מבני נתונים
פתרון תרגיל 5

גלעד אשרוב

25 באפריל 2013

שאלה 1.

א. תאר אלגוריתם המקבל כקלט רשימה מקושרת ממויינת ומחזיר רשימת דילוגים אידיאלית *ideal skip list*. מהו זמן הריצה של האלגוריתם שהצגת?

פתרון: האלגוריתם פשוט עובר בלולאה, ומעלה כל איבר שני "רמה". כלומר, לוקחים את הרשימה התחתונה (הרשימה המקורית), ומעלים כל איבר שני. לאחר שבנינו את הרמה השנייה, אנחנו מפעילים שוב את האלגוריתם שהצגנו כעת על רמה זו (כלומר, שוב מעלים כל איבר שני רמה). מסיימים לאחר $\log n$ רמות. סה"כ זמן ריצה:

$$n + n/2 + n/4 + n/8 + \ldots + 1 \leq 2n$$

כלומר, האלגוריתם רץ בזמן $O(n)$.

ב. מה ההסתברות שבהכנסת n איברים לרשימה, נגיע למצב שבו מספר הרמות מגיע ל - $\sqrt{n}$?

פתרון: כפי שראינו בשיעור, ההסתברות שאיבר בודד יעלה k רמות הוא 2^{-k}, וההסתברות שאחד מ-n האיברים יעלה יותר מ-k רמות הוא קטן מ-$n2^{-k}$ לפי חסם האיחוד. לפיכך, ההסתברות שמספר הרמות יעבור $\sqrt{n}$ קטן מ:

$$n \cdot 2^{-\sqrt{n}} = n^{\sqrt{n}/\log n - 1}$$

וזוהי הסתברות זניחה (קטנה מכל פולינום).

ג. הציגו אלגוריתם לאיחוד שתי רשימות דילוגים. מהו זמן הריצה? הסבר מדוע הפתרון שלך הוא האופטימלי.

פתרון: נסתכל פשוט על הרמה התחתונה. נקח את שתי הרשימות, שתיהן ממויינות, ונמזג אותן לרשימה ממויינת אחת. אם ברשימה הראשונה n איברים, ובשנייה m איברים, המיזוג ייעלה לנו $O(n+m)$. לאחר מכן, נבנה רשימת דילוגים אידיאלית על הנתונים האלה (כמו בסעיף א') - בעלות $O(n+m)$. זהו הזמן האופטימלי, מכיוון שרק לקרוא את הנתונים עולה $O(n+m)$.

שאלה 2. נתון מערך A של מספרים **חיוביים** (ושונים מ-0) בגודל n. השאילתא שעליה נרצה לענות בצורה מהירה היא:

$\boldsymbol{mult(i)}$: החזר את מכפלת כל המספרים במערך מלבד $A[i]$. כלומר, החזר $\prod_{j \neq i} A[j]$.

1. תארו אלגוריתם המבצע עיבוד מקדים על המערך[1], כך שניתן לענות על שאילתא בזמן $O(1)$. אלו נתונים אנו שומרים? מהי עלות העיבוד המקדים?

   פתרון: העיבוד המקדים פשוט יעבור על המערך ויחשב את מכפלת כל המספרים. נשמור את המכפלה בצד, במשתנה $mult$. בזמן שאילתא על אינדקס i, פשוט נחשב: $mult/A[i]$.
   עלות עיבוד מקדים: $O(n)$. עלות שאילתא - $O(1)$.

2. לסעיפים הבאים אנו מניחים שהמחשב לא יודע לבצע פעולות חילוק. בכל סעיף יש לציין מהי עלות העיבוד המקדים. תארו את תהליך העיבוד המקדים, כך שעלות שאילתא תהיה $O(\sqrt{n})$. יש לציין במפורש כיצד ייתבצע העיבוד המקדים, וכיצד תתבצע שאילתא. (**רמז:** חישבו על "נציגים")

   פתרון: נחלק ל-$\sqrt{n}$ קבוצות בגודל $\sqrt{n}$ כל אחת, ונשמור מערך בגודל $\sqrt{n}$ שישמור את כל המכפלות של המספרים בכל קבוצה. כלומר, נשמור מערך B בגודל $\sqrt{n}$, כך ש:

$$B[i] = \prod_{k=i\sqrt{n}}^{i\sqrt{n}+\sqrt{n}-1} A[k]$$

   בנוסף, נשמור אינדקס התחלה וסוף של כל קבוצה.
   בהינתן שאילתא, נחשב את מכפלת כל המערך B מלבד הקבוצה בה נמצא האיבר $B[i]$ (ניתן לעשות זאת בלולאה, ונדע מתי מתחילה הקבוצה שבה-i נמצא בזכות האינדקסי התחלה וסוף ששמרנו לכל תא). בנוסף, נחשב את מכפלת כל איברי הקבוצה בה נמצא $A[i]$, מלבד האיבר $A[i]$.
   עלות העיבוד המקדים היא $O(n)$. בשאילתא אנחנו מבצעים $2\sqrt{n}$ מכפלות, ולכן עלות $O(\sqrt{n})$.

3. תארו תהליך של עיבוד מקדים ואלגוריתם לשאילתא, כך שעלות שאילתא תהיה $O(\sqrt[3]{n})$.

   פתרון: בדומה למה שראינו בכיתה. מחלקים את המערך לקבוצות בגודל $\sqrt[3]{n}$. מחשבים מכפלה של כל קבוצה כזו, ושומרים הכל במערך B. גודל המערך B הוא - $n/\sqrt[3]{n} = \sqrt[3]{n^2}$. כעת, נחלק גם את המערך B לקבוצות בגודל $\sqrt[3]{n}$, נקבל $\sqrt[3]{n}$ קבוצות. נשמור גם אינדקסים בגל קבוצה.
   עלות בנייה - $O(n)$. שאילתא מתבצעת בצורה דומה לסעיף הקודם. עלות $3\sqrt[3]{n}$ מכפלות, ולכן עלות כוללת $O(\sqrt[3]{n})$.

4. הכלילו את התהליך, כך שעלות שאילתא תהיה $O(k\sqrt[k]{n})$. מהי כמות הזיכרון הנצרכת? מהו k האופטימלי? מהי עלות שאילתא במקרה זה?

   פתרון: בצורה דומה. נחלק את המערך לקבוצות בגודל $\sqrt[k]{n}$ כל קבוצה. נקבל סה"כ $n/\sqrt[k]{n} = n^{1-1/k}$ קבוצות. גם אותם נחלק לקבוצות בגודל $\sqrt[k]{n}$, נקבל $n^{1-2/k}$ קבוצות. נמשיך כך k רמות, עד שנקבל ברמה העליונה $n^{1/k} = \sqrt[k]{n}$ איברים. עלות שאילתא תהיה - $O(k\sqrt[k]{n})$.
   אם נגזור כדי למצוא את k האופטימלי, נקבל $k = \log n$ ואז: $O(k\sqrt[k]{n}) = O(\log n \cdot \sqrt[\log n]{n}) = O(2\log n)$, וזוהי עלות השאילתא.
   עלות העיבוד המקדים - $O(n)$.

5. חשבו על רעיון אחר לחלוטין, כך שעלות שאילתא תהיה $O(1)$, ועלות העיבוד המקדים - $O(n)$.

[1] כלומר, אנו מניחים שני שלבים - בשלב הראשון אנחנו מקבלים את המערך כקלט, מבצעים עליו איזשהו עיבוד, ושומרים את התוצאות. בשלב השני, אנחנו מקבלים סידרה של שאילתות שעליהן נרצה לענות בצורה יעילה ומהירה בזכות המידע ששמרנו בשלב העיבוד המקדים.

נשמור שני מערכים B, C, שניהם מגודל n. במערך B נשמור:

$$B[i] = \prod_{k=1}^{i} A[k] \qquad \text{and} \qquad C[i] = \prod_{k=i}^{n} A[k]$$

קל לראות שניתן לבנות את המערכים ב-$O(n)$: בכדי לחשב את $B[i]$, פשוט נחשב $B[i-1] \cdot A[i]$. לכן, ניתן לרוץ בלולאה מ-1 ועד n בכדי לחשב את B. בצורה דומה, ע"י לולאה הפוכה, ניתן לחשב את המערך C.

כעת, בהינתן שאילתא i, נחזיר $B[i-1] \cdot C[i+1]$ ונקבל:

$$B[i-1] \cdot C[i+1] = \prod_{k=1}^{i-1} A[k] \cdot \prod_{k=i+1}^{n} A[k] = \prod_{k \neq i} A[i]$$

עלות שאילתא היא $O(1)$ (ניתן לאחר חישוב המערכים הנ"ל לשמור מערך של תוצאות ועלות שאילתא תהיה גם כן -$O(1)$).

שאלה 3. (ממבחן + תוספת) מספרי פיבונאצ'י מוגדרים ע"י

$$F_i = F_{i-1} + F_{i-2}, \qquad F_0 = 0, F_1 = 1$$

1. כתבו אלגוריתם רקורסיבי הפותר את הבעיה (בצורה הכי נאיבית). מהי סיבוכיות האלגוריתם? כתבו נוסחא המתארת את זמן הריצה של האלגוריתם, ותנו חסם עליון וחסם תחתון הדוק ככל שניתן.

   פתרון: האלגוריתם פשוט בודק תנאי סיום (האם הקלט הוא - 0 או 1) ומחזיר תוצאה בהתאם. אחרת - האלגוריתם קורא לעצמו רקורסיבית על קלט $i-1$, ואז קורא לעצמו שנית על קלט $i-2$ ומחזיר את סכום התוצאות.
   זמן הריצה הוא: $T(n) = T(n-1) + T(n-2) + 1$. קשה לפתור נוסחא זו סתם כך, ולכן נשים לב:

   $$T(n-1) \geq T(n-2)$$

   ולפיכך:

   $$T(n) = T(n-1) + T(n-2) + 1 \geq 2T(n-2) + 1$$

   מהצד השני,

   $$T(n) = T(n-1) + T(n-2) + 1 \leq 2T(n-1) + 1$$

   נפתור כל נוסחא בנפרד, ונקבל חסם עליון וחסם תחתון ל-$T(n)$. (החסמים - $[2^{n/2}+1, 2^n+1]$)

2. נרצה לעשות זאת מהר יותר. לכל האורך, נניח שכל פעולת כפל, וכמו־כן - כפל של שתי מטריצות מגודל 2×2 עולה $O(1)$ זמן.

   תהי $A = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$. נשים לב שמתקיים: $\begin{pmatrix} F_{n-1} \\ F_{n-2} \end{pmatrix} \cdot \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} F_n \\ F_{n-1} \end{pmatrix}$.

   הסק מכאן נוסחא ל־ $\begin{pmatrix} F_n \\ F_{n-1} \end{pmatrix}$ כפונקציה של־A, ושל $\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} F_1 \\ F_0 \end{pmatrix}$. הסק מכאן אלגוריתם לחישוב מספר פיבונאצ'י. מהי עלות האלגוריתם?

פתרון: כדי לחשב $F(n)$ פשוט נחשב $A^n \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix}$. העלות היא k העלאות בחזקה. העלות היא פשוט $O(n)$.

3. נניח וחישבנו את A^8. האם צריך עוד - 8 פעולות של כפל מטריצות בכדי לחשב את A^{16}? הראה/י כיצד לעשות כן בפעולה אחת.

פתרון: ברור שלא. פשוט נעלה בריבוע.

4. לפי אותו עיקרון כמו בסעיף הקודם, כמה פעולות נדרשות בכדי לחשב את A^8? הכלל לכל k, כאשר k חזקה שלמה של 2 (כלומר, הצג אלגוריתם). בנוסף, הצג נוסחא רקורסיבית לזמן הריצה, ופתור אותה (בעזרת אחת השיטות שלמדנו).

פתרון: נחשב $((A^2)^2)^2$, כלומר, 3 פעולות.

5. איך נחשב את A^5? ואת A^{12}? הכלל לחזקה k כלשהי, לא בהכרח חזקה שלמה של 2.

פתרון: נחשב את A^4 ע"י 2 פעולות, ונכפיל ב-A. עבור 12, נשים לב ש- $12 = 8 + 4$, לפיכך נחשב A^8 ואת A^4 (3 פעולות לראשון, 2 פעולות לשני) ונכפיל בינהם. בסה"כ נקבל 5 פעולות.

6. סיכום - בהינתן k - כמה עולה האלגוריתם שהצגת לחישוב F_k?

פתרון: באופן כללי נסתכל על הייצוג הבינארי של -k, ונחשב את כל החזקות של A עד ל-$2^{\log k}$. כלומר, חישבנו את $A, A^2, A^4, A^8, \ldots$. כעת, בהינתן ייצוג בינארי של -k, נגיד 110101, אנחנו יודעים בדיוק אילו מטריצות לקחת לחישוב, ואילו לא. במקרה של הדוגמא הנ"ל, נקח את A^1, A^4, A^{16}, A^{32}. עלות חישוב כל החזקות - $O(\log k)$. עלות הכפלת כל המטריצות (יש לנו $\log k$ איברים בייצוג) - $O(\log k)$, ולכן סה"כ עלות - $O(\log k)$. תשובה תתקבל (ניקוד מלא) גם אם לא שומרים בצד את כל המכפלות, אלא מחשבים מחדש כל חזקה בעלות $\log k$, (כלומר, כדי לחשב A^{16} לא משתמשים בחישוב A^8 שכבר חישבנו, אלא מחשבים אותו מחדש רקורסיבית). במקרה כזה, נגיע לעלות של - $O(\log^2 n)$.