

ממש פעולה חיצונית ב־ Java או Exist ב־ C#.

הפעולה תקבל עץ בינרי t מטיפוס שלם ומספר שלם x. הפעולה תחזיר true אם יש בעץ צומת שערכו x, אחרת – הפעולה תחזיר false. אם העץ ריק – הפעולה תחזיר false. לפניך הפעולה ב־ Java check(t1, t2) וב־ C# Check(t1, t2).

ב.

הפעולה מקבלת שני עצים בינריים לא ריקים מטיפוס שלם, t1 ו־ t2, ומחזירה רשימה המכילה את כל המספרים הנמצאים בעץ t1 ואינם נמצאים בעץ t2. הפעולה מזמנת פעולה נוספת המקבלת שלושה פרמטרים.

Java

```
public static Node<Integer> check(BinNode<Integer> t1, BinNode<Integer> t2)
{
    Node<Integer> first = new Node<Integer> (- 1);
    first = check(t1 , t2 , first);
    return first.getNext();
}
```

C#

```
public static Node<int> Check(BinNode<int> t1 , BinNode<int> t2)
{
    Node<int> first = new Node<int> (- 1);
    first = Check(t1 , t2 , first);
    return first.GetNext();
}
```

ממש את הפעולה:

ב־ Java:

```
public static Node<Integer> check( BinNode<Integer> t1,
                                   BinNode<Integer> t2 , Node<Integer> list)
```

או ב־ C#:

```
public static Node<int> Check(BinNode<int> t1 , BinNode<int> t2 , Node<int> list)
```

אתה יכול להשתמש בפעולה שמימשת בסעיף א.

ג.

מה היא סיבוכיות זמו הריצה של הפעולה שמימשת בסעיף ב ? נמק את תשובתך.