In [13]:
```
#Alan Uthuppan

import math
from math import comb
```

In [14]:
```
print("Problem 1\n")

n = 4000
p = 0.001

avg = p*(n-1)
print("Graph would have an average degree = " + str(avg) + "\n")

print("Graph would have a variance = " + str(avg*(1-p)) + "\n")

atLeastTwiceLarger = []
for i in range(math.ceil(2*avg), 100):
    atLeastTwiceLarger.append(comb(n-1, i)*(p**i)*(1-p)**(n-1-i))
print("# of nodes with degree greater than or equal to average times 2 = " + st
```

```
Problem 1

Graph would have an average degree = 3.999

Graph would have a variance = 3.9950010000000002

# of nodes with degree greater than or equal to average times 2 = 203.93917397
722828
```

In [15]:
```
print("Problem 2\n")

print("Part a:\nProbability of creating an edge = (# of edges)/(Maximum # of ec

print("Partb:\nWe need three nodes all sharing edges for a triangle. There are
```

```
Problem 2

Part a:
Probability of creating an edge = (# of edges)/(Maximum # of edges). Therefor
e, p = (c*n/2)/(n(n-1)/2) = c/(n-1)

Partb:
We need three nodes all sharing edges for a triangle. There are n choose 3 pos
sible triangles which equals n!/(3!(n-3)!). Every edge has a c/(n-1) probabili
ty of exisiting (as shown in part a). Therefore, in order for a particular tri
angle to exist there is a (c/(n-1))^3 chance.
Therefore, total number of triangles equals the probability of a triangle exis
ting times the # of potential triangles, which equals n!/(3!(n-3)!)*(c/(n-1))^
3, which simplifies to (c^3)/6 as n approaches infinity.
```