תרגיל בית מספר 3 ~ מבוא מורחב למדעי המחשב

שחר פרץ

11 ביולי 2024

1

א) 1. $64^{\log_4 n} = O(n^4)$. **נוכיח.** צריך להוכיח קיום a, n_0 המקיים את אי־השוויון הבא. יהי $n \geq n_0$:

$$64^{\log_4 n} \leq a \cdot n^4 \iff 4^{3\log_4 n} \leq an^4 \iff n^3 \leq an^4 \iff n^3 = O(n^4)$$

כאשר הטענה האחרונה ידועה מן החסמים שהוצגו בהרצאה.

2. $\boldsymbol{\log(n) = O(\log(\log(n^4)))}$. **נסתור.** נניח בשלילה ונראה סתירה.

$$\log n = O(\log(\log(n^4))) = O(\log(4\log(n))) = O(\underbrace{\log 4}_{\text{const.}} + \log n) = O(\log\log n)$$

אך מההירכיית החסמים שהוצגה בכיתה, $\log\log n$ הוא חסם אסימפטוטי תחתון ממש מ־$\log n$, ולכן לא ייתכן שיחסום אותו מלמעלה.

3. $\boldsymbol{\forall f_1(n) = O(g_1(n)), f_2(n) = O(g_2(n)). f_1 \circ f_2 = O((g_1 \circ g_2)(n))}$. **נסתור.** נבחר:

$$\begin{aligned} &f_1 = 2^n, \quad g_1 = 2^n, \qquad n = O(0.5n) \\ &f_2 = n \qquad g_2 = 0.5n \quad 2^n = O(2^n) \end{aligned}$$

נניח בשלילה שהטענה נכונה. מכאן:

$$(f_1 \circ f_2)(n) = O(g_1 \circ g_2)(n) \implies 2^n = O(2^{0.5n}) \implies 2^n = O(\sqrt{2}^n)$$

אך זו סתירה לטבלה שראינו להררכיית חסמים אסימפטוטיים.

4. $\boldsymbol{f = O(6^n) \land g = 6^{O(n)} \implies g = O(f)}$. **נסתור.** עבור $g = 6^{2n}$ (תקין כי $2n = O(n)$), וגם $6^n = O(6^n)$, יתקיים:

$$g(n) = 12^n = O(6^n) = O(f(n))$$

לפי הטענה, אך זו סתירה להררכיית החסמים שראינו בהרצאה.

ב) 1. יהיו $0 \leq a_1, a_2, \dots$ סדרת מספרים אי־שליליים. נניח קיום קבועים $0 < b, c \leq 1$ כך שלכל n לפחות $n \cdot b$ מאיברי הסדרה הם בגודל של לפחות $c \cdot \max\{a_1, \dots a_n\}$. נוכיח $\sum_{i=1}^n a_i = \Theta(n \cdot \max\{a_1, \dots a_n\})$.

הוכחה. תהה $a_1, a_2, \dots$ סדרה של מספרים. יהי $n \in \mathbb{N}$. נניח קיום קבוע $b, c \in \mathbb{N}$ כך ש־$b \cdot n$ מאיברי הסדרה הם בגודל של לפחות $c \cdot \max\{a_1, \dots a_n\}$. בפרט, הסדרה $a_1, \dots, a_n$ תקיים את התנאי הזה. ידוע:

$$\underbrace{cb}_{const.} \cdot n \max\{a_1, \dots, a_n\} \leq \sum_{i=1}^n \begin{cases} a_i & a_i = c\max\{a_1, \dots a_n\} \\ 0 & \text{else} \end{cases} \leq \sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} \boldsymbol{a_i} \leq \sum_{i=1}^n \max\{a_1, \dots a_n\} = n \cdot \max\{a_1, \dots a_n\}$$

■

סה"כ $\sum_{i=1}^n a_i = \Theta(n \cdot \max\{a_1, \dots, a_n\})$ מהגדרת החסם כדרוש.

2. **צ.ל.** $n \log n = O(\log(n!))$.

הוכחה. נתבונן בסדרה הבאה:

$$a_i = \log i, \ 1 \leq i \leq n, \ \sum_{i=0}^n a_i = \sum_{k=0}^n \log i = \log\left(\prod_{k=1}^n k\right) = \log n!$$

עבור קבועים $b, c = 0.5$ יתקיים הא"ש הבאה בנכונות ל־$0.5n$ מהאיברים (בעיגול):

$$\forall 0.5n \leq i \leq n. a_i \geq a_{0.5n} = \log(0.5n) = -1 + \log n \overset{!}{\geq} 0.5\log n = 0.5a_n = 0.5\max\{a_1, \dots, a_n\}$$

כאשר הא"ש לעיל מתקיים לכל $n \geq 4$ מתוך מציאת נק' החיתוך בין הפונקציות בא"ש והיותן מונוטוניות עולות, אך אין זה משנה כי ניקח $n \to \infty$ בשביל לחשב חסם אסימפטוטי. סה"כ הוכחנו את התנאים לטענה לעיל כך ש־$\log n! = \Theta(n \cdot \max\{a_1, \dots a_n\}) =$ $\Theta(n \log n)$. בפרט $n \log n = O(\log n!)$, כדרוש. ■

3. נגדיר $p_k(n) = \sum_{i=1}^{n} i^k$. צ.ל. לכל $k \geq 1$ מתקיים $p_k(n) = \Theta(n^{k+1})$.

הוכחה. נתבונן בסדרה $a_i = i^k,\ 1 \leq i \leq n$. נגדיר קבועים $c, b = 0.5$. עבור $n \cdot b = 0.5n$ מאיברי הסדרה (לפחות), הם $a_j \in \{a_{0.5n}, \dots a_n\}$ (בעיגול) יקיימו ממונוטוניות הסדרה: $a_j \geq a_{0.5n} = (0.5n)^k \geq 0.5 \cdot n^k = c \cdot \max\{a_1, \dots a_n\}$ (כי $0.5 \geq 0.5^k \forall k \geq 1$), כלומר הסדרה עונה על התנאים של הטענה שהוכחה בסעיף 1. אזי:

$$p_k(n) = \sum_{i=1}^{n} a_i = \Theta(n \cdot \max\{a_1, \dots a_m\}) = \Theta(n \cdot n^k) = \Theta(n^{k+1})$$

כדרוש. ■

4. צ.ל. $s := \sum_{i=1}^{n} 2^i i^k = \Theta(2^n \cdot n^k)$.

הוכחה. נוכיח את שני החסמים.

- **חסם עליון:** נוכיח $\sum_{i=1}^{n} 2^i i^k = O(2^n n^k)$. ידוע:

$$\forall n \geq \underbrace{1}_{n_0}.s = \sum_{i=1}^{n} 2^i i^k = \underbrace{\sum_{i=1}^{n-1} 2^i (i)^k}_{\geq 0} + 2^n n^k \leq \underbrace{1}_{a} \cdot 2^n n^k$$

סה"כ הוכח החסם האסימפטוטי העליון בהתאם להגדרתו.

- **חסם תחתון:** נוכיח $s = \Omega(2^n n^k)$.

$$\forall n \geq \underbrace{1}_{n_0}.s = \sum_{i=1}^{n} 2^i i^k = \underbrace{\sum_{i=1}^{n-1} 2^i i^k}_{\leq 2^n n^k} + 2^n n^k \leq \underbrace{2}_{a} \cdot 2^n n^k$$

נוכיח את טענת העזר שהסתמכנו עליה, $\sum_{i=1}^{n} 2^i i^k \leq 2^n n^k$. נוכיח באינדוקציה. בסיס $n = 1$: יתקיים $2^2 \cdot 2^1 = \sum_{i=1}^{1} 2^i i^k$. צעד: נניח באינדוקציה את נכונות הטענה על n ונוכיח על $n+1$:

$$\sum_{i=1}^{n} 2^i i^k = \underbrace{\sum_{i=1}^{n-1} 2^i i^k}_{\leq 2^n n^k \text{ By Induction}} + 2^n n^k \leq 2 \cdot 2^n n^k = 2^{n+1} n^k \leq 2^{n+1} (n+1)^k$$

כאשר המעבר האחרון נכון ממונוטוניות הפונקציה x^k. סה"כ האינדוקציה הושלמה, ובכך טענת העזר, כלומר האי־שוויון לעיל מתקיים ואכן $s = \Omega(2^n n^k)$ לפי הגדרה.

מתוך הגדרת Θ, הוכחנו $s = \Theta(2^n n^k)$ כדרוש. ■

ג) 1. הלולאה תתחיל מ־n ותגמור לאחר k איטרציות בהן כל פעם n קטן פי 2, עד ש-$n = 1$. נוכל למצוא את כמות האיטרציות הזו:

$$\begin{aligned} n \cdot 0.5^k &= 1 \quad \Big\downarrow \cdot 0.5^{-k} \\ n &= 2^k \quad \Big\downarrow \log_2 \\ \log n &= k \end{aligned}$$

בפנים, יש לולאה שתלויה ב־n, ושם בדיקת קיום i בתוך L – פעולה שבמקרה הגרוע ביותר תיאלץ לבצע מעבר על כל L, שגודלו n (בהתאם למצב של לולאת ה־while). נחשב:

$$Ans. = \sum_{i=1}^{\log n} \left(\frac{n}{2^i} \cdot (n+i) + 1\right) = \log n + n \sum_{i=1}^{\log n} \frac{(n+i)}{2^i} \leq \log n + \sum_{i=1}^{\infty} \frac{n+n}{2^i} = \log n + n \cdot 2n \cdot 1 = \boldsymbol{O(n^2)}$$

2. כדי לנתח את סיבוכיות לולאת ה־while הפנימית כתלות ב־n, נדרוש ש־k יוכפל i פעמים ב־2 עד שיתשווה ל־n, ומכיוון שהוא מתחיל מ־1 אז $1 \cdot 2^i = n \implies i = \log n$. סה"כ נסכום בתהאם ל־range של הלולאות:

$$Ans. = \sum_{i=0}^{n-500} \sum_{j=0}^{\log i} \log n = \log n \cdot \sum_{i=0}^{n-500} \log i \leq \log n (n-500) \log n = \boldsymbol{O(n \log^2 n)}$$

3. חיתוך של L עד $i+1$ ידרוש $i+2$ פעולות, וכן יהיה אורכו. הוספה לרשימה, השוואה ושינוי איבר ברשימה מתבצעים ב־$O(1)$.

$$Ans. = \sum_{i=0}^{n}(i+2) = 2n + \sum_{i=0}^{n} i = 2n + \frac{1}{2}(n^2+n) = \boldsymbol{O(n^2)}$$

המשך בעמוד הבא

א) .a נרצה למצוא את הייצוב העשרוני של `0 10000000000 11000…0`:

$$(-1)^0 \cdot 2^{2^{10}-1023} \cdot \left(1+\frac{1}{2}+\frac{1}{4}\right) = 2 \cdot 1.75 = \mathbf{3}$$

.b נרצה למצוא את הייצוג העשרוני של `1 10000000010 1000…00`:

$$(-1)^1 \cdot 2^{2^{10}+2^1-1023} \cdot \left(1+\frac{1}{2}\right) = 2^3 \cdot 1.5 = \mathbf{-12}$$

ב) למען הנוחות, נסמן $\mathrm{exp} = exponent,\ \mathrm{frc} = fraction$ נניח $sign = 0$, $-1023 \le n \le 1024, n \in \mathbb{N}$.

.a צ.ל. בין כל זוג מספרים סמוכים בתחום $[2^n, 2^{n+1})$ יש את אותו ההפרש. **נסמן את ההפרש הזה ב-Δ^n.**

הוכחה. יהי $N_1, N_2 \in [2^n, 2^{n+1})$ מספרים סמוכים שונים הניתנים לייצוג. למען הנוחות, נסמן $\Delta_{a,b} = N_b - N_a$. בה"כ נניח $N_1 < N_2$, ונוכיח $\Delta_{2,3} = \Delta_{1,2}$. ידוע מההרצאה שלכל float יש ייצוג יחיד, פרט ל-0. אזי, בתחום $[2^n, 2^{n+1})$. הכרח הוא $n = \mathrm{exp} - 1023$. נתבונן בייצוג הבינארי של frc_{N_1}. נדע, שהביט האחרון בו יהיה היחיד ששונה מזה של frc_{N_2}, כי זהו השינוי המזערי ביותר שנוכל לבצע (עם שינוי של $\frac{1}{2^{52}}$, לפי הנוסחה לחישוב frc). סה"כ:

$$\Delta_{1,2} = 1 \cdot 2^{\mathrm{exp}+1023} \cdot (1+\mathrm{frc}_{N_1}) - 1 \cdot 2^{n+1023} \cdot \left(1+\mathrm{frc}_{N_1}+\frac{1}{2^{52}}\right) = 2^n \left(\cancel{\mathrm{frc}_{N_1} - \mathrm{frc}_{N_1}} + \frac{1}{2^{52}}\right) = \frac{2^n}{2^{52}} = \mathbf{2^{n-52}}$$

■

.b נרצה למצוא פי כמה גדול הרווח בתחום $[2^{n+1}, 2^{n+2})$ מהרווח בתחום $[2^n, 2^{n+1})$. נמצא את היחס:

$$\frac{\Delta^{n+1}}{\Delta^n} = \frac{2^{n+1-52}}{2^{n-52}} = 2^{n+1-n-52+52} = \mathbf{2}$$

.c אם היינו מוסיפים ביט נוסף ל-fraction, אז השינוי הכי קטן שיכולנו לבצע בו היה מוסיף $\frac{1}{2^{53}}$ לערך של frc, ובהתאם לדרך בה חישבנו את ההפרש המינימלי בסעיף a – התשובה הייתה הופכת להיות $\frac{2^n}{2^{53}} = 2^{n-53}$. הסעיף השני מדבר על יחס והחלוקה ב-2^{53} מתבטלת (כמפורט בסעיף b על מספרים שונים) ולכן התוצאה תיוותר 2.

.d בשני החישובים הראשונים, החישוב מתבצע בעבור אובייקט `int`, כזה שמשנה את גודלו בהתאם לכמות המקום בזכרון שיצרוך. באחרון, משום שהמספר `2.0` הוא מטיפוס `float`, אזי לאחר העלאה בחזקת 53 פייתון ישמור על הטיפוס. ידוע שאפשר לייצג באופן מדוייק 2^{53} בזכרון בעבור $\mathrm{exp} = 52,\ \mathrm{frc} = 2^{52}$ (שאפשר לייצג בייצוג בינארי בגודל המתאים `1…000 ,00…110100` בהתאמה), אך נבחין לב שהרווח בתחום $[2^{52}, 2^{53})$ הוא $\Delta^{52} = 2^{52-52} = 2 \ge 1$, כלומר, המספר הבא שיהיה אפשר לייצג באמצעות `float`, יהיה $2^{52}+2$ ולא $2^{52}+1$, הוא החישוב שמתבקש מהמחשב לבצע. על־כן המחשב בוחר לעגל מטה למספר הקרוב ביותר שהוא יודע לייצג.

המשך בעמוד הבא

ד. ראשית כל, נשקיע 5^k פעולות בשביל ליצור רשימה באורך הזה. נמיר כל מחורזת למספר (זה יקח $O(k)$ עבור n מחרוזות כלומר $O(nk)$). אחרונה, תהיה לולאה שתרוץ 5^k פעמים ובתוכה פעולות בזמן ריצה קבוע, פרט ללולאה נוספת שתרוץ במהלך כל הרצת הקוד n פעמים ללא תלות בלולאת האב שלה (היא תבצע ריצה על כל איבר בתשובה הסופית, שניתן להניח שהיא תקינה). סה"כ תהיה הסיבוכיות $O(5^k + nk + n) = O(kn + 5^k)$. הפונקציה עומדת בדרישות הזכרון, כי כל מה שנשמר בזכרון זה זכרון העזר בגודל 5^k, וערכים בגודל קבוע כמו משתנה החזרה של לולאות ה־for.

ו. הפונקציה עומדת בדרישות הזמן כי היא כוללת לולאה שתרוץ 5^k פעמים, ובתוכה חישוב של int_to_string שיארך זמן ריצה של k ולולאה נוספת שתרוץ ב־$O(n)$ מאיך שהיא בנויה, כלומר סה"כ הסיבוכיות היא $5^k(n+k) = O(5^k nk)$ (כי כי כמובן שהחסם ההדוק קטן יותר). הפונקציה גם עומדת בדרישות הזכרון, כי פרט למשתנה העזר ttt שאורכו k תווים, כל השאר יתפוש מקום קבוע בזכרון.

המשך בעמוד הבא

2. b. בעבור כל אינדקס, נמיין את k האיברים שלאחריו בזמן ממוצע $k \log k = O(1)$, כך שעבור האינדקס ה־i כל האיברים לפניו כבר יהיו ממוינים (כלומר אין צורך למיין גם אותם), וסה"כ כך נמיין את הרשימה. סיבוכיות במקרה הכי גרוע $n \cdot k^2 = O(n)$ (כי k קבוע וחסום ב־100).