{}^\circ g_2)$:

$$g_2{}^\circ f_2 = \log^2 n\,, \qquad f_2{}^\circ g_2 = \log n^2$$

$$\lim \frac{g_2{}^\circ f_2}{f_2{}^\circ g_2} = \lim \frac{\log^2 n}{\log n^2} = \lim \frac{\log^2 n}{2\log n} = \lim \frac{\log n}{2} = \infty$$

$$\Rightarrow g_2{}^\circ f_2 = \omega(f_2{}^\circ g_2)$$

סעיף ד

ראשית נציין כי היחסים יבדקו מהצורה $\mathrm{f}_1 = O(f_2)$ ולא בשני הכיוונים, הרי שאם $\mathrm{f}_1 = O(f_2)$ אזי $\mathrm{f}_2 = \Omega(f_1)$ וכיו"ב.

נבחין כי $\sqrt{n^3} \cdot \log n = \omega\left(\sqrt[3]{n^4} \cdot \log^5 n\right)$ ולכן:

$$\max\left(\sqrt{n^3} \cdot \log n , \sqrt[3]{n^4} \cdot \log^5 n\right) = \Theta(\sqrt{n^3} \cdot \log n)$$

$f_1 = O(f_2)$:

נניח בשלילה שמתקיים $f_1 = O(f_2)$, כלומר קיימים $c > 0, n_0 \in \mathbb{N}$ כך שלכל $n > n_0$ מתקיים

$$f_1(n) \le c \cdot f_2(n)$$

עבור חים זוגיים, צריך להתקיים:

$$\sqrt{n^3} \cdot \log n \le \mathrm{c} \cdot \mathrm{n} \cdot \log^3 n$$

$$c \ge \frac{\sqrt{n}}{\log^2 n}$$

אבל מתקיים $\lim_{n\to\infty} \frac{\sqrt{n}}{\log^2 n} = \infty$ ומכאן נקבל סתירה. שכן, לא יהיה c שיספק את כל החים החל ממקום מסויים, כי תמיד יגיע הn עבורו נצטרך c גדול יותר.

$f_1 = \Omega(f_2)$:

נניח בשלילה שמתקיים $f_1 = \Omega(f_2)$, כלומר קיימים $c > 0, n_0 \in \mathbb{N}$ כך שלכל $n > n_0$ מתקיים $f_1(n) \ge c \cdot f_2(n)$

עבור חים אי-זוגיים, צריך להתקיים:

$$\sqrt{n^3} \cdot \log n \ge \mathrm{c} \cdot \mathrm{n}^3 \log n$$

$$c \le \frac{1}{\sqrt{n^3}}$$

אבל מתקיים $\lim_{n\to\infty} \frac{1}{\sqrt{n^3}} = 0$, ומכאן נקבל סתירה. שכן, לא יהיה c שיספק את החים כיוון שלא ייתכן $c = 0$ כאשר הn יהיה מספיק גדול (לפי הגבול כאשר $n \to \infty$).

$f_1 = \Theta(f_2)$:

כיוון שלא מתקיים $f_1 = O(f_2)$ וגם $f_1 = \Omega(f_2)$ אזי שלא מתקיים $f_1 = \Theta(f_2)$.

$f_1 = o(f_2)$:

כיוון שלא מתקיים $f_1 = O(f_2)$ אזי בהכרח לא מתקיים $f_1 = o(f_2)$ (אם לא קיים c אחד עבורו קיים n_0 שמקיים את התנאי, אזי שבהכרח הוא לא יתקיים לכל c).

$f_1 = \omega(f_2)$:

כיוון שלא מתקיים $f_1 = \Omega(f_2)$ אזי בהכרח לא מתקיים $f_1 = \omega(f_2)$ (אם לא קיים c אחד עבורו קיים n_0 שמקיים את התנאי, אזי שבהכרח הוא לא יתקיים לכל c).

שאלה 2

סעיף א

ראשית, נבחין כי כל מספר שהוא ריבוע שלם הוא בעל מספר <u>אי זוגי</u> של מחלקים, ואילו לכל מספר שאינו ריבוע שלם ישנו מספר <u>זוגי</u> של מחלקים:

יהי n מספר שלם, ותהי D קבוצת המחלקים של n הקטנים ממש מ-$\sqrt{n}$: $D \coloneqq \{p : p | n, p < \sqrt{n}\}$.

לכל $p \in D$, קיים מספר שלם יחיד $q > p, q \notin D$ כך ש-$p * q = n$. לפיכך, הקבוצה $D' \coloneqq D \cup \{ q : \exists p \in D \; such \; that \;\; q * p = n\}$ מכילה מספר זוגי של איברים.

אם n הוא ריבוע שלם, אז $\sqrt{n}$ הוא מספר שלם המחלק את n, שאינו נכלל בקבוצה D'. אחרת, לא קיימים ל-n מחלקים שאינם ב-D'. לפיכך, מספר המחלקים של n הוא אי-זוגי אם ורק אם n הוא ריבוע שלם.

כעת נוכיח את הטענה המבוקשת:

<u>שמורת הלולאה:</u> אחרי האיטרציה ה-i (שורה 3 בשגרה TRUE_SQAURE), עבור תת-המערך $A[1 \ldots i]$ מתקיים לכל $1 \leq j \leq i$ שערך התא $A[j]$ הוא True אם"ם j ריבוע שלם.

<u>אתחול:</u> הטענה נכונה עבור $i = 1$: המספר 1 הוא ריבוע שלם, והתבצע בדיוק היפוך אחד.

<u>תחזוקה:</u> נראה שאם הטענה נכונה אחרי איטרציה i כלשהי, אז היא תהיה נכונה גם אחרי האיטרציה הבאה, $i + 1$.

מהנחתנו נובע שלכל $j < i + 1$, ערך התא $A[j]$ תואם את היותו של j ריבוע שלם. נשים לב שבכל איטרציה, k מאותחל להיות כמספר האיטרציה הנוכחית (שורה 4), והקריאה ל-FLIP נעשית בלולאה הפנימית (שורה 5) רק על ערכים גדולים או שווים לאותו k – כלומר ערכם של התאים ב-A מאינדקס קטן מ- $i + 1$ לא ישתנה. לפיכך מספיק להראות שערך התא $A[i + 1]$ הוא True אם"ם $i + 1$ הוא ריבוע שלם.

הלולאה הפנימית (שורה 5) קוראת לשגרה FLIP על כל מספר ש-k הוא אחד המחלקים שלו. לכן, בסוף האיטרציה ה-$i + 1$ התא $A[i + 1]$ עבר היפוכים כמספר המחלקים שלו. מכאן נובע שאם $i + 1$ הוא ריבוע שלם, אז השגרה FLIP נקראה על $A[i + 1]$ מספר אי-זוגי של פעמים וערכו בסוף האיטרציה הוא True (כיוון שכל הערכים ב-A מאותחלים להיות False בתחילת האלגוריתם). אחרת, אם $i + 1$ אינו ריבוע שלם, הוא עבר מספר זוגי של היפוכים וערכו נשאר כשהיה בעת האתחול – False.

<u>סיום:</u> כאשר $i = n$ כל התאים עברו היפוכים כמספר המחלקים שלהם ולפיכך ערכו של כל תא הוא True אם"ם האינדקס שלו הוא ריבוע שלם, כנדרש.

סעיף ב

בלולאה הראשונה (שורות 1-2) עוברים פעם אחת על כל איברי המערך – סה"כ n פעולות.

השגרה FLIP מבצעת 2 פעולות – פעולת השוואה אחת ופעולת השמה אחת. לפיכך זמן הריצה שלה הוא $\Theta(1)$.

הלולאה החיצונית (שורה 3) רצה מ-1 עד n, והלולאה הפנימית (שורות 5-7) מתבצעת $\frac{n}{k}$ פעמים בכל פעם. לכן בסך הכל זמן הריצה של שתי הלולאות הוא

$$\sum_{i=1}^{n} \frac{n}{i} = n * \sum_{i=1}^{n} \frac{1}{i} = n\left(\ln n + O(1)\right)$$

(השוויון $\sum_{i=1}^{n} \frac{1}{i} = \ln n + O(1)$ נובע מעמוד 271 בספר).

לפיכך זמן הריצה של השגרה כולה הוא:

$$T(n) = n + n\big(\ln n + O(1)\big) = \Theta(n \log n)$$

שאלה 3

סעיף א

a. חמשת ההיפוכים הם: $(2,1), (3,1), (8,6), (8,1), (6,1)$

b. מספר ההיפוכים הרב ביותר יהיה במערך ממויין בסדר הפוך, כלומר במערך ממויין בסדר יורד, כיוון שבמקרה זה כל איבר יהיה היפוך עם כל האיברים שלפניו, ומספר ההיפוכים במקרה כזה יהיה $\Sigma_{i=1}^{n}(n-i) = \frac{n(n-1)}{2}$.

c. מיון הכנסה מבצע החלפה עבור כל היפוך שקיים לאיבר מסוים החל מהאינדקס שלו כלפי מטה בסדר יורד (כל היפוך שקיים איתו בתור האינדקס הגדול יותר), ואז ברגע שמריצים את האלגוריתם החל מהאיבר הראשון עד האיבר האחרון, נקבל בכל שלב מערך ממויין עד האינדקס שעליו רצנו עכשיו. לכן, ככל שיהיו יותר היפוכים במערך, זמן הריצה של אלגוריתם מיון-הכנסה יהיה יותר ארוך.

d. נשתמש ברעיון של מיון מיזוג, שבו ממזגים מערכים ממויינים, ובכל שלב באלגוריתם כאשר רוצים למזג שתי רשימות ממויינות בודקים האם איבר מרשימה אחת גדול מאיבר מהרשימה השניה, ובמילים אחרות, האם נדרש היפוך בין שני האיברים הללו.

נסתמך על האלגוריתם בספר הלימוד בעמוד 25-27 ונוסיף את השורות הבאות:

1. אחרי שורה 2 בפונקציה MERGE נוסיף $counter < -0$, משתנה שיספור את מספר ההיפוכים עבור שני המערכים הממוזגים
2. אחרי שורה 17 (עדיין בתוך הelse) נוסיף:
$coutner < -counter + length(L) - i$ (במצב זה נמצא עוד היפוך שצריך להתקיים ולכן נוסיף אותו ואת שאר האיברים ברשימה שגדולים ממנו ובהכרח מצריכים היפוך).
3. בסוף הפונקציה MERGE נחזיר את counter
4. נשכתב את הפונקציה MERGE-SORT בצורה הבאה:

MERGE-SORT(A,p,r):

1. If p < r
2. Then q← $\left\lfloor \frac{p+r}{2} \right\rfloor$
3. c_1←MERGE-SORT(A,p,q)
4. c_2←MERGE-SORT(A,q+1,r)
5. c_3←MERGE(A,p,q,r)
6. Return $c_1 + c_2 + c_3$
7. Return 0

כל היפוך נספר פעם אחת כי עבור כל זוג איברים, האלגוריתם יעבור עליהם פעם אחת בלבד (לאחר מכן הם יהיו באותו מערך בשלב הבא באלגוריתם)

סעיף ב

א) נסמן ב$f(A)$ את מספר ההיפוכים בA.

מספר הזוגות במערך - $\frac{(n-1)n}{2}$ כיוון ש$\Sigma_{i=1}^{n}(n-i) = \frac{n(n-1)}{2}$.

מספר השגיאות (היפוכים) במערך A^R הוא כמספר הזוגות בA פחות מספר ההיפוכים בA (כל זוג שאינו היפוך בA הוא היפוך בA^R ולהיפך), כלומר:

$$f(A^R) = \frac{(n-1)n}{2} - f(A) = \frac{n(n-1)}{2} - K$$

שאלה 4

סעיף א

<u>$IS\_MISSING\_V(A[1 \dots n])$:</u>

1. $Return\ A[n] - A[1] \neq n-1$

קל לראות שזמן הריצה שלהשגרה הוא $\Theta(1)$, שכן נעשה מספר קבוע של פעולות (2 פעולות חיסור ופעולת השוואה אחת).

נוכיח את נכונות השגרה:

אם כל v המקיים $A[1] < v < A[n]$ <u>מופיע</u> במערך, אזי כל המספרים מהערך של האיבר הראשון ב-A ועד לאחרון מופיעים במערך בזה אחר זה, בסדר עולה (לדוגמא - [3,4,5,6]). במקרה זה הביטוי $A[n] - A[1]$ יהיה שווה בדיוק $n-1$.

מכך שהמערך ממוין, נסיק כי בין $A[n]$ לבין $A[1]$ קיימים לכל היותר $A[n] - A[1] - 1$ ערכים. אם קיים v כנ"ל <u>שלא מופיע</u> ב-A, אז בין $A[1]$ ל-$A[n]$ קיימים <u>פחות</u> מספרים מאשר ההפרש ביניהם. כלומר, $A[n] - A[1] < n-1$.

סעיף ב

נבנה את האלגוריתם $FIND\_MISSING\_V$ בשיטה רקורסיבית, בדומה לחיפוש בינארי: בכל איטרציה, אם גודל המערך הנתון גדול מ-2, נבדוק האם חסר איבר v כמבוקשנו בחצי השמאלי של המערך הנתון (נעשה זאת באמצעות שימוש ב-$IS\_MISSING\_V$ מהסעיף הקודם). אם כן – נקרא רקורסיבית לשגרה $FIND\_NISSING\_V$, עם תת-המערך השמאלי של המערך הנתון. אחרת – נקרא רקורסיבית לשגרה, הפעם עם תת-המערך הימני.

אם גודל המערך הנתון הוא בדיוק 2, אז האיבר החסר נמצא בין שני האיברים הקיימים במערך – לכן נחזיר את $A[1]+1$.

<u>$FIND\_MISSING\_V(A[1 \dots n])$:</u>

1. $\boldsymbol{if}\ n = 2, \boldsymbol{return}\ A[1] + 1$
2. $i \leftarrow \left\lceil \frac{n}{2} \right\rceil$
3. $\boldsymbol{if}\ \underline{IS\_MISSING\_V(A)}$
   3.1. $\boldsymbol{return}\ \underline{FIND\_MISSING\_V(A[1 \dots i])}$
4. $\boldsymbol{else}$
   4.1. $\boldsymbol{return}\ \underline{FIND\_MISSING\_V(A[i \dots n])}$

<u>תקציר</u> נכונות: נסתמך על נכונות שגרת העזר מהסעיף הקודם. כעת, נניח באינדוקציה שהשגרה נכונה עבור ערכים קטנים מ-n. ברור שאם חסר מספר אז הוא חסר או בצד שמאל או בצד ימין (או בשניהם). לפיכך נבדוק ע"י שגרת העזר אם הוא חסר בשמאל. אם כן, ע"פ הנחת האינדוקציה הוא יימצא. אחרת, הוא יימצא בצד ימין.

נראה שסיבוכיות השגרה היא $\Theta(\log n)$:

עבור $n = 2$, $T(n) = \Theta(1)$, כיוון שנעשה מספר קבוע של פעולות (שורה 1).

לכל $n > 2$, נקראת פעם אחת השגרה $IS\_MISSING\_V$, שזמן הריצה שלה הוא $\Theta(1)$, ופעם אחת השגרה $FIND\_MISSING\_V$ עם מערך שגודלו $\frac{n}{2}$. לפיכך, נגדיר את $T(n)$ באמצעות נוסחת הנסיגה הבאה:

$$T(n) = 1 + T\left(\frac{n}{2}\right)$$

משיטת עץ הרקורסיה נקבל מיד ש-$T(n) = \Theta(\log n)$. **מ.ש.ל.**

שאלה 5

1. נשתמש במשפט האב:

$$a = 8, b = 2 \Rightarrow \log_b a = \log_2 8 = 3$$
$$f(n) = n + n^3 = \Theta(n^3)$$

כלומר, $f_n = \Theta(n^3) = \Theta\left(n^{\log_2 8}\right) = \Theta\left(n^{\log_b a}\right)$.

ומכאן לפי מקרה 2 של משפט האב, נקבל כי $T(n) = \Theta\left(n^{\log_b a} \log n\right) = \Theta(n^3 \log n)$

2. $a = k, b = 2, k \geq 2$, נחלק לכמה מקרים:

<u>$k = 2$:</u>

$f(n) = 0$, ויהי $\varepsilon = \frac{1}{2}$, מתקיים $f(n) = 0 = O(n^{\log_2 2 - \frac{1}{2}})$

ומכאן נקבל לפי מקרה 1 של משפט האב כי $T(n) = \Theta\left(n^{\log_b a}\right) = \Theta(n)$

<u>$2 < k < 8$:</u>

במקרה זה $1 < \log_2 k < 3$, נבחר $\varepsilon = 3 - \log_2 k$ ונקבל ש:

$$f(n) = \Theta(n^3) = \Theta\left(n^{\log_2 k + (3 - \log_2 k)}\right) = \Theta\left(n^{\log_2 k + \varepsilon}\right) = \Omega(n^{\log_2 k + \varepsilon})$$

בנוסף, עבור $c = \frac{k}{8}$ מתקיים $\frac{k}{8} n^3 = k\left(\frac{n}{2}\right)^3 = kf\left(\frac{n}{2}\right) \leq \frac{k}{8} f(n) = \frac{k}{8} n^3$.

ולכן מתקיים מקרה 3 של משפט האב, ולכן: $T(n) = \Theta\big(f(n)\big) = \Theta(n^3)$

<u>$k = 8$:</u>

$f(n) = \Theta(n^3) = \Theta(n^{\log_2 8})$ ולכן לפי מקרה 2 של משפט האב, נקבל כי:

$$\mathrm{T(n)} = \Theta\left(\mathrm{n}^{\log_2 8} \log n\right) = \Theta(n^3 \log n)$$

<u>$k > 8$:</u>

במקרה זה, $\log_2 k > 3$. נבחר $\varepsilon = \log_2 k - 3$ ונראה שמתקיים:

$$f(n) = \Theta(n^3) = \Theta\left(n^{\log_2 k - (\log_2 k - 3)}\right) = \Theta\left(n^{\log_2 k - \varepsilon}\right) = O\left(n^{\log_2 k - \varepsilon}\right)$$

ולכן לפי מקרה 1 של משפט האב, נקבל כי $T(n) = \Theta\left(n^{\log_2 k}\right)$.

3. נשתמש במקרה 2 המורחב של משפט האב:

$$a = 2, b = 4 \Rightarrow \log_b a = \log_4 2 = 0.5$$

ומכאן נקבל:

$$f(n) = \sqrt{n} \log n = n^{0.5} \log n = \Theta(n^{0.5} \log n) = \Theta(n^{\log_4 2} \log^1 n)$$

ומכאן לפי מקרה 2 המורחב של משפט האב, נקבל $T(n) = \Theta\left(n^{\log_4 2} \log^2 n\right) =$ $\Theta\left(\sqrt{n} \log^2 n\right)$

4. לפי שיטת האיטרציות:

$$T(n) = T(n-1) + n \cdot \log n + n$$
$$T(n-1) = T(n-2) + (n-1) \cdot \log(n-1) + n - 1$$
$$\dots$$
$$T(2) = T(1) + 2\log 2 + 2$$
$$T(1) = T(0) + 1\log 1 + 1$$
$$T(n) = T(0) + \Sigma_{i=1}^{n} i \cdot \log i + \Sigma_{i=1}^{n} i = \Theta(n^2 \log n) + \Theta(n^2) = \Theta(n^2 \log n)$$

5. $T(n) = n^2\sqrt{n} \cdot T\left(\sqrt{n}\right) + n^5 \cdot \log^3 n + \log^5 n$

$$\frac{T(n)}{n^5} = \frac{T\left(\sqrt{n}\right)}{n^2\sqrt{n}} + \log^3 n + \frac{\log^5 n}{n^5}$$

נסמן $U(n) = \frac{T(n)}{n^5}$ ונקבל:

$$U(n) = U\left(\sqrt{n}\right) + \Theta(\log^3 n)$$

כעת נסמן:

$n = 2^m, m = \log n, S(m) = U(2^m)$ ונקבל:

$$S(m) = S\left(\frac{m}{2}\right) + m^3$$

כעת, לפי מקרה 3 של משפט האב נקבל $S(m) = \Theta(n^3)$.

מכאן ש$\frac{T(n)}{n^5} = \Theta(\log^3 n)$ ולבסוף נקבל כי $T(n) = \Theta(n^5 \log^3 n)$.