<u>מבני נתונים – פתרון 6 – עץ חיפוש בינארי</u>

1. לכל אחת מהבעיות הבאות הציעו אלגוריתם יעיל ביותר (מבחינת זמן הריצה) ונתחו את זמן הריצה שלו:

א. קלט: עצי חיפוש T_1, T_2
פלט: "כן", אם המפתחות המיוצגים בעצים T_1, T_2 הם אותם מפתחות, אחרת, "לא".

<u>פתרון-</u>

i. בצע סריקת inorder על עץ T_1 לתוך מערך A_1
ii. בצע סריקת inorder על עץ T_2 לתוך מערך 2_1
iii. בדוק שהמערכים A_1, A_2 זהים

זמן ריצה: $\Theta(n)$

הוכחה: סריקת inorder על עץ חיפוש בינארי נותנת רשימה ממוינת של מפתחות העץ. לכן, אם המפתחות שבשני העצים הם אותם מפתחות, אז הרשימות הממוינות שלהם תהיינה זהות, ואחרת לא. לכן האלגוריתם יחזיר תשובה נכונה.

ב. קלט: עצי חיפוש T_1, T_2
פלט: "כן", אם העצים T_1, T_2 זהים, אחרת, "לא".
שים לב: נאמר ששני עצים זהים, אם הם מיוצגים על ידי <u>אותו מבנה של עץ</u> וכן עם אותם מפתחות.

<u>פתרון-</u>

נגדיר: x1 מצביע לשורש העץ T1 ו- x2 מצביע לשורש העץ T2

```
1.  identical(x1,x2)
2.      if x1 ≠ null AND x2 ≠ null then
3.          return(identical(left(x1),left(x2)) AND (key(x1) = key(x2)) AND
            identical(right(x1),right(x2));
4.      else if x1 = null AND x2 = null then return(true);
5.          else return(false);
```

<u>הסבר והוכחה</u> [בהתייחס למספרי השורות]:

שורה 2-3: אם הצמתים קיימים – נבדוק שהמפתחות שלהם שווים וכן נבדוק רקורסיבית שהתתי עץ של שניהם זהים. [אם המפתחות שונים אז ייכנס ערך false לתוך הביטוי של ה-and ולכן יוחזר false.]

שורה 4: אם שני הצמתים הם null – אזי הם שווים וזהו תנאי העצירה ולכן נחזיר true.

שורה 5: אחרת – בעץ אחד הצומת קיים ובשני לא –> מבנה העצים אינו זהה ולכן נחזיר false.

לסיכום – עוברים על שני העצים באופן רקורסיבי וזה אפשרי רק אם המבנה של שני העצים זהה.

אפשרות נוספת לפתרון – כמו סעיף א, אבל עם סריקת postorder או preorder.

הוכחה:

ניתן לשחזר עץ בינארי אם נתונות **שתי** סריקות שונות שלו [שתיים מתוך inorder, postorder או preorder].

מאחר ונתון שהעצים הם עצי חיפוש בינאריים, הרי שידוע שסריקת inorder עליהם תתן רשימות ממוינות של המפתחות. לכן מספיקה עוד סריקה אחת שתתן רשימות זהות של מפתחות שני העצים, כדי לוודא שהמבנה של העצים אכן זהה.

[אם יש מפתחות שונים – ברור שהשוואה של הרשימות תזהה אותם, ואם המפתחות שווים אך המבנה של העצים שונה – סדר המפתחות בשתי הרשימות יהיה שונה.]

זמן הריצה [בשתי האפשרויות לפתרון]: $\Theta(n)$ כי עוברים על כל האיברים בעצים מספר פעמים קבוע.

שאלה 2 נתון עץ חיפוש בינארי המכיל מספרים בין 1 ל- 1000 (לאו דווקא את כולם). אנו מעוניינים לחפש את המספר 363.
האם הסדרה הבאה יכולה להיות סדרת המספרים בהם ביקרנו במהלך החיפוש? (משמאל לימין):
942, 258, 344, 621, 311, 383, 350, 363.

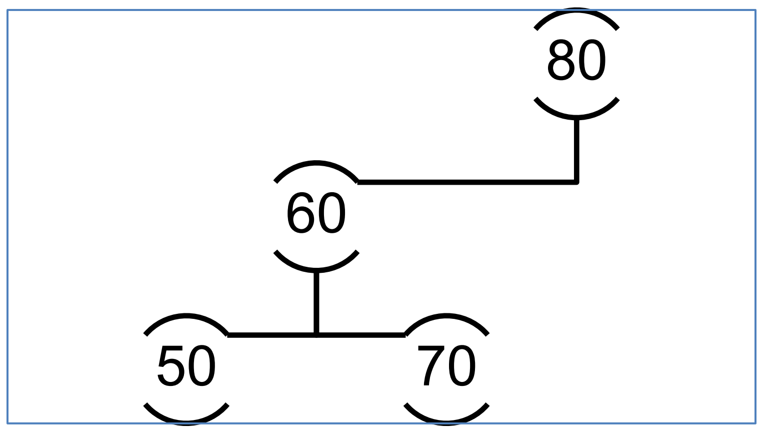
תשובה:
הסדרה הנתונה לא יכולה להיות סדרת המספרים בהם מבקרים במהלך החיפוש.
אלגוריתם החיפוש בעץ חיפוש בינארי פועל כך: אם הערך אותו אנו מחפשים קטן מהצומת הנוכחי, ממשיכים לבן השמאלי שלו, ואם הוא גדול ממשיכים לבן הימני. כמו כן, נזכור שכדי שעץ יהיה עץ חיפוש בינארי צריכים להתקיים התנאים הבאים: כל המפתחות בתת העץ השמאלי של השורש קודמים למפתח של השורש, המפתח של השורש קודם לכל המפתחות בתת העץ הימני שלו, ותתי העציים השמאלי והימני של השורש הם עצי חיפוש בינארי.

1. 942 - שורש העץ. 363 קטן ממנו ולכן עלינו להמשיך לבן השמאלי, כלומר כל המפתחות הבאים צריכים להיות קטנים מ- 942.
2. 258 – תקין כי 942>258. 363 גדול ממנו ולכן עלינו להמשיך לבן הימני, כלומר כל המפתחות הבאים צריכים להיות בין 258 ל- 942.
3. 344 – תקין כי 258 < 344 ו- 942 > 344. 363 גדול ממנו ולכן עלינו להמשיך לבן הימני, כלומר כל המפתחות הבאים צריכים להיות בין 344 ל- 942.
4. 621 – תקין כי 258<344<621 ו- 942>621. 363 קטן ממנו ולכן עלינו להמשיך לבן השמאלי, כלומר כל המפתחות הבאים צריכים להיות בין 344 ל- 621.
5. 311– לא תקין כי 344 >311, לא מקיים את התנאי השני שהמפתח של השורש (344) יהיה קודם לכל המפתחות בתת העץ הימני שלו.

<u>שאלה 3</u> פרופ' בינארוס סבור שגילה תכונה מעניינת ביותר של עצי חיפוש בינאריים. נניח שחיפוש מפתח k בעץ חיפוש בינארי מסתיים בעלה. נתבונן בשלוש קבוצות: A – המפתחות משמאל למסלול החיפוש. B – המפתחות במסלול החיפוש. C – המפתחות מימין למסלול החיפוש. פרופ' בינארוס טוען שכל שלושה מפתחות $a \in A,\ \ b \in B,\ \ c \in C$ מקיימים בהכרח $a \le b \le c$.

הביאו דוגמא נגדית לטענתו של הפרופסור, של עץ החיפוש **הקטן ביותר האפשרי**.

<u>תשובה</u>:
דוגמא נגדית לטענת הפרופ׳: k=50, ונתון עץ החיפוש הבינארי הבא:



לפי ההגדרות של הפרופ׳, אלו האיברים בכל קבוצה:
A = { } (קבוצה ריקה)
B = {80, 60, 50}
C = {70}

ניקח c=70 ו- b=80 וכמובן שלא מתקיים b<=c .

(**<u>הערה</u>** – שימו לב שהתבקשתם למצוא דוגמא על העץ הבינארי **הקטן ביותר**.
לכן תשובה נכונה כוללת עץ בעל 4 צמתים בלבד!)