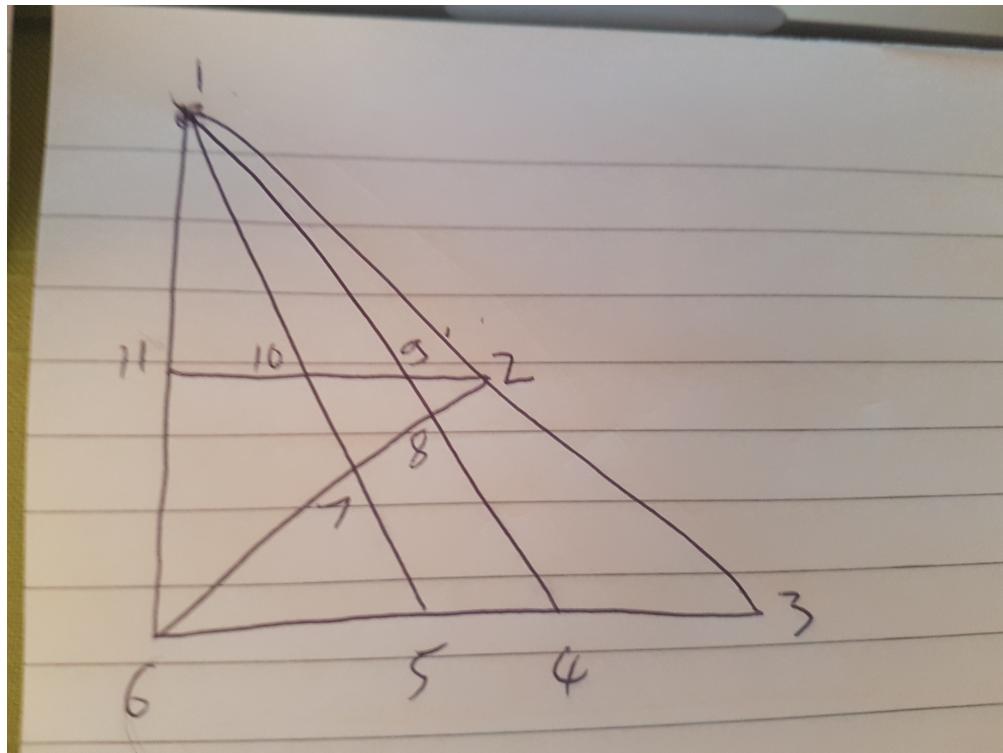


Q: How many triangles are in the picture?



The following is an attempt for a general solution for any figure with straight lines.

To solve, we need to know which nodes are connected and which nodes (if any) have to be traversed for each connection.

This grid layout has all nodes in the index/ columns. The traversed nodes were filled into the grid (0 == immediate connection, NaN == no connection).

	1	2	3	4	5	6	7	8	9	10	11
1	NaN	(0,)	(2,)	(9, 8)	(10, 7)	(11,)	(10,)	(9,)	(0,)	(0,)	(0,)
2	(0,)	NaN	(0,)	NaN	NaN	(7, 8)	(8,)	(0,)	(0,)	(9,)	(10, 9)
3	(2,)	(0,)	NaN	(0,)	(4,)	(5, 4)	NaN	NaN	NaN	NaN	NaN
4	(9, 8)	NaN	(0,)	NaN	(0,)	(5,)	NaN	(0,)	(8,)	NaN	NaN
5	(10, 7)	NaN	(4,)	(0,)	NaN	(0,)	(0,)	NaN	NaN	(7,)	NaN
6	(11,)	(8, 7)	(4, 5)	(5,)	(0,)	NaN	(0,)	(7,)	NaN	NaN	(0,)
7	(10,)	(8,)	NaN	NaN	(0,)	(0,)	NaN	(0,)	NaN	(0,)	NaN
8	(9,)	(0,)	NaN	(0,)	NaN	(7,)	(0,)	NaN	(0,)	NaN	NaN
9	(0,)	(0,)	NaN	(8,)	NaN	NaN	NaN	(0,)	NaN	(0,)	(10,)
10	(0,)	(9,)	NaN	NaN	(7,)	NaN	(0,)	NaN	(0,)	NaN	(0,)
11	(0,)	(10, 9)	NaN	NaN	NaN	(0,)	NaN	NaN	(10,)	(0,)	NaN

The algorithm starts at the first node and looks for connections. It then tests each of those connections for a potential triangle, by investigating the second node's connections and so on...

A triangle is found, IFF there are three unique nodes which are traversed exactly once.

In [4]:

```
import pandas as pd
import itertools
```

In [5]:

```
df = pd.read_msgpack('triangles')
```

In [3]:

```

def find_used_nodes(df, node1, node2):
    result = [node1, node2]
    for i in df[node1][node2]:
        if i != '0': result.append(i)
    return list(set(result))

triangles = []
used_nodes = []
for first_node in df.index:
    for second_node in df.index:
        waypoint_null = df.isnull()[second_node][first_node]
        if waypoint_null: continue
        # Second node now technically possible

        for third_node in df.index:
            second_waypoint_null = df.isnull()[second_node][third_node]
            third_waypoint_null = df.isnull()[first_node][third_node]
            if second_waypoint_null or third_waypoint_null: continue

            used_nodes = find_used_nodes(df, first_node, second_node)
            node_used = third_node in used_nodes

            second_waypoints_used = any([x in used_nodes for x in df[second_node][third_node]])
            used_nodes.append(find_used_nodes(df, third_node, second_node))
            third_waypoints_used = any([x in used_nodes for x in df[first_node][third_node]])

            if any((node_used, second_waypoints_used, third_waypoints_used)): continue
            triangles.append(sorted([first_node, second_node, third_node]))

triangles.sort()
triangles = list(k for k,_ in itertools.groupby(triangles))
print('I got {} triangles, correct?'.format(len(triangles)))
triangles

```

I got 24 triangles, correct?

Out[3]:

```

[['1', '10', '11'],
 ['1', '10', '2'],
 ['1', '10', '9'],
 ['1', '11', '2'],
 ['1', '11', '9'],
 ['1', '2', '6'],
 ['1', '2', '7'],
 ['1', '2', '8'],
 ['1', '2', '9'],
 ['1', '3', '4'],
 ['1', '3', '5'],
 ['1', '3', '6'],
 ['1', '4', '5'],
 ['1', '4', '6'],
 ['1', '5', '6'],
 ['1', '6', '7'],
 ['1', '6', '8'],
 ['1', '7', '8'],
 ['10', '2', '7'],
 ['11', '2', '6'],
 ['2', '3', '6'],
 ['2', '8', '9'],
 ['3', '4', '5'],
 ['3', '4', '6'],
 ['3', '5', '6'],
 ['3', '6', '7'],
 ['3', '7', '8'],
 ['4', '5', '6'],
 ['4', '5', '7'],
 ['4', '6', '7'],
 ['4', '7', '8'],
 ['5', '6', '7'],
 ['5', '6', '8'],
 ['5', '7', '8'],
 ['6', '7', '8'],
 ['6', '7', '9'],
 ['7', '8', '9'],
 ['7', '9', '10'],
 ['8', '9', '10'],
 ['9', '10', '11'],
 ['10', '11', '11']]

```

```
[ '4', '6', '8'],  
[ '5', '6', '7']]
```