```
def naive_triangle_counting():
        triangles = 0
        for u in range(n):
                for v in adj_list[u]:
                        triangles = triangles + len(set(adj_list[u]).intersection(set(adj_list[v])))
        # print(int(triangles/6))



def improved_triangle_counting():
        directed_adj_list = [[] for _ in range(n)]
        for u in range(n):
                for v in adj_list[u]:
                        if len(adj_list[v]) < len(adj_list[u]) or (len(adj_list[v]) ==
len(adj_list[u]) and v < u):
                                directed_adj_list[u].append(v)

        triangles = 0

        for u in range(n):
                nbr_set_of_u = set(directed_adj_list[u])

                for v in directed_adj_list[u]:
                                for w in directed_adj_list[v]:
                                        if(w in nbr_set_of_u):
                                                triangles += 1
                                                print("<"+str(u)+","+str(v)+","+str(w)+">")

        print(triangles)

improved_triangle_counting()


# optional
# improved version?
```