CS 284: Homework Assignment 3

1 Assignment Policies

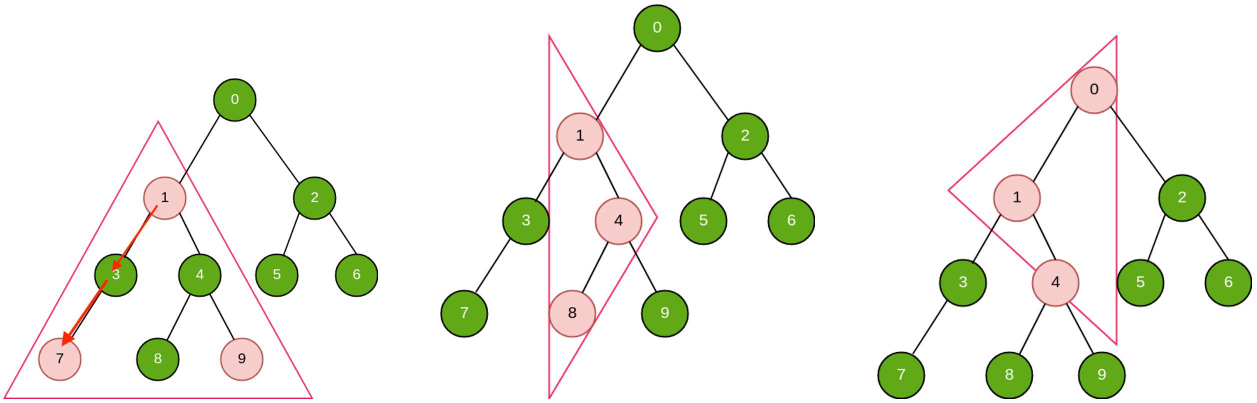
Collaboration Policy. Homework will be done individually: each student must hand in their own answers. It is acceptable for students to collaborate in understanding the material but not in solving the problems or programming. Use of the Internet is allowed, but should not include searching for existing solutions.

Under absolutely no circumstances code can be exchanged between students. Excerpts of code presented in class can be used.

Your code must include a comment with your name and section .

2 Assignment

[Counting Type-2 and Type-3 Iso Triangles]. There are 3 types of Iso triangles:



(a) Type-1 Iso Triangle

(b) Type-2 Iso Triangle

(c) Type-3 Iso Triangle

In class, we learned how to count the number of Type-1 iso triangles in a binary search tree. In this problem, you will need to implement a method:

```
Pair<Integer> count_type2_iso_triangle(Node<Integer> root, ...){}; //... stands for any potential paramete
```

which counts the total number of Type-2 and Type-3 triangles in a binary tree. Create a test method `test_count_iso_triangle`, show your implementation is correct with a few test cases you come up with. The total number of Type-2 and Type-3 triangles should be stored in a global variable named `total_iso_triangle` at the end of `count_iso_triangle`.

Notice you are not allowed to add pointers from the children to the parent, or change the structure of the Node class in IsoTriangle.java.

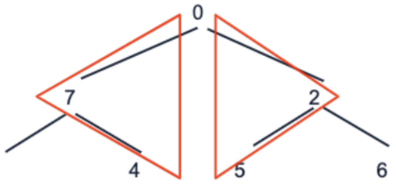
Hint. In class we learned that the number of Type-1 iso triangles whose root is node n is:

$$count(n\ as\ root) = min(n.left\_path\_len, n.right\_path\_len)$$

We can derive a similar formula for Type-2 and Type-3 iso triangles. When implementing `count_iso_triangle2`, think about the 4-step process we learned in class: How to update the corresponding path lens in a recursive algorithm? What information should the parent pass to children? What information should the child return to its parent? **Write down your 4-step process in the comment before your method, like this**:

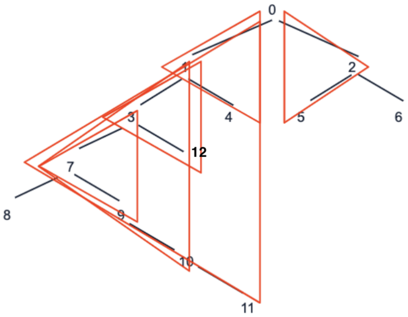
```java
/**
 * 4-step process:
 * (1) What info to pass to children?
 * (2) What info to return to parent?
 * (3) How to handle terminal cases?
 * (4) How to update the solution?
 * count(n as root) = ... (write down your formula)
 * @param ...
 * @return ...
 */
public Pair<Integer> count_type2_iso_triangle(Node<Integer> root, ...)
```

3 Example



(a) Test case 1: `total_iso_triangle = 2`

Figure 1: Test case 1: `total_iso_triangle=2`



(b) Test case 2: `total_iso_triangle = 6`

Figure 2: Test case 2: `total_iso_triangle=6`

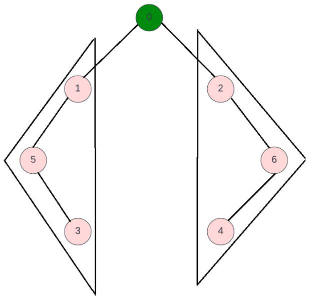


Figure 3: Test case 3: `total_iso_triangle=2`

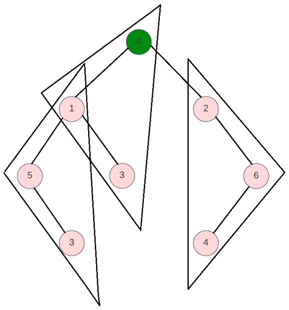


Figure 4: Test case 4: `total_iso_triangle=3`

4 Submission instructions

Submit `IsoTriangle.java` to GradeScope. Your raw GS score is decided by the test cases your code passes. Your final score depends on your raw GS score, the late penalty, and the result of our plagiarism detection.