מבני נתונים 89-120
תרגיל 2 - פתרונות

גלעד אשרוב

24 באפריל 2014

שימו לב: הפתרונות המובאים כאן הם פתרונות חלקיים.

שאלה 1. נניח שישנה רשימה המצביעה במקום כלשהו אל תוך רשימה אחרת. כלומר, תהי:

$$L_1 = a_1 \to \ldots \to a_k \to x_1 \to \ldots \to x_n$$

ותהי

$$L_2 = b_1 \to \ldots \to b_\ell \to x_1 \to \ldots \to x_n$$

כלומר, לשתי הרשימות יש התחלה שונה, והחל ממקום מסויים, שתיהן מתמזגות לאותה רשימה. בכל שלושת הסעיפים הבאים, אינכם מורשים לשנות את הרשימות, ועל האלגוריתמים להשתמש ב־$O(1)$ זיכרון נוסף.

1. הצג אלגוריתם המקבל כקלט את הראשים של שתי הרשימות (a_1, b_1) ומחזיר את נקודת המפגש (x_1). האלגוריתם צריך לרוץ בזמן $O(|L_1| + |L_2|) = O(k + \ell + 2n)$.
2. במקרה ש k ו ־ ℓ קטנים ממש מ ־ n, האלגוריתם שהצגת בסעיף 1 הוא בזבזני. יהי $d \stackrel{\text{def}}{=} \max\{k, \ell\}$. הצג אלגוריתם הרץ בזמן $O(d^2)$. (הצג פסאדו־קוד).
3. חשוב על שיפור לאלגוריתם שהצגת בסעיף הקודם, והצג אלגוריתם שרץ בזמן $O(d)$ (שוב, יש להציג פסאדו־קוד).

פתרון:

1. נעבור על הרשימה הראשונה ונמצא את אורכה (כלומר, נרוץ על כל האיברים ונספור כמה צעדים עלינו לעבור עד שנצביע ל־$NULL$). כעת, מצאנו למעשה את $k + n$. בצורה דומה, נמצא את $\ell + n$, ע"י ריצה על הרשימה השנייה. בהינתן שני הערכים $k + n, \ell + n$, נוכל למצוא איזו רשימה ארוכה יותר, ולמצוא את ההפרש בין ארכי הרשימות. נניח בלי הגבלת כלליות שהרשימה L_1 היא הארוכה יותר, וההפרש בין ארכי הרשימות הוא δ, כלומר $k - \ell = \delta$.

   נאתחל מצביע לראש רשימת L_1. נקדמו ב־δ צעדים. כעת, נאתחל מצביע נוסף לתחילת L_2. נריץ את שני המצביעים במקביל ־ על שתי הרשימות ־ עד שייפגשו. אנו טוענים שנקודת המפגש היא למעשה x_1. בכדי לראות זאת, נשים לב שלאחר $\ell + 1$ צעדים בדיוק, המצביע לרשימה השנייה יצביע על x_1, והמצביע לרשימה הראשונה יצביע על האיבר במקום $\delta + \ell + 1 = k + 1$, ולכן יצביע גם הוא על x_1, ולמעשה שני המצביעים ייפגשו.

2. נעבור על כל הערכים האפשריים של (k,ℓ) בצורה "חכמה" שתחסוך לנו זמן ריצה. בכדי לבדוק האם $(k,\ell)=(a,b)$, נריץ מצביע על L_1 בדיוק $a+1$ צעדים, ועל L_2 בדיוק $b+1$ צעדים. אם שניהם מצביעים לאותו המקום - מצאנו את (k,ℓ). שים לב שבכדי לעבור מזוג (a,b) לזוג $(a,b+1)$, אין צורך ב - $a+b+1$ פעולות חדשות, אלא בפעולה בודדת אחת (פשוט מקדמים סוינטר על אחת הרשימות).

נבצע את החיפוש בצורה הבאה:

- אתחל $D=1$.
- כל עוד לא נמצא:
  - $a=D, b=0$.
  - כל עוד $b<D$:
    - $b=b+1$.
    - בדוק האם (a,b) - אם כן שמור נמצא.
  - $a=0$, $b=D$.
  - כל עוד $a<D$:
    - $a=a+1$.
    - בדוק האם (a,b) - אם כן שמור נמצא.
  - $D=D+1$.

זמן הריצה הוא d^2: זאת מכיוון שבלולאה הפנימית אנחנו מבצעים $O(D)$ פעולות, ולמעשה יש לנו את המשוואה הבאה:

$$TIME=\sum_{D=1}^{d} D = 1+2+\ldots+d=\frac{d(d+1)}{2}\in O(d^2)$$

3. בסעיף הקודם עברנו על כל האפשרויות למרחקים, עד שמצאנו ממש את d. הפעם - נעבוד בצורה שונה. במקום להתחיל עם $D=1$ ובכל פעם להגדיל אותו ב-1, נתחיל עם $D=1$ ובכל פעם נכפיל אותו פי שניים. בצורה כזו, נבדוק את ההפרשים הבאים: $1,2,4,8,16,\ldots$. נעצור כאשר נגיע לזוג מצביעים המצביע לאותו מקום ברשימה. נשים לב, שהפעם ייתכן שנפספס את הערך d ונגיע אפילו עד ל$2d$ (כלומר, נמצא התנגשות בין הרשימות במקום במקום d במקום עד ל- $2d$). בכדי למצוא את d במדויק - נחשוב כאילו "קיצצנו" את הרשימה במקום שבו נפגשנו - וחזרנו למעשה לסעיף א'. יש לנו שתי רשימות (קצרות יחסית, באורך $O(d)$) שמתלכדו אל תוך רשימה אחת. נוכל למצוא את נקודת המפגש ב-$O(d)$.

בכדי להבין מדוע החיפוש נקודת המפגש עולה הפעם $O(d)$, נשים לב שאנחנו עוברים, במקרה הגרוע ביותר, על הערכים הבאים של D:

$$1,2,4,8,\ldots,2d \;=\; 1,2^1,2^2,2^3,\ldots,2^{\log 2d}$$

בנוסף, בכל איטרציה אנחנו מבצעים $O(D)$ פעולות. לפיכך,

$$\sum_{i=0}^{\log 2d} 2^i = 1+2+4+\ldots+2d$$

שאלה 2. נתון ביטוי בעל n תווים המכיל שלושה סוגים של סוגריים: (), [], $\{\}$.

1. תארו אלגוריתם הבודק האם הביטוי חוקי או לא.
2. פתרו את התרגיל כאשר קיים סוג רביעי של סוגריים שבו אין הבדל בין סוגר שמאלי לסוגר ימני: | |.

(בביטוי חוקי לכל סוגר ימני יש סוגר שמאלי מתאים וכן אין זוג סוגריים אשר מכיל רק אחד מתוך זוג סוגריים אחר). על האלגוריתמים להיות בסיבוכיות זמן $O(n)$. יש להציג פסאודו קוד והסבר (ברור!) מדוע הוא עובד עבור כל אחד מהסעיפים.

סקיצת פתרון:

1. נשתמש במחסנית. נקרא את הביטוי משמאל לימין, ובכל פעם שנראה סוגר הפותח סוגריים - נכניס אותו למחסנית. בכל פעם שנקרא סוגר הסוגר סוגריים - נוציא את מה שנמצא בראש המחסנית - ונשווה לסוגר שזה עתה קראנו. נבדוק שישנה התאמה ביניהם. אם אין - נחזיר שהביטוי אינו חוקי. כמובן שלאחר קריאת הביטוי, נבדוק שהחמחסנית ריקה וכל הסוגריים השמאליים נסגרו עם ימניים.
2. במקרה זה, לא נוכל לדעת האם הסוגר | הוא פותח סוגריים, או סוגר אותם. נעבוד בצורה דומה לסעיף הקודם. בכל פעם שנראה סוגר פותח - נכניס אותו למחסנית. כאשר נראה סוגר מסוג | - נתבונן בראש המחסנית. אם היא מכילה סוגר | - נניח שהתו שקראנו הוא סוגר ונוציא אותו מהמחסנית. אחרת (המחסנית ריקה או מכילה $\{ ,[,($) - הסוגר הוא סוגר פותח, ונכניס אותו למחסנית.

שאלה 3. נתון מערך שבו k האיברים הראשונים זוגיים, והשאר אי-זוגיים. המטרה היא למצוא את k בזמן:

1. $O(\log n)$.
2. $O(\min\{k, \log n\})$.
3. $O(\log k)$.

הראו אלגוריתם לכל סעיף, ונתחו את זמן הריצה שלו.

סקיצת פתרון:

1. פשוט נבצע חיפוש בינארי. נלך לאמצע המערך. אם האיבר הנמצא באמצע המערך הוא "זוגי", אנו יודעים שk אינו נמצא במקומות $1, \ldots, n/2$. אם האיבר באמצע המערך הוא "אי זוגי", אנו יודעים ש-k אינו נמצא במקומות $n/2 + 1, \ldots, n$. בכל אופן, ע"י בדיקה יחידה, אנחנו יודעים להיפטר מחצי מהאיברים במערך. נמשיך בצורה כזו באופן רקורסיבי, עד שנמצא את k.
2. נשים לב שבחיפוש סידרתי - אנחנו יכולים למצוא את k ע"י k פעולות. נבצע את שני האלגוריתמים במקביל - נריץ חיפוש בינארי (כפי שעשינו בסעיף הקודם), יחד עם חיפוש סידרתי. בכל פעם, נבצע שלב אחד בלבד מכל אלגוריתם. בצורה כזו, נסיים עם האלגוריתם שעובד מהר יותר, ונשלם סה"כ פי שניים מהעלות שלו.

   פתרון נוסף: נבדוק האם $k > \log n$. כלומר, ניגש למערך במקום $\log n$. אם קיבלנו האיבר במקום זה הוא זוגי, נדע כי $k > \log n$, ולכן נריץ חיפוש בינארי. אם k הוא אי זוגי - נדע כי $k < \log n$, ונריץ את האלגוריתם מהסעיף הקודם.
3. נרוץ על האיברים הבאים במערך, ונעצור באיבר הראשון שהוא אי זוגי: $1, 2, 4, 8, \ldots$. נשים לב שבמקרה זה, נעצור באיבר שהוא לכל היותר פי שניים מ-k, כלומר לכל היותר באיבר $2k$. כעת, כדי למצוא את k, נבצע חיפוש בינארי מתחילת המערך ועד לאיבר הראשון שגילינו שהוא אי זוגי. עלות החיפוש הבינארי היא ודאי $O(\log k)$.

   ההסבר מדוע החיפוש של האיבר הראשון שהוא אי זוגי הוא $O(\log k)$ - נפנה ל-1.3 (הניתוח זהה).

שאלה 4. בשני תורים לתשלום בסופרמרקט Q_1 , Q_2 עמדו אנשים. הקופאית בתור Q_1 הפסיקה לעבוד ולכן נאלצו כל האנשים מתור Q_1 לעבור לתור Q_2 על פי הסדר הבא: בכל מקום שני בתור Q_2 יתמזג אדם מ ־ Q_1 וכך הלאה. כתוב פרוצדורה המחזירה את תור Q_2 מעודכן. ניתן להשתמש בפעולות הבסיסיות על תור. אין להשתמש בתור נוסף, כמו כן אין להשתמש במבנה נתונים נוסף.

סקיצת פתרון: נבצע את הפעולות הבאות: ראשית, נשמור בצד את ראש התור $Q2$. כעת, נוציא את האיבר בראש התור ב־$Q2$ ונכניס אותו בחזרה לתור (שימו לב, הוא ייכנס לסוף התור!). כעת, נקח את האיבר בראש התור של $Q1$, נוציא אותו, ונכניסו ל-$Q2$ (שוב, הוא ייכנס לסוף התור, לאחר האיבר שכבר הכנסנו מקודם). בצורה כזו נמשיך עד שנגיע חזרה לאיבר שהיה בראש התור של $Q1$ (ניתן להניח, לשם פשטות, ששני התורים הם מאותו אורך; מה גם שההנחה הזו סבירה מאוד כשמודבר בתור לסופרמרקט...).

שאלה 5. בשיעור התרגול ראינו כיצד ממירים ביטוי מצורת $infix$ לביטוי בצורת $postfix$. הנחנו שהביטוי אינו מכיל סוגריים.

הציגו אלגוריתם הממיר ביטוי מצורת $infix$ לצורת $postfix$ התומך גם בביטויים המכילים סוגריים. מהו זמן הריצה של האלגוריתם שהצגתם?

פתרון: ראה שקף 17 ב:
http://u.cs.biu.ac.il/~gelernn/89120/tirguls/DS_04_LinearDataStructures.pdf