מבני נתונים
פתרון תרגיל 7

גלעד אשרוב

9 במאי 2013

תרגיל 1. נתונים 3 עצים בינאריים T_1, T_2, T_3 וידוע שכל אחד מהם מכיל את n המספרים: $\{1, \ldots, n\}$. עוברים על T_1 ב-$inorder$, על T_2 ב-$preorder$ ועל T_3 ב-$postorder$ ומקבלים כתוצאה את אותה סדרה $1, \ldots, n$ בשלושת המעברים, כלומר המספרים מודפסים בסדר עולה. הוכח/י או הפרכ/י את הטענות הבאות:

1. T_1 הוא בהכרח עץ חיפוש בינארי.
2. T_2 הוא בהכרח עץ חיפוש בינארי.
3. אם נעבור על T_2 ב-$postorder$ ועל T_3 ב-$preorder$ נקבל את אותה הסדרה.

פתרון:

1. T_1 הוא עץ חיפוש בינארי. הטענה שעלינו להוכיח היא שאם ריצת $inorder$ על עץ עם מפתחות $1, \ldots, n$ מובילה לסידרה $1, \ldots, n$, אזי T הוא עץ חיפוש בינארי. מספיק להראות שאם T הוא לא עץ חיפוש בינארי, אזי ריצת $inorder$ לא יוצאת ממויינת.

   יהי T עץ בינארי כלשהו שהוא לא עץ חיפוש. כלומר, בהכרח מתקיימים אחד משני המצבים הבאים: קיים איבר x בתת העץ השמאלי של השורש עם מפתח גדול ממפתח השורש r, או שקיים איבר y בתת העץ הימני של השורש עם מפתח קטן מהשורש r. במקרה הראשון, בריצת $inorder$ נקבל בסידרה את x לפני r (ולכן הסדרה אינה ממויינת), ובמקרה השני בריצת $inorder$ נקבל את r לפני y. בשני המקרים, הסידרה המתקבלת אינה ממויינת בסדר עולה.

2. ודאי שלא. נתבונן במקרה הפשוט של $\{1, 2\}$, כאשר 2 הוא הבן השמאלי של 1. זהו אינו עץ חיפוש, אבל ריצת ה-$preorder$ יוצאת ממויינת.

3. ודאי שלא. מספיק להתבונן בשורש - השורש יודפס בריצת $postorder$ כאיבר האחרון בסדרה, ובריצת $preorder$ כאיבר הראשון בסידרה.

תרגיל 2. עץ בינומי מסדר i מוגדר בצורה הבאה: עץ בינומי מסדר 0 הוא קודקוד בודד. עץ בינומי מסדר i מורכב מ-2 עצים בינומיים מסדר $i-1$, A, B, כאשר השורש של A הוא בנו של השורש של B (שימו לב העץ אינו בינארי).

1. צייר עץ בינומי מסדר 3.
2. מה מספר הקודקודים בעץ בינומי מסדר n? הוכח/י.
3. מה עומק של עץ בינומי שיש בו M קודקודים? הוכח/י את טענתך.

פתרון:

1. השלם לבד.

2. נסמן ב- B_i את העץ הבינומי מסדר i, וב-$T(B_i)$ את מספר הקודקודים בעץ בינומי B_i. נקבל אם כן: $T(B_0) = 1$. $T(B_i) = 2T(B_{i-1})$. ולכן, $T(B_i) = 2^i$.

3. נראה שעץ בינומי מסדר i יהיה i רמות (כאשר מתחילים לספור מ-0). ההוכחה לטענה זו היא באינדוקציה (נראה בהמשך). בהינתן שמספר הרמות לעץ בינומי מסדר i היא i, ובהינתן שעץ בינומי מסדר i יש 2^i קודקודים, נקבל אם כן כי בעץ בינומי מסדר $\log M$ יש M קודקודים, ולכן גובה עץ עם M קודקודים.

   כעת, נראה את הטענה עבור עץ בינומי מסדר i יש i רמות:

   בסיס - עץ בינומי מסדר 0 הוא קודקוד בודד, ומספר הרמות הוא 0-.

   נניח שהטענה מתקיימת עבור עץ בינומי מסדר k - מספר הרמות הוא k.

   עץ בינומי מסדר $k+1$ הוא עץ בינומי מסדר k, כשהוספנו לו כבן עץ בינומי מסדר k. כלומר, כמו בהגדרה - השורש של A הוא בנו של השורש של B - ולפיכך, מספר הרמות בעץ שהתקבל הוא מספר הרמות ב-A + 1. לפי הנחת האינדוקציה, ב-A יש k רמות, ולפיכך בעץ שקיבלנו יש בסה"כ $k+1$ רמות, כפי שרצינו.

תרגיל 3. יהי BT מרחב כל העצים הבינאריים האפשריים, נסמן ב-$\langle\rangle$ את העץ הריק, ובהינתן שני עצים L, R, נסמן ב-$\langle L, x, R\rangle$ את העץ ששורשו הוא x, תת העץ השמאלי שלו הוא L ותת העץ הימני הוא R. נגדיר את הפונקציה $f : BT \to \mathbb{N}$ בצורה הבאה:

$$\begin{aligned} f\left(\langle\rangle\right) &= 0 \\ f\left(\langle L, x, R\rangle\right) &= \begin{cases} 1 & \text{if} L = R = \langle\rangle \\ f(L) + f(R) & \text{otherwise} \end{cases} \end{aligned}$$

1. תאר/י בצורה פשוטה מה הפונקציה מחשבת (כלומר, ללא רקורסיה), והוכח/י את תשובתך בעזרת אינדוקציה על מבנה העץ.

2. נניח שנשנה את הגדרת f כך ש- $f(\langle L, x, R\rangle) = f(L) + f(R) + 1$ במקום הסכום שרשום לעיל. מה הפונקציה מחשבת במקרה זה? הוכח/י את תשובתך.

פתרון:

1. הפונקציה מחשבת את מספר העלים בעץ. נוכיח באינדוקציה על המבנה, כאינדוקציה על מספר הרמות.

   אם העץ ריק - מספר העלים הוא 0. אם העץ מורכב משורש בלבד $\langle\langle\rangle, x, \langle\rangle\rangle$: מספר העלים הוא 1.

   כעת, נניח שעבור עץ עד k רמות, מספר העלים הוא $f(T)$. נתבונן על עץ עם $k+1$ רמות: $\langle L, x, R\rangle$. מספר הרמות בעצים L, R הוא פחות מ-k. עבור עץ כזה, $f(L)$ הוא מספר העלים בעץ L, ו-$f(R)$ הוא מספר העלים בעץ R (נשים לב שאם L או R ריקים, מספר העלים הוא 0). מספר העלים הכולל בעץ $\langle L, x, R\rangle$ הוא באמת מספר העלים בשני העצים L, R, כלומר $f(L) + f(R)$.

2. הפוקנציה מחשבת את מספר הקודקודים בעץ. ההוכחה היא שוב - באינדוקציה (השלם לבד).

תרגיל 4. כתבו פסאדו קוד לאלגוריתם המקבל כקלט עץ חיפוש בינארי וזוג צמתים בעץ k_1, k_2, ומוצא את האב הקדמון הקרוב ביותר (ניתן להניח כל צומת מחזיק גם מצביע לאביו).

פתרון: נראה פסאדו קוד:

Algorithm 1 $Find(T, k1, k2)$

```
if T = NULL then
  return NULL
end if
if T.value = k1 or T.value = k2 then
  return T.parent
end if
if k1 < T.value and k2 < T.value then
  return Find(T.left, k1, k2)
end if
if k1 > T.value and k2 > T.value then
  return Find(T.right, k1, k2)
end if
return T
```

תרגיל 5. כתבו אלגוריתם המקבל כקלט מספר n, ומחזיר כפלט את מספר עצי החיפוש הבינאריים האפשריים בעלי מפתחות $\{1, \ldots, n\}$. לדוגמא, עבור $n = 1$, נקבל עץ בודד. עבור $n = 2$ נקבל שני עצים אפשריים. עבור $n = 3$, נקבל כבר שישה עצים אפשריים, וכן הלאה. עליכם לכתוב פסאדו קוד לפונקציה. שימו לב - אתם לא אמורים ליצור את העצים; רק להחזיר את מספר העצים.

פתרון: נניח אנחנו יודעים לפתור את הבעיה לכל עץ חיפוש מסדר $k < n$. בכדי לפתור לסדר n, נשים כל איבר אפשרי כשורש, ונחלק (עפ"י תכונות עץ חיפוש) לתת עץ שמאלי ותת עץ ימני. נקבל את האלגוריתם הבא:

Algorithm 2 *int countTrees(int numKeys)*

```
if numKeys <= 1 then
  return 1
end if
sum = 0
{//Iterate through all the values that could be the root...}
for root = 1 to root <= numKeys do
  left = countTrees(root − 1)
  right = countTrees(numKeys − root)
  sum += left ∗ right
end for
return sum
```