

חזרה בנושא עץ

קוטר של עץ בינרי

מטרות

פתרון בעיה תוך שימוש בפעולות עזר על עץ בינרי.

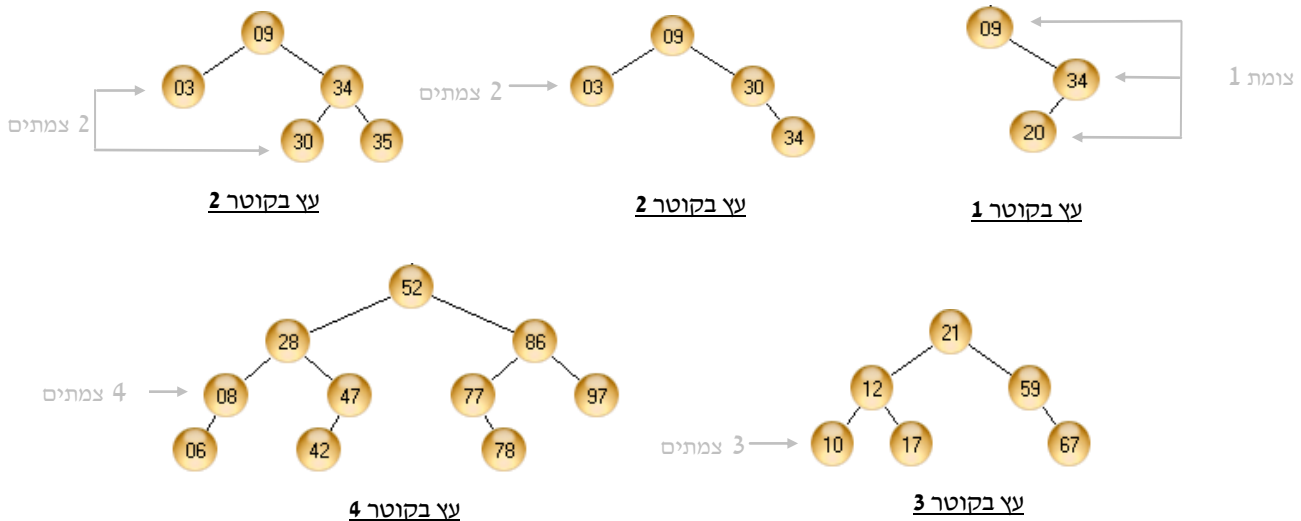
רמת השאלה

בינונית - קשה. מתאים גם כשאלת בגרות.

נוסח השאלה

"קוטר" של עץ בינרי מוגדר כך: מספר הצמתים הגדול ביותר ברמה כל שהיא בעץ.

לדוגמה, לפניך 5 עצים ועבור כל אחד מצוין קוטרו:



מה עליכם לעשות?

א. ציירו עץ בינרי בגובה 4 שקוטרו 4 ובו מינימום צמתים.

ב. מהו הקוטר הגדול ביותר שיכול להיות לעץ בינרי בגובה h? נמקו!

ג. ממשו את פעולה הבאה:

```
public static int diameter(BinTree<Integer> bt)
```

הפעולה מקבלת עץ בינרי bt ומחזירה את קוטרו.

הנחיות מיוחדות

- שאלה זו בודקת את היכולת של התלמיד לפרק את הבעיה לתת-בעיות תוך שימוש בפעולות עזר. שימו לב שלא נדרש לממש את הפעולה diameter בצורה רקורסיבית. פתרון רקורסיבי, במקרה זה, קשה מאוד. ניתן ורצוי לממש את הפעולה בצורה איטרטיבית.
- נראה כי סריקה לפי רמות יכולה לעזור לנו כאן, אבל לא כך הדבר. אלגוריתם סריקת העץ לפי רמות, כפי שהוא מופיע בספר, אינו מאפשר לנו לזהות מתי מתחילה או מסתיימת רמה, לכן לא ניתן לחשב את מספר הצמתים ברמה מסוימת.
- שימו לב לעובדה שהעץ המתקבל הוא עץ קונקרטי של מספרים שלמים, למרות שהפעולה diameter לא מחייבת זאת וניתן לחשב קוטר של עץ בינרי כלשהו. במסגרת יחידה זו אנו חייבים לציין את טיפוס איברי העץ בכל פעולה חיצונית.

רמז לפתרון

יש להשתמש בשתי פעולות עזר: גובה-עץ ומספר-צמתים-ברמה-מסוימת. הרעיון הוא לעבור על כל הרמות בעץ, מהשניה ועד האחרונה, לספור את מספר הצמתים ולשמור את המספר המקסימלי.

פתרון

ב. הקוטר הגדול ביותר שיכול להיות לעץ בינרי בגובה h הוא 2^h כאשר העץ מלא. כי ברמה הגבוהה ביותר בעץ זה יש את מספר הצמתים הגדול ביותר.
ג. הפעולה:

```
public static int diameter(BinTree<Integer> bt)
{
    int treeDiameter = 1; // הנחה כי קוטר העץ הוא 1
    int h = height(bt); // חישוב גובה העץ

    // נעבור על כל רמה בעץ, נחשב את מספר הצמתים בה ונשמור את המקסימום
    for(int level=1; level<=h; level++)
    {
        int n = numNodesInLevel(bt, level); // חישוב מספר הצמתים ברמה
        if(n>treeDiameter)
            treeDiameter = n;
    }

    return treeDiameter;
}
```

שימו לב שהפעולה משתמשת בשתי פעולות עזר `height` ו-`numNodesOnLevel` לחישוב גובה העץ ומספר הצמתים ברמה נתונה בעץ. את שתי הפעולות צריך לממש:

```
public static int numNodesInLevel (BinTree<Integer> tree, int level)
{
    int nl = 0;
    int nr = 0;

    if (level==0)
        return (1);
    if (tree.getLeft () != null)
        nl = numNodesInLevel (tree.getLeft (), level-1);
    if (tree.getRight () != null)
        nr = numNodesInLevel (tree.getRight (), level-1);

    return nl+nr;
}

public static int height (BinTree<Integer> tree)
{
    int hl = 0;
    int hr = 0;

    if (tree.getLeft () != null)
        hl = 1 + height (tree.getLeft ());
    if (tree.getRight () != null)
        hr = 1 + height (tree.getRight ());

    return Math.max (hl, hr);
}
```

מקור השאלה

אילן פרץ, מכללת אורט גבעת רם, ירושלים.