מבני נתונים - תרגול 9
תרגול על עצים

גלעד אשרוב

19 במאי 2014

משפחה של עצים תיקרא מאוזנת אם כל עץ הוא בעל גובה $O(\log n)$, כאשר n הוא מספר הצמתים בעץ. לכל תכונה בהמשך, קבע האם משפחה של עצים המקיימים את התכונה היא מאוזנת. אם התשובה היא "לא", הצג דוגמא נגדית. אם התשובה היא "כן", הצג הוכחה (בד"כ ע"י אינדוקציה). זכור שהתכונה היא אסימפטוטית, ולכן הדוגמא הנגדית צריכה להכיל סדרה אינסופית של עצים במשפחה, ולא רק עץ יחיד.

א. כל צומת בעץ הוא או עלה, או שיש לו שני בנים.

פתרון: התכונה אינה מספיקה. ניתן להסתכל על עץ שרוך ימני, כשלכל צומת בשרוך הוספנו גם בן שמאלי. מספר הרמות הוא בערך $n/2$.

ב. הגודל של על תת עץ יכול להיכתב כ-2^k-1, כאשר k הוא איזשהו מספר שלם (k אינו זהה לכל תת עץ).

פתרון: התשובה היא כן.
נתבונן בתת עץ כלשהו עם שורש r. אנחנו יודעים שגודלו של תת העץ הזה הוא $2^{k_1}-1$, בנוסף גודלו של תת העץ הימני הוא $2^{k_2}-1$, וגודלו של תת העץ השמאלי הוא $2^{k_3}-1$. לכן, נקבל כי:

$$\begin{aligned} 2^{k_1}-1 &= 2^{k_2}-1+2^{k_3}-1+1 \\ 2^{k_1} &= 2^{k_2}+2^{k_3} \end{aligned}$$

נראה שהנ"ל גורר כי $k_2=k_3$. ברור כי $k_1 \geq k_2$ וגם $k_1 \geq k_3$. נניח בלי הגבלת כלליות ש-$k_2 \leq k_3$. נקבל אם כן: $2^{k_1-k_2}=1+2^{k_3-k_2}$ הנ"ל גורר כי: $k_1-k_2=1$, וגם $k_3-k_2=0$. במילים אחרות, $k_2=k_3$, ו- $k_1=k_2+1$. נקבל אם כן כי כל קודקוד מכיל תת עץ ימני בגודל 2^k, ותת עץ שמאלי בגודל 2^k, והעץ מאוזן בצורה מושלמת.

ג. קיים קבוע $0<c<1$ כך ש, לכל צומת בעץ, גודל תת העץ הקטן יותר הוא לפחות c פעמים גודל תת העץ הגדול יותר.

פתרון: כן. ההוכחה היא באינדוקציה. נניח ששני תתי העצים של עץ x מגודל n הם y ו- z עם גדלים n_1 ו- n_2 בהתאמה. לפי הנחת האינדוקציה, הגובה של y הוא לכל היותר $d\log n_1$, והגובה של z הוא לכל היותר $d\log n_2$. נניח בלי הגבלת כלליות ש - $n_1 \geq n_2$. לכן, מתקיים כי $n_2 \geq cn_1$. לכן, אנו מקבלים: $n=n_1+n_2+1 \geq (c+1)n_1$. עומק העץ הוא:

$$d\log n_1+1 \leq d\log\left(n/(c+1)\right)+1 = d\log n - d\log(c+1)+1 \leq d\log n$$

כאשר $d\log(c+1) \geq 1$. לכן, עבור d מספיק גדול, אי השיוויון הנ"ל מתקיים, ונוכל לקבל את הדרוש.

ד. קיים קבוע c כך שלכל צומת של העץ, הגבהים של שני תתי העצים שונים בלכל היותר c.

פתרון: כן. זוהי למעשה הכללה של עצי AVL (שם ראינו כי ההבדל הוא 1).
תהי $n(h)$ פונקציה המקבלת גובה של עץ, ומחזירה את המספר המינימלי של צמתים של עץ המקיים את התכונה שאנו מתבוננים בה. נראה באינדוקציה כי: $n(h) \geq (1+\alpha)^h - 1$, עבור איזשהו קבוע $0 < \alpha \leq 1$. מכאן, נסיק כי: $h \leq \log_{1+\alpha}(n+1) = O(\log n)$.
בסיס - תת עץ בגובה 0 יש צומת יחיד, ומתקיים: $1 \geq (1+\alpha)^0 - 1$ לכל קבוע $\alpha \leq 1$.
צעד - נניח שהטענה נכונה עבור גובה $k < h$. כעת, נתבונן בעץ בגובה h, ונתבונן בשני תת העצים שלו. אנחנו יודעים שגובהו של האחד הוא לכל היותר $h-1$, ולכן של השני הוא לכל הפחות $h-1-c$. לכן, נקבל:

$$n(h) \geq n(h-1) + n(h-1-c) + 1$$

בעזרת טענת האינדוקציה, נקבל:

$$\begin{aligned} n(h) &\geq (1+\alpha)^{h-1} - 1 + (1+\alpha)^{h-1-c} - 1 + 1 \\ &\geq 2(1+\alpha)^{h-1-c} - 1 = 2(1+\alpha)^h \cdot (1+\alpha)^{-(c+1)} - 1. \end{aligned}$$

כעת, נוכל לבחור α כך ש: $(1+\alpha)^{c+1} < 2$, מה שאומר כי: $(1+\alpha)^{-(c+1)} \geq 1/2$. נקבל אם כן:

$$n(h) \geq (1+\alpha)^h - 1$$

ולכן צעד האינדוקציה מתקיים.

ה. הגובה הממוצע של כל צומת בעץ הוא $O(\log n)$.

פתרון: לא. נניח עץ שבו $n - \sqrt{n}$ צמתים הם ממש עץ מאוזן לחלוטין, ושאר $\sqrt{n}$ הצמתים הם מעין שרוך. גובה העץ הוא: $\log(n-\sqrt{n}) + \sqrt{n} = \Omega(\sqrt{n})$, בעוד שממוצע גבהי הצמתים:

$$\begin{aligned} & (1/n) \cdot \left(\sqrt{n} \cdot \left(\sqrt{n} + \log n\right) + \left(n - \sqrt{n}\right) \cdot \log n\right) \\ = \;& (1/n) \cdot (n + n\log n) = O(\log n) \end{aligned}$$

תרגיל. נתונים שני עצי חיפוש. כיצד ניתן למזג אותם לעץ חיפוש אחד?

פתרון: הרעיון הוא לבצע מעבר $infix$, ולקבל שני מערכים ממויינים. לאחר מכן, נבצע מיזוג של שני המערכים למערך אחד ממויין. לבסוף, נבנה עץ חיפוש מאוזן ממערך ממוין בצורה הבאה:

בניית עץ מאוזן למערך (A, i, j). (בהתחלה, $i = 1$, $j = n$)

1. אם $i = j$, נחזיר עלה - $A[i]$. אחרת:
   (א) נסמן $med = (i+j)/2$.
   (ב) ניקח את $A[med]$ להיות השורש -r.
   (ג) נבנה בצורה רקורסיבית עץ מאוזן עבור $(A, i, med-1)$. נסמן את העץ המתקבל כ- T_1.
   (ד) נבנה בצורה רקורסיבית עץ מאוזן עבור $(A, med+1, j)$. נסמן את העץ המתקבל כ- T_2.
   (ה) נבנה את העץ שבו r הוא השורש, הבן השמאלי הוא T_1, הבן הימני הוא T_2.

נקבל כי: $T(n) = 2T(n/2) + 1$, ולכן בסה"כ - $O(n)$.

הוצאה הכנסה מ-AVL. שיחקנו עם עץ AVL בלינק הבא: `http://www.qmatica.com/DataStructures/Trees/AVL/AVLTree.html`. הכנסנו את האיברים הבאים לפי סדר (משמאל לימין):

$$40, 30, 50, 20, 35, 45, 60, 70, 80, 41, 47, 48, 49, 36$$

שימו לב! עליכם לתרגל גם הוצאה.

תרגיל: בנה מבנה נתונים, התומך בפעולות הבאות:

1. הכנסה.
2. הוצאה.
3. חיפוש.
4. מציאת האיבר ה - k בגודלו (כלומר, $k-1$ מהאיברים קטנים ממנו, ושאר $n-k$ האיברים - גדולים ממנו).

נרצה מבנה נתונים התומך בכל הפעולות הנ"ל, בסיבוכיות $O(\log n)$ לכל פעולה.

פתרון: ניקח עץ חיפוש מאוזן (לדוגמא, AVL) ונוסיף שדה $rank$ לכל קודקוד v, המציין את מספר הצמתים בתת העץ המושרש בקודקוד v. כעת, כדי למצוא את האיבר ה -k בגודלו, נבצע את האלגוריתם הבא:

1. נתאחל משתנה rank להיות ה-$rank$ של תת העץ השמאלי.
2. אם $\text{rank} = k-1$ - נחזיר את השורש (יש $k-1$ איברים בתת העץ השמאלי - יש $k-1$ איברים שקטנים מהשורש, השורש הוא האיבר ה-k בגודלו).
3. אם ה-$\text{rank} > k-1$, אזי האיבר ה־k בגודלו נמצא בתת העץ השמאלי. נכנס רקורסיבית לתת עץ זה עם חיפוש איבר k.
4. אם ה-$\text{rank} < k-1$ של הבן השמאלי קטן מ-$k-1$, אזי האיבר ה־k בגודלו נמצא בתת העץ ימני. נכנס רקורסיבית לתת עץ הימני זה עם חיפוש איבר $k - \text{rank} - 1$.