מבני נתונים ~ תרגיל בית 5 ~ 2025B

שחר פרץ

9 ביולי 2025

ת.ז.: 334558962
שם במודל: shaharperets

.... (1) ....

1. נניח שבאלגוריתם MedofMed מחלקים את המערך לתשיעיות, ובמקום לבחור את החציון נבחור את האיבר ה־i בגודלו (כאשר $9 \leq i \leq 1$). נמצא את ה־i המינימלי כך שזמן הריצה לינארי.

פתרון. בהרצאה, צוין ש־$T(n) = cn + T(\alpha n) + T(\beta n)$ מקיימת $T(n) = \mathcal{O}(n)$ אם $\alpha + \beta < 1$. נוכיח אמ"מ, כלומר נוכיח גם את הגרירה השנייה. נעשה זאת בקונטרא־פוזיטיב. נניח $\alpha + \beta \geq 1$, נפרק למקרים.

- אם $\alpha + \beta = 1$, אזי זוהי הוכחה זהה לזו ש־merge sort מסתיים בזמן $n \log n \neq \mathcal{O}(n)$.
- אם $\alpha + \beta > 1$, אז $\alpha > 0.5 \vee \beta > 0.5$, ולכן נוכל להניח בה"כ $\alpha > 0.5$. מתקיים:

$$T(n) = cn + T(\alpha n) + T(\beta n) \geq cn + T(\alpha n) > cn + T(0.5n)$$

ממשפט האב $T(n) = cn + T(0.5n)$ מקיים $T(n) = \Theta(n \log n)$ ולכן בפרט $T(n) = \Omega(n \log n)$ ומכאן נסיק $T(n) \neq \mathcal{O}(n)$.

סה"כ הוכחנו שקילות. נחזור לשאלה עצמה. בדומה להרצאה, סיבוכיות MedofMed ניתנת לחסימה ע"י:

$$T(n) = cn + T\left(n - \frac{ni}{22.5}\right) + T\left(\left\lceil \frac{n}{9} \right\rceil\right)$$

כאשר הביטוי חוסם במדויק שכן גודל המלבן:

$$\text{width} = \left\lceil \frac{1}{(9+1)/i} \left\lceil \frac{i}{9} \right\rceil \right\rceil \overset{n\to\infty}{=} \frac{ni}{90},\ \text{length} = 5 \implies \text{size} = \frac{ni}{90} \cdot 5 - 1 - 5 \overset{n\to\infty}{=} \frac{ni}{18}$$

ועלות הקריאה הרקורסיבית למציאת האיבר ה־i בגודלו היא כמובן $T(\lceil \frac{n}{9} \rceil)$. עתה נדרוש ש־:

$$1 - \frac{i}{18} + \frac{1}{9} < 1 \iff \frac{i}{18} > \frac{1}{9} \iff i > 2$$

ה־i הטבעי המינימלי שייקים זאת הוא $\boldsymbol{i = 3}$. ■

2. נחזור על הסעיף הקודם בעבור חלוקה לשביעיות.

פתרון. בדומה לסעיף הקודם:

$$\text{width} = \left\lceil \frac{1}{(7+1)/i} \left\lceil \frac{i}{7} \right\rceil \right\rceil \overset{n\to\infty}{=} \frac{ni}{56},\ \text{length} = 4 \implies \text{size} = \frac{ni}{56} \cdot 4 - 1 - 4 \overset{n\to\infty}{=} \frac{ni}{14}$$

כלומר נקבל:

$$T(n) = cn + T\left(n\left(1 - \frac{i}{14}\right)\right) + T\left(\left\lceil \frac{n}{7} \right\rceil\right)$$

ונדרוש:

$$1 - \frac{i}{14} + \frac{1}{7} < 1 \implies \frac{i}{14} > \frac{1}{7} \implies \mathbb{N} \ni i > 2 \implies i = 3$$

גם כאן נקבל ש־$i = 3$ המינימלי. עם זאת, התבקשנו למצוא מקסימלי. משום ברמת הקבועים, מציאת האיבר ה־i בגודלו ומציאת האיבר ה־$(7+1) - i$ בגודלו מסמטריה, אז ה־i הטבעי המקסימלי הוא $\mathbf{5} = 3 - 8$. ■

המשך בעמוד הבא

. . . . (2) . . . .

נתון מערך בגודל n, המכיל n איברים שונים זה מזה. נתון ששלכל $i \in [n]$, האיבר ה־i בגודלו נמצא בטווח $[i-k, i+k] \cap \mathbb{N}$ עבור $k \in [n]$ כלשהו.

1. נוכיח ש־insertion sort ימיין את המערך ב־$\mathcal{O}(n \log k)$.

הוכחה. לשם כך, נוכיח טענת עזר: בעבור A מערך המקיים את התכונות לעיל, $I(A) \leq cnk$ עבור c קבוע כלשהו, כאשר $I(A)$ מוגדר להיות $I(A) = \#\{i < j \in [n] \colon A_i > A_j\}$. נתבונן באיבר ה־$i$ במערך מסודר. אזי ב־A, במקרה הגרוע, יהיה במיקום $i+k$ או במיקום $i-k$. במקרה הראשון, יהיו לכל היותר $2k-1$ שיכולים להיות גדולים ממנו ב־A אך לא בסדר ממוין, שכן הם בהכרח מצויים בטווח $[i-k, i+k) \cap \mathbb{N}$ ובמקרה השני יכולים להיות לכל היותר $2k-1$ מספרים שקטנים ממנו, באופן דומה. סה"כ נקבל, אם נסכום על כל $i \in [n]$:

$$I(A) \leq n(2k-1) = 2nk - n \leq 2nk$$

ואכן הראינו את הדרוש בעבור $c = 2$.

ניגש להוכיח את הטענה עצמה. בתרגיל בית 3, הוכחנו בסעיף האחרון ש־merge sort פועל בסיבוכיות:

$$O\left(n \log\left(\frac{I(A)}{n} + 2\right)\right) = \text{cost} \overset{\forall n \geq n_0}{\leq} \gamma \cdot n \log\left(\frac{I(A)}{n} + 2\right) \leq \gamma n \log\left(\frac{2nk}{n} + 2\right) = \gamma n \log(2k+2) \leq \gamma n \log k$$

(כאשר γ קבוע ממשי) סה"כ $\text{cost} = \mathcal{O}(n \log k)$ כדרוש. ■

2. עבור $n = \sqrt{n}$, נוכיח $\Omega(n \log n)$ למיון המערך.

הוכחה. נניח בשלילה שעבור $n = \sqrt{n}$ אפשר למיין בסיבוכיות $\neq \Omega(n \log n)$.

נתבונן בבעיה הבאה: יהיו $(A^i)_{i=1}^{\sqrt{n}}$ מערכים מגודל $\sqrt{n}$. בעזרת אלגו' המיון שהנחנו בשלילה את קיומו, נוכל לקחת את איברי כל המערכים, לסמן כל אחד מהם במספר מהערך שמהם הגיעו, ולשרשר אותם יחדיו, ולקבל $A = \bigoplus_{i=1}^{\sqrt{n}} (\langle i, A_j^i \rangle)_{j=1}^{\sqrt{n}}$, כלומר:

$$A = \langle 1, A_1^1 \rangle \cdots \langle 1, A_{\sqrt{n}}^1 \rangle, \langle 2, A_1^2 \rangle \cdots \langle 2, A_{\sqrt{n}}^2 \rangle \cdots\cdots \langle \sqrt{n}, A_1^{\sqrt{n}} \rangle \cdots \langle \sqrt{n}, A_{\sqrt{n}}^{\sqrt{n}} \rangle$$

נמיין אותם בסדר הלקסיקוגרפי על $\langle i, j \rangle$, ובכך נקבל את $\sqrt{n}$ המערכים שלנו ממויינים כדרוש, בפחות מ־$\Omega(n \log n)$. הפעלת האלגו' שהנחנו בשלילה את קיומו חוקית, כי כל איבר נמצא במרחק $\sqrt{n}$ לכל היותר ממקומו המקורי, במקרה הגרוע בו המערך ה־A_i הפוך לסדר המסודר שלו (נובע ישירות מהבנייה שלנו ומהגדרת הסדר הלקסיקוגרפי).

לעומת זאת, ידוע שאת בעיית המיון בעבור A_i ניתן לפתור ב־$\Omega(\sqrt{n} \log \sqrt{n})$. נכפיל זאת ב־$\sqrt{n}$ הפעלות ונקבל:

$$\sqrt{n}\Omega(\sqrt{n} \log \sqrt{n}) = \Omega((\sqrt{n})^2 \log n^{0.5}) = \Omega(0.5 n \log n) = \Omega(n \log n)$$

כלומר, החסם התחתון לבעיה זו הוא $\Omega(n \log n)$ על אף שתחת הנחת השלילה פתרנו אותה בפחות מכך. סתירה, כדרוש. ■

המשך בעמוד הבא

בהינתן A, B מערכים בגודל n, המכילים איברים מתחום סדור כלשהו, נרצה למצוא מערך C כך ש־C_i הוא מספר האיברים ב־A הקטנים או שווים ל־$B[i]$.

א. נמצא אלגו' לחישוב C

תשובה. ראשית כל, נמיין את A באמצעות merge sort בסיבוכיות $\mathcal{O}(n \log n)$. לאחר מכן, נבצע את ההשמה הבאה:

$$\forall i \in [n]\colon C\texttt{[}i\texttt{]} \leftarrow \texttt{Search(}A\texttt{, }B\texttt{[}i\texttt{])} + 1$$

כאשר `Search` מבצע חיפוש בינארי של $B\texttt{[}i\texttt{]}$ ב־A (משום ש־A מוין, הוא מחזיר כמה איברים ב־A קטנים מ־$B\texttt{[}i\texttt{]}$). עלות החיפוש הבינארי היא $\mathcal{O}(\log n)$ ואנו מבצעים אותו n פעמים, סה"כ $\mathcal{O}(n \log n)$ לפעולה זו. סיבוכיות כוללת $= \mathcal{O}(n \log n) + \mathcal{O}(n \log n)$ $.\mathcal{O}(n \log n)$ ■

ב. נוכיח חסם תחתון הבדוק במודל ההשוואות לבעיה.

הוכחה. ישנו חסם תחתון $\Omega(n \log n)$ לבעיה: משום ש־$C[i] \in [n-1]$ (לא יכולים להיות יותר מ־n איברים שקטנים או שווים מאיבר נתון, כאשר ישנם רק n איברים), אזי $C \in [n] \to [n-1]$ ולכן ישנם n^{n-1} אפשרויות ל־output של האלגו'. לכן, יש $\ell = n^{n-1}$ עלים בעץ ההשוואות, ועומקו $\Omega(\log n^{n-1}) = \Omega((n-1)\log n) = \Omega(n \log n)$ מה שגם מהווה חסם תחתון על זמן הריצה של האלגו'. חסם זה הדוק שכן הראינו קיום אלגו' במודל ההשוואות בסעיף א'. ■

המשך בעמוד הבא

א. נממש את ה־ADT הבא: `ApproxMedian(Q)` , `Delete(`Q, X`)` , `Insert(`Q, k`)`כאשר `AprroxMedian` מחזיר משפר הגדול מלפחות $\frac{n}{4}$ מאיברי Q וקטן מלפחות $\frac{n}{4}$ מהם.

פתרון. ניעזר ברשימה מקושרת עצלה. הפעולה `Insert` תכניס אליה את k למבנה Q. הפעולה `Delete` תמחק את x מהמבנה, בצורה עצלה: כלומר, תסמן אותו כאיבר מחוק, אך לא תזיז את הרשימה בפועל. כבר הוכחנו בהרצאה שפעולות אלו פועלות באמורטייז $\mathcal{O}(1)$ (כאשר `Insert` מסוגל לתקן את הרשימה במעבר על איברים מחוקים).

ננקט בשיטה הבאה כדי לתחזק שדה Median: נחזיק counter, שבכל פעם שעובר חזקה של שתיים, הוא מחשב Median באמצעות MedofMed. הערה לגבי מימוש: יש צורך לדלג על איברים מחוקים. נעשה זאת בכל הוספה שבא $\lfloor \log n \rfloor$ גדל ב־1, ובכל מחיקה שבא $\lfloor \log(0.75n) \rfloor$ (הכפל בקבוע נועד כדי למנוע רצף של חיסורים ואז הוספות שבכל אחד מהם מחשבים Median).

- **סיבוכיות:** הוספת חישוב ב־$\mathcal{O}(n)$ לפעולה בכל פעם לאחר 2^i ריצות, שקול לחלוטין להעתקת מערכים שמכפילים את עצמם, מה שהוכחנו כבר שלוקח $\mathcal{O}(1)$ אמורטייז. נוכל להוכיח זאת בשיטת הבנק, כאשר מוסיפים מספר קבוע של 2 מטבעות על כל איבר שמוסיפים, וכן 2 מטבעות על כל איבר שמוחקים. נשתמש במטבעות שעל המחוקים כדי לעבור עליהם ולהוציא אותם מהרשימה, ומשום שיש לפחות $\frac{n}{2}$ איברים שנוספו מאז הקריאה האחרונה ל־MedofMedians, ועליהם שתי מטבעות, יהיה לנו n מטבעות כדי לחשב את החציון. נשמור אותו בשדה המתאים.
- **תקינות:** משום שהחציון מחושב כל לכל היותר חזקה של שניים, במקרה הגרוע בהוספה הבאה המערך יצטרך לחשב חציון מחדש, כלומר החציון חושב לפני $\frac{n+1}{2}$ הוספות. בפרט, אז היה החציון של $\frac{n+1}{2}$ האיברים, ובמקרה הגרוע שבו כל האיברים שנוספו מאז גדולים ממש או קטנים ממש ממנו, החציון כרגע עם $\frac{n+1}{4} > \frac{n}{4}$ איברים לפניו/אחריו.

לסיום, הקריאה ל־`AprroxMedian` תחזיר את החציון שחושב ונשמר בשדה. ■

ב. נוכיח אי־קיום מימוש ל־ADT עם הפעולות `Median(`Q`)` , `Insert(`Q, k`)` כאשר שתיהן פועלות ב־$\mathcal{O}(1)$ אמורטייז, תחת מודל ההשוואות.

פתרון. נניח בשלילה שקיים מבנה מתאים. יהיו $a_1 \dots a_n \in \mathbb{N}$. נעשה `Init` ל־$Q$ ונפעיל `Insert(`$Q, -a_i$`)` , `Insert(`Q, a_i`)` לכל $i \in [n]$ (אומנם $a_i \in \mathbb{N}$, אך יחס הסדר שאנחנו עובדים מעליו הוא $<_{\mathbb{Z}}$). עתה, החציון עומד עם המינימום. נמצא את המקסימום הנוכחי m, ונבצע את הסדרה הבאה:

`Insert(`$Q, m{+}1$`)` , $b_1 \leftarrow$ `Median(`Q`)` , `Insert(`$Q, m{+}2$`)` , `Insert(`$Q, m{+}3$`)` , $b_2 \leftarrow$ `Median(`Q`)` , `Insert(`$Q, m{+}4$`)` , `Insert(`$Q, m{+}5$`)` , $b_3 \leftarrow$ `Median(`Q`)` , ... `Insert(`$Q, m{+}2n$`)` , `Insert(`$Q, m{+}2n{+}1$`)` , $b_n \leftarrow$ `Median(`Q`)`

בסדרה זו ישנן $3n$ פעולות שאורכות $\mathcal{O}(1)$ אמורטייז, כלומר סה"כ אורכות $\mathcal{O}(n)$. נבחין ש־$b_1 \dots b_n$ הם $a_1 \dots a_n$ מסודרים, שכן לבצע שתי הוספות גדולות מהחציון (ובפרט ביצוע הוספות גדולות מהמקסימום) שקולות ללעשות $median \leftarrow$ `successor(`$median$`)` (כלומר שהחציון יהפוך להיות העוקב של עצמו). עוד נבחין ש־$b_n = m$ כמצופה. סה"כ מיינו n איברים בסיבוכיות $\mathcal{O}(n)$ בתוך מודל ההשוואות, בסתירה לחסם תחתון $\Omega(n \log n)$. ■

המשך בעמוד הבא

א. נתון עץ השוואות המתאים לאלגוריתם הממיין n מספרים. נמצא את העומק המינימלי של עלה בעץ.

הוכחה. נגיע לעלה כאשר האלגו' ביצע מספיק השוואות כדי להיות בטוח בתוצאה. נוכיח שקיים ומינימלי עלה בעומק $n-1$.

- **קיום:** עבור מערך מסודר, רצף ההשוואות שמשווה את האיבר ה־i ל־$i+1$, המתחיל ב־1 ומסתיים ב־$n-1$, מוכיח לאלגו' שמערך ממוין ובכך מאפשר לעשות halting. כלומר, מצאנו מקרה ועלה מעומק $n-1$.
- **מינימליות:** נניח בשלילה שקיים מסלול קצר יותר. משום שהאלגו' ממיין את הקלט, אז הוא בפרט מוצא את המינימום בעץ. כלומר, מצאנו את המינימום במערך תוך $a \leq n-2$ השוואות. בפרט, לא השוונו את המינימום לכל האיברים באופן ישיר או עקיף (כלומר, קפלסיבית או טרנזטיבית) וייתכן איבר גדול יותר מהמינימום במערך, מה שמהווה סתירה. לכן אורך מסלול הוא לפחות $n-1$.

סה"כ הראינו ש־$n-1$ העומק המינימלי של עלה בעץ, והראנו שחסם זה התחתון ההדוק ביותר שניתן לתת. ■

ב. נתון אלגו' השוואות המקבל כקלט שתי רשימות ממויינות באורך n כל אחת, וממזג אותן לרשימה ממוינת אחת.

1. נראה שמספר העלים בעץ הוא לפחות $\binom{2n}{n}$.

הוכחה. כדי להראות שקיימים $\binom{2n}{n}$ עלים בעץ, באופן שקול נוכיח שבהינתן מערכים $(a_i)_{i=1}^n$ ו־$(b_i)_{i=1}^n$ כלשהם, ישנם $\binom{2n}{n}$ פלטים אפשריים. נתבונן ב־$(a_i)_{i=1}^n$ – הסדר שלהם מקובע, ובהכרח נצטרך בפלט לשבץ את איברי $(b_i)_{i=1}^n$ בינהם. יש $n+1$ מקומות בינהם, והבעיה שקולה קומבינטורית לחלוקת n איברים ל־$n+1$ דליים, מה שמותיר אותנו עם $\binom{2n}{n} = \binom{n+1+n-1}{n} = S(n+1, n)$ (סימון מקורס בדידה 2) אפשרויות כדרוש. מספר עלי העץ כמספר הפלטים האפשריים מהגדרה. ■

2. נסיק חסם תחתון לגבי זמן הריצה.

הוכחה. בהרצאה ראינו למה, על־פיה בהינתן ℓ עלים, גובה עץ בינארי הוא בהכרח $h \geq \log \ell$. משום שעץ ההשוואות הוא בפרט עץ בינארי, מתקיים שגובהו:

$$h \geq \log \ell \geq \log \binom{2n}{n} = \log \left((-1)^n 4^n \binom{-1/2}{n} \right) = \log \left(4^n \left| \binom{-1/2}{n} \right| \right) > \log 4^n = n \log 4 = \Omega(n)$$

סה"כ הוכחנו חסם תחתון לינארי. (את הנוסחה לבינום האמצעי רואים בבדידה 2) ■

המשך בעמוד הבא

נתון מערך A של n מספרים שלמים. ננסה למצוא אלגו' הבודק האם קיימים $i,j\in[n]$ כך ש־$|A_j-A_i|=j-i$.

1. בסעיף זה, נניח $\forall i\in[n]\colon A_i\in[-n^2,n^2]\cap\mathbb{Z}$.

פתרון. נספק אלגו' דטרמיניסטי בסיבוכיות לינארית. ניצור מילון הממומש באמצעות vector בגודל n, כשערכיו $i\mapsto A_i$. נבחין ש־:

$$j-i=|A_j-A_i| \iff \begin{cases} j-i=A_j-A_i \\ j-i=-A_j+A_i \end{cases} \iff \begin{cases} j-A_j=i-A_i \\ A_j+j=A_i+i \end{cases}$$

לכן, נוכל לבדוק באופן שקול שוויון זוגות כמתוארים לעיל. ניצור שני מערכים:

$$\mathcal{A}_1[i]=i-A_i$$
$$\mathcal{A}_2[i]=i+A_i$$

קיום כפילות ב־$\mathcal{A}_1$ או ב־$\mathcal{A}_2$, שקול לקיום $i\neq j\colon A_j+j=A_i+i\vee A_i-i=A_j-i$, או בניסוח שקול, $j-i=|A_j-A_i|$, שזה מה שאנו רוצים לבדוק. לשם בדיקת הכפילויות, נמיין את $\mathcal{A}_1,\mathcal{A}_2$ באמצעות radix sort על איבריהם בבסיס n. משום שהערכים בטווח $-n^3<-n^2-n\cdots n^2+n<n^3$, ומתקיים שכמות הספרות של n^3 בבסיס n היא $\log_n n^3=3\log_n n=3$ (קבועה), נצטרך להריץ את count sort הלינארי כמות קבועה של פעמים ולכן ה־radix sort יפעל בסיבוכיות לינארית. לאחר המיון, נותר לבדוק האם קיימים שני איברים עוקבים זהים ב־$\mathcal{A}_1,\mathcal{A}_2$, כלומר לעבור עליהם $n-1=\mathcal{O}(n)$ פעמים, ולבדוק האם $\mathcal{A}_i[j]=\mathcal{A}_[j+1]$ (עבור $i\in\{1,2\},j\in[n-1]$).

משום שביצעו פעולות לינאריות בלבד, בסידור, השוואה, ויצירת $\mathcal{A}_1,\mathcal{A}_2$, סיבוכיות האלגו' שלנו לינארית דטרמיניסטית במקרה הגרוע. ■

2. עתה, נוותר על ההנחה, ונפתור בתוחלת.

פתרון. נעזר בשיטה דומה לזו של הסעיף הקודם, כלומר, נוכל לבנות טבלת hash בגודל n בשיטת ה־chaining, ולשמור בה את הזוגות הסדורים (i,A_i+i) (יש n כאלו ולכן בתוחלת יקח $\mathcal{O}(n)$ לשמור אותם) כאשר ה־hash מקבל כקלט את A_i+i. כדי למצוא איברים מגודל זהה, נעבור ב־$\mathcal{O}(n)$ על כל המספרים ונבדוק האם בתת־הרשימה המקושרת בתא ה־$h(A_i+i)$ יש עוד איבר מלבדם (בתוחלת יהיה שם $\mathcal{O}(1)$ איברים, כלומר המעבר יארך $n\cdot\mathcal{O}(1)$) בעל אותו הערך. באופן דומה נוכל לשמור מערך עם הערכים $(i,i-A_i)$ וב־$\mathcal{O}(n)$ לבדוק עבור כל j האם קיים אחר מלבדו בתא ה־$h(i-A_i)$ בעל אותו הערך.

סה"כ סיבוכיות $\mathcal{O}(n)$ בתוחלת, וטבלת ה־hash מגודל לינארי. ■

שחר פרץ, 2025

קומפל ב־LaTeX ונוצר באמצעות תוכנה חופשית בלבד