Submission

February 4, 2021

[7]:
```python
# Craig Fox
print('Name: Craig Fox')
# TUID: 915781095
print('TUID: 915781095')

from math import comb
import math
```

```
Name: Craig Fox
TUID: 915781095
```

[29]:
```python
# Problem 1
print('Problem 1\n')

# Erdős-Rényi graph: N=4000, p=0.001
n = 4000
p = 0.001

# What is the average degree of a node in this graph?
print('Erdős-Rényi graphs have an average degree = p(n-1)')
k = p*(n-1)
print('This graph would have an average degree = ' + str(k) + '\n')

# What is the variance in the degrees of the nodes?
print('Erdős-Rényi graphs have a variance = p(1-p)(n-1)')
print('This graph would have a variance = ' + str(k*(1-p)) + '\n')

# What is the expected number of nodes with a degree which is at least twice␣
↪larger that the average degree?
print('The Probability of a node having k degrees = (n-1 pick␣
↪k)(p^k)(1-p)^(n-1-k)')
at_least_twice_degree = []
for x in range(math.ceil(2*k), 100):
    #Note: The range should go from math.ceil(2*k) to n-1, but since n-1 is␣
↪3999 the factorial in the combination becomes incredibly large. 100 is␣
↪already high enough for a high accuracy
    at_least_twice_degree.append(comb(n-1, x)*(p**x)*(1-p)**(n-1-x))
```

```
print('So the number of nodes greater with greater than or equal to 2k degree =␣
↪' + str(sum(at_least_twice_degree)*n) + '\n')
```

```
Problem 1

Erdős-Rényi graphs have an average degree = p(n-1)
This graph would have an average degree = 3.999

Erdős-Rényi graphs have a variance = p(1-p)(n-1)
This graph would have a variance = 3.9950010000000002

The Probability of a node having k degrees = (n-1 pick k)(p^k)(1-p)^(n-1-k)
So the number of nodes greater with greater than or equal to 2k degree =
203.93917397722828
```

[32]:
```
# Problem 2
print('Problem 2\n')

# Consider Gn,p, an Erdös-Rényi random graph with n nodes, m edges, and mean␣
↪degree c

# a) Compute the probability p of creating an edge in Gn,p.
print('The maximum number of edge is n(n-1)/2. A graph has c*n/2 edges.␣
↪Therefore p = (c*n/2)/(n(n-1)/2) = c/(n-1)\n')

# b) Show that in the limit (large n) the expected number of triangles in Gn,p␣
↪is (1/6)*c^3
print('For a triangle to exist, there has to be three codes all sharing edges.␣
↪So there are n Pick 3 possible triangles which equals n!/(3!(n-3)!). Each␣
↪edge in the triangle has a c/(n-1) chance of exisiting (as shown in the␣
↪previous question). So for a specific triangle to appear there is a (c/
↪(n-1))^3 chance. So the total number of triangles = the probability of a␣
↪triangle times the number of potential triangles = n!/(3!(n-3)!)*(c/(n-1))^3.
↪ As n approaches infinity this simplies to (c^3)/6')
```

```
Problem 2

The maximum number of edge is n(n-1)/2. A graph has c*n/2 edges. Therefore p =
(c*n/2)/(n(n-1)/2) = c/(n-1)

For a triangle to exist, there has to be three codes all sharing edges. So there
are n Pick 3 possible triangles which equals n!/(3!(n-3)!). Each edge in the
triangle has a c/(n-1) chance of exisiting (as shown in the previous question).
So for a specific triangle to appear there is a (c/(n-1))^3 chance. So the total
number of triangles = the probability of a triangle times the number of
potential triangles = n!/(3!(n-3)!)*(c/(n-1))^3. As n approaches infinity this
```

```
simplies to (c^3)/6
```

```
[ ]:
```