Java

```
public  static boolean isUpZigZag (BinNode<Integer> t)
{ // הפעולה מקבלת עץ בינארי ומחזירה אמת אם הוא עץ זיג-זג-בגודל-עולה ושקר אחרת
      if  (t == null || isLeaf(t))
            return true;
      BinNode<Integer>p = t;
      int m = 1, n = 1;
      int side = 1;              // 1 - right, 2 - left
      while (!isLeaf(p))
      {
        if (side == 1)
        {
            if  (p.getLeft() == null && p.getRight() != null)
            {
                  n++;
                  p = p.getRight();
            }
            else
                  return false;
         }
         else if (side == 2)
         {
            if  (p.getRight() == null && p.getLeft() != null)
            {
                  n++;
                  p = p.getLeft();
              }
            else
                   return false;
            }
            if (m < n)
            {
                  m++;
                  n = 1;
                  if (side == 1)
                        side = 2;
                  else
                        side = 1;
            }
      }
      if (n == 1)
            return true;
      return false;
}
public  static boolean  isLeaf (BinNode<Integer>t)
{ // הפעולה מקבלת שורש לעץ בינארי ומחזירה אמת אם הוא עץ עלה ושקר אחרת
      if  (t == null)
            return false;
      return (t.getLeft() == null && t.getRight() == null);
}
```

<u>הסבר</u>: המונה m מתאר את אורך רצף הצמתים שעבור כולם מתקיים שאין להם תת-עץ בכיוון אחד ויש להם תת-עץ בכיוון שני. כדי לדעת את הכיוון יש שימוש במשתנה side שיכול להכיל את הערכים ימין / שמאל, והוא מנחה את הבדיקה. כאשר מסתיים רצף באורך נדרש מתחלף side, וכך הרצף הבא ייבדק על פי הכיוון ההפוך של הרצף.
הבדיקה תסתיים כאשר במהלך התהליך לא התקיים התנאי ואז מוחזר ערך שקר, או כאשר מגיעים לעלה. במצב כזה יש לבדוק האם הרצף האחרון הסתיים כהלכה. הבדיקה נעשית באופן הבא: – אם האלגוריתם מוכן לתחילת רצף חדש אזי מתקיים התנאי (n = 1), ומוחזר ערך אמת, אחרת הסיום הוא באמצע רצף ואינו תקין ולכן מוחזר שקר.