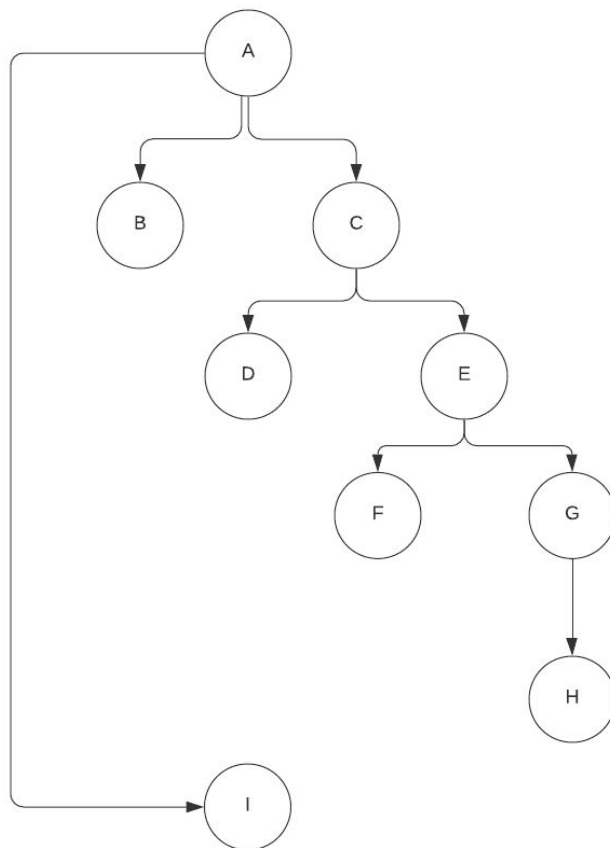


**Node 1:** if (side1 + side2 <= side3 || side1 + side3 <= side2 || side2 + side3 <= side1)  
**Node 2:** return "it's not a triangle";  
**Node 3:** if ((side1 == side2 && side2 == side3) || (side1 == side3 && side3 == side2) || (side2 == side3 && side3 == side1))  
**Node 4:** return "Equilateral Triangle";  
**Node 5:** else if (side1 != side2 && side2 != side3 && side3 != side1)  
**Node 6:** return "Scalene Triangle";  
**Node 7:** else if ((side1 == side2 && side2 != side3) || (side1 == side3 && side3 != side2) || (side2 == side3 && side3 != side1))  
**Node 8:** return "Isoceles Triangle";  
**Node 9:** return "error";

```

7  TriangleClass
8
9  9 references
10 public static class Triangle
11 {
12     9 references
13     public static string Analyze(int side1, int side2, int side3)
14     {
15         1  if (side1 + side2 <= side3 || side1 + side3 <= side2 || side2 + side3 <= side1)
16         {
17             2  return "it's not a triangle";
18         }
19         else
20         {
21             3  if ((side1 == side2 && side2 == side3) || (side1 == side3 && side3 == side2) || (side2 == side3 && side3 == side1))
22             {
23                 4  return "Equilateral Triangle";
24             }
25             else if (side1 != side2 && side2 != side3 && side3 != side1)
26             {
27                 5  return "Scalene Triangle";
28             }
29             else if ((side1 == side2 && side2 != side3) || (side1 == side3 && side3 != side2) || (side2 == side3 && side3 != side1))
30             {
31                 6  return "Isoceles Triangle";
32             }
33             7  return "error";
34         }
35     }
36 }
37

```



### **Cyclomatic Complexity**

$$M = E - N + 2P$$

$$M = 8 - 9 + 2(5) = 9$$

There are 9 Nodes (N) followed by 8 edges(lines) (E) and 5 exit nodes represented by (P).