بسمه تعالی

دانشکده مهندسی برق

آمار و احتمال مهندسی (۲۵۷۳۲۴) (نیم‌سال ۱۴۰۰-۰۱)

دکتر مداح علی

پروژه پایانی (نرگس عرفان - ثریا چرکس)

آخرین مهلت ارسال تا ۱۴۰۰/۱۱/۲۱

درخت دوستی بنشان!

پرسش تئوری ۱

احتمال اینکه این هکر موفّق شده باشد تمام روابط دوستی را به درستی تعیین کند را به کمک توزیع ۲ جمله ای میتوان به دست آورد
با توجه به توزیع ۲ جمله ای خواهیم داشت:

$$P(T) = \left(\binom{n}{2} \atop m \right) p^m (1-p) \binom{n}{2}^{-m}$$

پرسش تئوری ۲

اگر مقدار دقیق m را بداتیم ، احتمال اینکه تمام روابط دوستی درست باشند، از رابطه زیر به دست می آید :

$$\left(\binom{n}{2} \atop m \right)$$

که $\binom{n}{2}$ تعداد کل میباشد و m تعداد روابط دوستی.

پرسش تئوری ۳

با استفاده از توزیع ۲ جمله ای مانند بخش ۱ داریم :

$$\binom{\binom{n}{2}}{\frac{m}{5}} p^{\frac{m}{5}} (1-p)^{\binom{n}{2}-\frac{m}{5}}$$

تنها تفاوت این است که از $\frac{m}{5}$ استفاده میکنیم تا با احتمال ۲۰ درصد به جواب برسیم.

پرسش شبیه سازی ۱

```
In [16]: import matplotlib.pyplot as plt
         from numpy import *
         import networkx as nx
         n = 1000
         p = 0.0034
         m = 3000

         l = []
         for i in range(10):
             g = nx.gnp_random_graph(n, p)
             s = 0
             for i in range(n):
                 s += len(g.adj[i])
             l.append(s)

         ans = sum(l)/len(l)
         print(l)
         print(ans)
         print(abs((ans - m)/m))

         [3270, 3388, 3418, 3510, 3352, 3344, 3464, 3316, 3402, 3514]
         3397.8
         0.13260000000000005
```

شکل ۱: شبیه سازی شماره ۱ و نتایج آن

ابتدا با استفاده از دستور موجود در کتابخانه networkx ، گرافی با تعداد راس ۱۰۰۰ و احتمال ارتباط 0.0034 بین هر دو راس تشکیل میدهیم.سپس با استفاده از دستور adj. ،تعداد یال ها را به دست می آوریم و آن را ۱۰ بار تکرار میکنیم.در خروجی به ترتیب تعداد یال ها در هر بار تکرار، میانگین و خطای آن با m داده شده را مشاهده میکنیم.همانطور که دیده میشود ، میزان این خطا از ۵ درصد بیشتر است.

پرسش تئوری ۴

میانگین از رابطه $\binom{\binom{n}{2}}{\frac{m}{5}} p^{\frac{m}{5}} (1-p)^{\binom{n}{2}-\frac{m}{5}}$ به دست می آید که مقدار آن برابر است با :

$$\frac{1000*999}{2} * 0.0034 = 1698$$

برای برابر بودن مقدار آن با m کافیست قرار دهیم:

$$\binom{\binom{n}{2}}{\frac{m}{5}} p^{\frac{m}{5}} (1-p)^{\binom{n}{2}-\frac{m}{5}} P = m$$

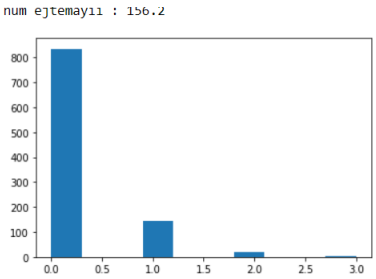
خلوت گزیده را به تماشا چه حاجت است؟

پرسش شبیه سازی ۲

```
In [4]: import matplotlib.pyplot as plt
from numpy import *
import networkx as nx
n = 1000
p = 0.00016

l = []
for i in range(10):
    g = nx.gnp_random_graph(n, p)
    s = 0
    for i in range(n):
        s += len(g.adj[i])
    s = s // 2
    s = s / n
    num = 0
    for i in range(n):
        if len(g.adj[i]) > s:
            num += 1
    l.append(num)

ans = sum(l)/len(l)
print(f'num ejtemayii : {ans}')

g = nx.gnp_random_graph(n, p)
l = []

for i in range(n):
    l.append(len(g.adj[i]))

plt.hist(l)
plt.show()

num ejtemayii : 156.2
```

شکل ۲: شبیه سازی شماره ۲ و نتایج آن

در این شبیه سازی نیز پس از تشکیل گراف رندم با تعداد راس و احتمال داده شده و تکرار ۱۰ بار ، تعداد دوستان هر فرد با میانگین تعداد دوستان را مقایسه میکنم و در صورت اجتماعی بودن آن فرد را در دسته افراد اجتماعی قرار میدهیم. نمودار به دست آمده به این شکل خواهد بود.



شکل ۳: نمودار شبیه سازی شماره ۲

پرسش تئوری ۵

$$\mathbb{E}[X] = \sum_{k=0}^{n-1} k\mathbb{P}[\alpha = k] = \sum_{k=0}^{n-1} k \begin{pmatrix} n-1 \\ k \end{pmatrix} p^k(1-p)^{n-p-1} = p(n-1)\sum_{k=1}^{n-1} \begin{pmatrix} n-2 \\ k-1 \end{pmatrix} p^{k-1}(1-p)^{n-p-1} =$$

$$p(n-1)$$

پرسش تئوری ۶

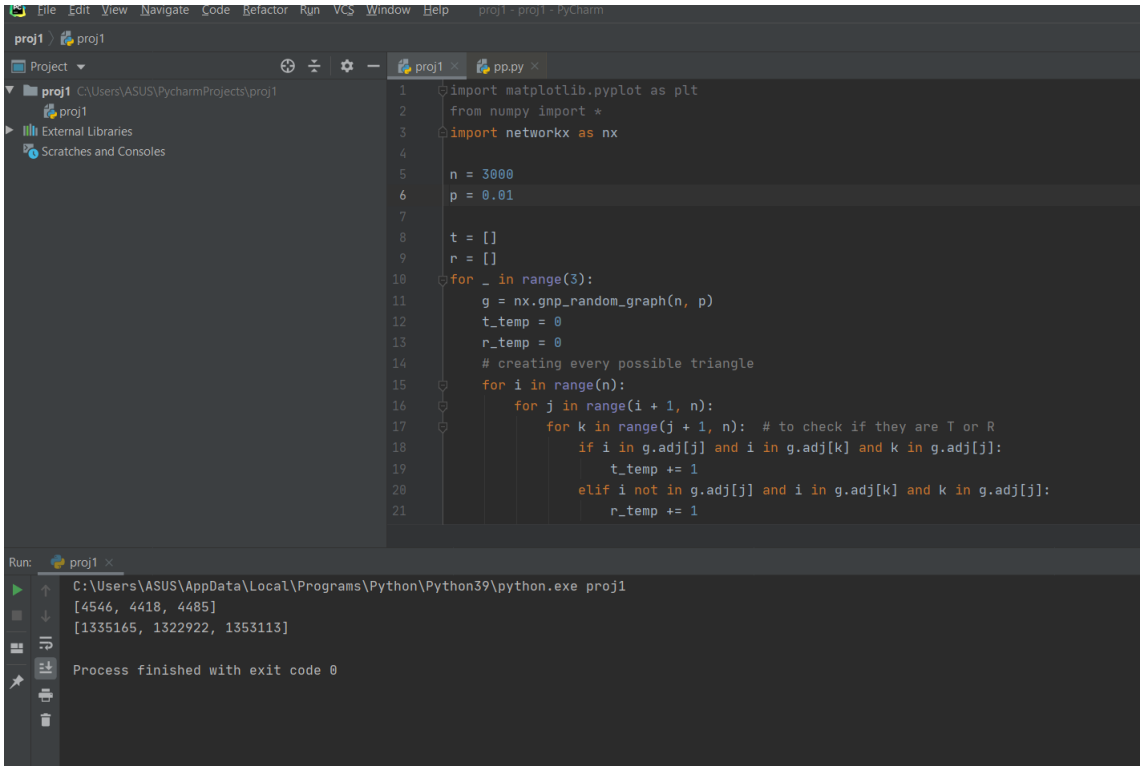
اگر A را اجتماعی بودن هر فرد و Y را تعداد افراد اجتماعی و $\mathbb{E}[x] = 0.16$ در نظر بگیریم ، خواهیم داشت:

$$\sum_{k=0.16}^{n-1} \left(n-1k \right) p^k (1-p) n-k-1 = 1 - \begin{pmatrix} n-1 \\ 0 \end{pmatrix} p^0 (1-p)^{n-1} = 1-(1-p)^{n-1} \simeq 0.15$$

$$\mathbb{E}[Y] = \sum_{k=0}^{n} k\mathbb{P}(Y=k) = \sum_{k=1}^{n} k \begin{pmatrix} n \\ k \end{pmatrix} \mathbb{P}[A]^k (1-\mathbb{P}[A])^{n-k} = \mathbb{A}n = 0.15n = 150$$

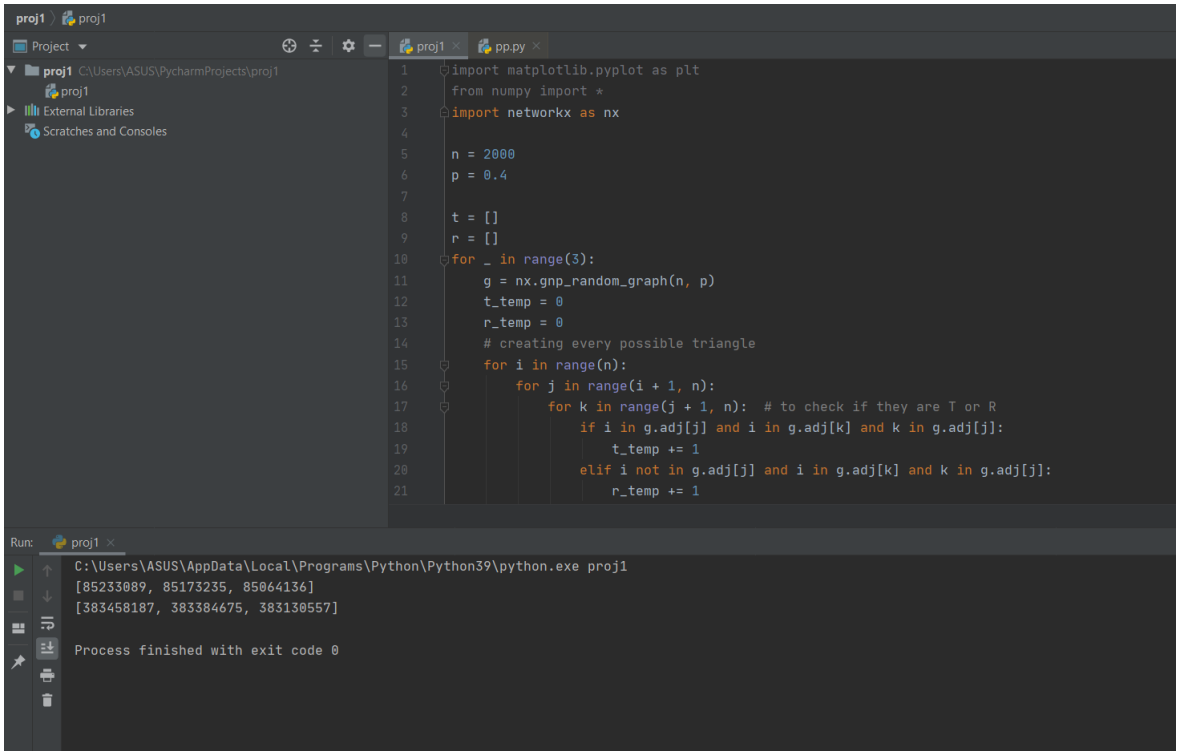
هواداران کویش را چو جان خویشتن دارم؟

پرسش شبیه سازی ۳



شکل ۴: نمودار شبیه سازی شماره ۳

پرسش شبیه سازی ۴



شکل ۵: نمودار شبیه سازی شماره ۴

پرسش تئوری ۷

$$i = \begin{cases} 1 & p^3 \\ 0 & 0 \end{cases} \Rightarrow \mathbb{E}[X] \sum_{0}^{n-1} X_i \mathbb{P}(X = x) \quad = \binom{n}{3} \sum_{0}^{n-1} ip^3 \tag{۱}$$

$$i = \begin{cases} 1 & p^2(1-p) \\ 0 & 0 \end{cases} \Rightarrow \mathbb{E}[X] = \binom{n}{3} \sum_{0}^{n-1} x_i p^2(1-p) \tag{۲}$$

پرسش تئوری ۸

$$\mathbb{P} = \frac{P^3}{P^3 + 2P^2(1-p)}$$

```python
import matplotlib.pyplot as plt
from numpy import *
import networkx as nx
n = 1000
p = 0.003

g = nx.gnp_random_graph(n, p)


l = []
for i in range(n):
    temp = g.adj[i]

    s = 0
    for j in temp:
        for k in temp:
            if j == k:
                continue
            if k in g.adj[j]:
                s += 1
    s = s // 2
    l.append(s)

ans = sum(l)/len(l)
print(ans)
```

```
0.009
```

شکل ۶: شبیه سازی شماره۵ و نتایج آن

پرسش شبیه سازی ۵

در این شبیه سازی نیز پس از ساختن گراف رندم ، با چک کردن این که آیا دو راس با هم برخورد دارند،تعداد روابط دوستی را به دست می آوریم و بعد از آن میانگین میگیریم.

پرسش تئوری ۹

اگر ازین وجه در نظر بگیریم که افرادی که با هم دوست اند دوست های مشترک بیشتری با هم نسبت به آدم های رندم دارند،میتوان گفت میانگین بیشتر میشود.

پرسش تئوری ۱۰

$$\begin{cases} X_A: & \textbf{تعداد دوستی فرد A} \\ Y_a: & \textbf{تعداد دوست های A} \end{cases} \Rightarrow \textbf{با توزیع دو جمله ای داریم:} \quad (۳)$$

$$X_A|Y_A = y \sim Binomial(\begin{pmatrix} y \\ 2 \end{pmatrix}, P) \Rightarrow \mathbb{E}[X_A] = \mathbb{E}[\mathbb{E}[X_A|Y_A]] = P\mathbb{E}[\begin{pmatrix} y \\ 2 \end{pmatrix}] = \frac{P}{2}(\mathbb{E}[y^2] - \mathbb{E}[y]) =$$

$$\frac{(n-1)P^2(1-P)}{2} + \frac{(n-1)^2P^3}{2} - \frac{(n-1)P^2}{2}$$

```python
# 6
import matplotlib.pyplot as plt
from numpy import *
import networkx as nx
n = 1000
p = 0.0033

g = nx.gnp_random_graph(n, p)
l = []

for i in range(n):
    for j in range(i + 1, n):
        try:
            l.append(nx.shortest_path_length(g, source = i, target = j))
        except nx.NetworkXNoPath:
            continue
print(f'expected dist : {sum(l)/len(l)}')
```

```
expected dist : 5.62979897688725
```

شکل ۷: شبیه سازی شماره۶ و نتایج آن

من از دیار حبیبم نه از بلاد غریب

پرسش شبیه سازی ۶

در این قسمت با استفاده از دستور کتابخانه networkx ، ابتدا فاصله ۲ راس را محاسبه میکنیم سپس از آن میانگین میگیریم.

پرسش شبیه سازی ۷

```python
import matplotlib.pyplot as plt
from numpy import *
import networkx as nx

n = 50
p = 0.34

l = []
for _ in range(100):
    g = nx.gnp_random_graph(n, p)
    l.append(nx.diameter(g))

print(sum(l)/len(l))
```

```
2.79
```

شکل ۸: شبیه سازی شماره۷ و نتایج آن

داخل حلقه for با ۱۰ بار تکرار ، گراف را تشکیل میدهیم و سپس با بررسی گراف هربار بیشترین فاصله را داخل آرایه ای قرار میدهیم و در نهایت از آن میانگین میگیریم.

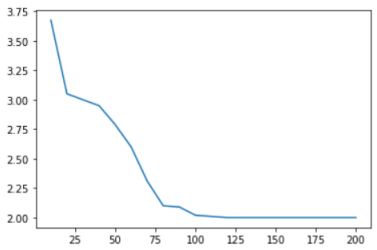
پرسش شبیه سازی ۸

ابتدا شبیه سازی به کار برده شده در بخش قبلی را به صورت تابعی در میاوریم که استفاده از آن راحت تر شود.سپس آن ها را به بازه بندی ۱۰ تایی تقسیم کرده و نمودار را میکشیم.میبینیم که با افزایش این،n مقدار به صفر میل میکند.

پرسش تئوری ۱۱

$$p[X = x] = p^x(1-p)^{1-x} \Rightarrow p[I_{UV} = 1] = p^1(1-p)^{1-1} = p \Rightarrow p = (2p(1-p) + (1-P)^2)^{n-2}$$

```python
import matplotlib.pyplot as plt
from numpy import *
import networkx as nx
def diameter_mean(n, p):
    l = []
    for _ in range(100):
        g = nx.gnp_random_graph(n, p)
        try:
            l.append(nx.diameter(g))
        except:
            continue
    return sum(l)/len(l)

x = []
y = []
for i in range(10, 201, 10):
    x.append(i)
    y.append(diameter_mean(i, 0.34))

plt.plot(x, y)
plt.show()
```



شکل ۹: شبیه سازی شماره۸ و نتایج آن

پرسش تئوری ۱۲

$$E[X_n] = E[I_{uv} = 1] = \sum_0^n p[I_{vu}]x_i = \binom{n}{2} p[I_{uv} = 1]$$

پرسش تئوری ۱۳

نامساوی مارکف:

$P(X_n \geq 1) \leq \frac{\mathbf{E}[x]}{1}$ اگر n به سمت بی نهایت برود ،مقدار به صفر میل میکند.

پرسش تئوری ۱۴

$$\frac{n(n-1)}{2}(1-p)^{n-2} \xrightarrow{n\to\infty} A(0) = 0 \Rightarrow P[X_n = 1] = 0$$

از عبارت بالا میتوان نتیجه گرفت که در گراف تصادفی اگر تعداد رُئوس زیاد باشند ، احتمال اینکه یک جفت راس همسایه مشترک نداشته باشند صفر است .همچنین حداقل فاصله ۲ است و از طرفی قطر متاگراف برای n بزرگ به P بستگی ندارد.

و ان یکاد بخوانید!

پرسش شبیه سازی ۹

```python
import matplotlib.pyplot as plt
from numpy import *
import networkx as nx

n = 100
p = 0.34



l = []
for _ in range(100):
    g = nx.gnp_random_graph(n, p)
    temp = nx.triangles(g)
    s = 0
    for i in temp:
        s += temp[i]
    s = s // 3
    l.append(s)

print(sum(l)/len(l))
```

```
6338.92
```

شکل ۱۰: شبیه سازی شماره۹ و نتایج آن

برای پیدا کردن روابط دوستی ۳ نفره از دستور مثلث استفاده میکنیم . به این ترتیب میتوانیم رئوسی که با هم تشکیل مثلث یا به عبارتی تشکیل رابطه دوستی ۳ نفره میدهند را به دست آوریم.

پرسش شبیه سازی ۱۰

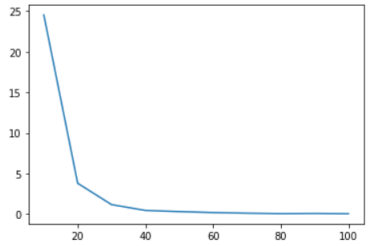
در این بخش نیز شبیه سازی بخش قبل را به صورت تابع در آورده ایم و از آن استفاده میکنیم.سپس با بازه بندی n و تعین p بر حسب n ، نمودار خواسته شده را رسم میکنیم.مشاهده میکنیم که میانگین به صفر میل میکند که منطقی به نظر می آید زیرا اگر T را تعداد مثلث ها در نظر بگیریم ، داریم :

$$\mathbb{E}[T] = \binom{n}{3} p^3 = \frac{n^3}{n^6}$$

که با افزایش n به صفر میل میکند.

```
import matplotlib.pyplot as plt
from numpy import *
import networkx as nx

def get_mean(n, p):


    l = []
    for _ in range(100):
        g = nx.gnp_random_graph(n, p)
        temp = nx.triangles(g)
        s = 0
        for i in temp:
            s += temp[i]
        s = s // 3
        l.append(s)

    return sum(l)/len(l)

n = []
for i in range(10, 101, 10):
    n.append(i)

y = []

for i in n:
    y.append(get_mean(i, 60/(i**2)))

plt.plot(n, y)
plt.show()
```

شکل ۱۱: شبیه سازی شماره۱۰ و نتایج آن



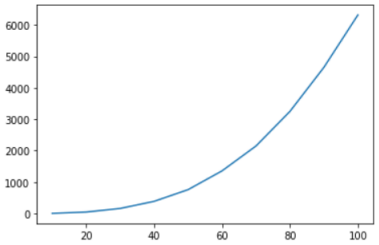
شکل ۱۲: نمودار شبیه سازی ۱۰

پرسش شبیه سازی ۱۱

در این بخش نمودار به دست آمده واگرا میشود زیرا طبق معادله گفته شده در بالا با ثابت شدن p ، مقدار مخرج ثابت میماند و با زیاد شدن صورت ، مقدار آن زیاد میشود.

```python
import matplotlib.pyplot as plt
from numpy import *
import networkx as nx

def get_mean(n, p):


    l = []
    for _ in range(100):
        g = nx.gnp_random_graph(n, p)
        temp = nx.triangles(g)
        s = 0
        for i in temp:
            s += temp[i]
        s = s // 3
        l.append(s)

    return sum(l)/len(l)

n = []
for i in range(10, 101, 10):
    n.append(i)

y = []

for i in n:
    y.append(get_mean(i, 0.34))

plt.plot(n, y)
plt.show()
```

شکل ۱۳: شبیه سازی شماره۱۱ و نتایج آن



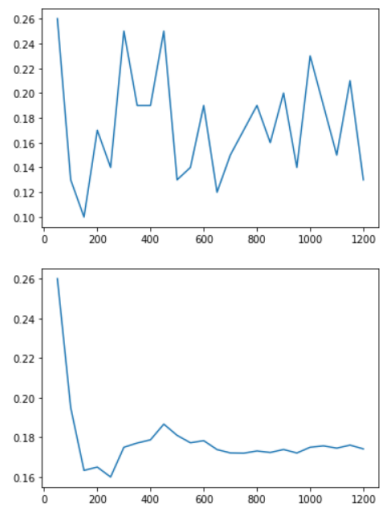
شکل ۱۴: نمودار شبیه سازی ۱۱

پرسش شبیه سازی ۱۲

میانگین تجمعی در این بخش به عددی ثابت حدود ۱۴.۰میل میکند که با توجه به تعریف آن درست است جون با توجه به نمودار اول ، میانگین ها نوسانی بین ۲۶.۰ و ۱.۰ دارند که به مرور این نوسان کم میشود پس با جمع شدن تکه ای اعداد و تقسیم شدن به مقداری ثابت ، این نوسانات

```
    l = []
    for _ in range(100):
        g = nx.gnp_random_graph(n, p)
        temp = nx.triangles(g)
        s = 0
        for i in temp:
            s += temp[i]
        s = s // 3
        l.append(s)
    temp = sum(l)/len(l)
    return temp

n = []
for i in range(50, 1201, 50):
    n.append(i)

y = []

for i in n:
    y.append(get_mean(i, 1/i))

plt.plot(n, y)
plt.show()
y2 = []
for i in range(len(y)):
    s = 0
    for j in range(i + 1):
        s += y[j]
    s = s / (i + 1)
    y2.append(s)


plt.plot(n, y2)
plt.show()
```

شکل ۱۵: شبیه سازی شماره۱۲ و نتایج آن



شکل ۱۶: نمودار شبیه سازی ۱۲

کمتر به چشم می آیند.

پرسش تئوری ۱۵

مثل این است که سه راس یک گراف با احتمال p به هم وصل باشند پس $\mathbb{P}[I_{u,v,w}=1]=p^3$

پرسش تئوری ۱۶

$$\mathbb{E}[X]=\sum_i x_i P(X=1)=\binom{n}{3}p^3$$

پرسش تئوری ۱۷

با استفاده از نامساوی مارکف، ($P(X_n \geq 1) \leq \frac{\mathbf{E}[x]}{1}$)

$$\mathbb{P}[T_3, n\geq 1] \geq \mathbb{E}[T_3, n] \xrightarrow{\text{کران بالا}} \mathbb{P}[T_3, n\geq 1] = \mathbb{E}[T_3, n] = \binom{n}{3}p^3$$

با قرار دادن $p(n)=\frac{1}{n^2}$ ،

$$\mathbb{P}[T_3, n\geq 1] = \binom{n}{3}\frac{1}{n^6}$$

با میل دادن n به بینهایت، کران به بینهایت میل میکند.

پرسش تئوری ۱۸

پرسش تئوری ۱۹

برای این سوال ، سه حالت داریم:
۱) ۲ مثلث جدا:

$$\binom{n}{3}\binom{n-3}{3}\frac{p^6}{2}$$

۲) ۲ مثلث راس مشترک :

$$\binom{n}{3}\binom{n-2}{2}\frac{p^6}{2}$$

۳) ۲ مثلث ضلع مشترک :

$$\binom{n}{3} \frac{n-3}{2} p^5$$

پرسش تئوری ۲۰

$$\mathbb{E}[T^2] = \mathbb{E}[(I_1 + I_2 + ...)(I_1 + I_2 + ...)] = \mathbb{E}[{I_1}^2 + {I_2}^2 + ... + 2(I_1 I_2 + I_2 I_3 + ...)]$$

با توجه به تئوری ۱۹ و ۱۶ و خاصیت خطی بودن امید ریاضی ، مقدار برابر است با :

$$\frac{n^3}{6} p^3 + \frac{n^6}{36} p^6 + \frac{n^6}{36} p^5$$

پرسش تئوری ۲۱

با توجه به بخش های ۲۰ و ۱۶و با قرار دادن $p = c$ داریم:

$$\lim_{n\to\infty} \frac{\mathbb{E}[{T_{n,3}}^2] - \mathbb{E}[T_{n,3}]^2}{\mathbb{E}[T_{n,3}]^2} = 0$$

پرسش تئوری ۲۲

با توجه به صورت اثبات ۱۸ و تئوری ۲۱ ،

$$\forall n \in N : \mathbb{E}[T_{n,3}] > 0$$
$$\lim_{n\to\infty} \frac{Var[T_{n,3}]}{\mathbb{E}^2[T_{n,3}]}$$

T تعداد مثلث ها است. اگر تعداد مثلث ها صفر باشد با در نظر گرفتن هر مثلث ،

پرسش تئوری ۲۳

میدانیم:

$$\mathbb{E}[g(x)] = \sum_x g(x)\mathbb{P}(X = x)$$

پس،

$$\mathbb{E}(x(x-1)...(x-r+1)) = \sum_{k=0}^{\infty} k(k-1)...(k-r+1)e^{-\lambda}\frac{\lambda^k}{k!}$$

چون برای k های کوچک تر از r ، g(x)=۰ است، داریم:

$$\mathbb{E}(x(x-1)...(x-r+1)) = \sum_{k=r}^{\infty} k(k-1)...(k-r+1)e^{-\lambda}\frac{\lambda^k}{k!}$$

اگر y=x-r قرار دهیم، داریم:

$\sum_{y=0}^{\infty}(y+r)...(y+1)\frac{e^{-\lambda}\lambda^{y+r}}{(y+r)!}$

پس :

$\lambda^r \sum_{y=0}^{\infty} e^{-\lambda}\frac{\lambda^y}{y!}$

و میدانیم عبارت درون سیگما برابر با یک است پس عبارت برابر است با λ^r

پرسش تئوری ۲۴

پرسش تئوری ۲۵

قومی به جد و جهد نهادند وصل دوست، قومی دگر حواله به تقدیر می کنند!

پرسش شبیه سازی ۱۳

```python
import matplotlib.pyplot as plt
from numpy import *
import networkx as nx

c = [] # true, false for connected
l = [] # num no friends

n = 150
p = 0.2

for _ in range(100):
    g = nx.gnp_random_graph(n, p)

    c.append(nx.is_connected(g))

    num = 0
    for i in range(n):
        if len(g.adj[i]) == 0:
            num += 1
    l.append(num)

print(f'prob. connected : {sum(c)/len(c)}')
print(f'prob. tanha : {sum(l)/(len(l)*n)}')
```

```
prob. connected : 1.0
prob. tanha : 0.0
```

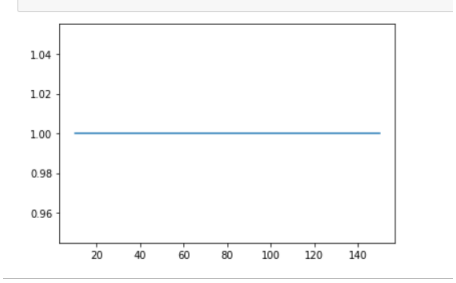
شکل ۱۷: شبیه سازی شماره۱۳ و نتایج آن

با توجه به خروجی شبیه سازی ، متوجه میشویم که احتمال اینکه گراف همبند نباشد یا شخصی بدون دوست باشد وجود ندارد.

پرسش شبیه سازی ۱۴

```python
import matplotlib.pyplot as plt
from numpy import *
import numpy as np
import networkx as nx
def connected_no_friends(n, p):
    c = [] # true, false for connected
    l = [] # num no friends
    for _ in range(100):
        g = nx.gnp_random_graph(n, p)
        c.append(nx.is_connected(g))
        num = 0
        for i in range(n):
            if len(g.adj[i]) == 0:
                num += 1
        l.append(num)
    return (sum(c)/len(c), sum(l)/(len(l)*n))


def p(n):
    return (4*np.log(n))/(n)

x = []
y1 = []
y2 = []
for i in range(10, 151, 10):
    x.append(i)
    temp = connected_no_friends(i, p(i))
    y1.append(temp[0])
    y2.append(temp[1])

plt.plot(x, y1)
plt.show()
```

شکل ۱۸: شبیه سازی شماره۱۴ و نتایج آن



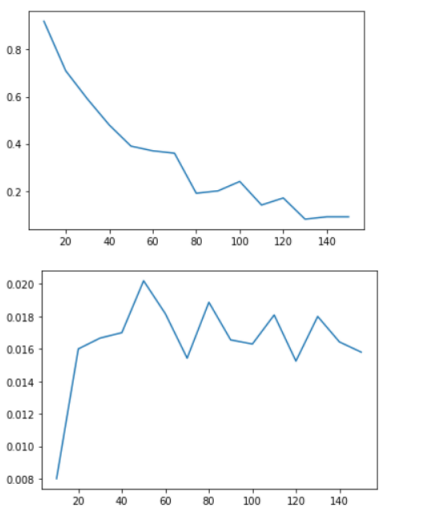
شکل ۱۹: نمودار شبیه سازی ۱۴

در این بخش نیز نتایج نهایی با تغییر P ، فرقی با قسمت قبل ندارند و همچنان میبینیم که به همان مقدار ۱ رسیدیم.

پرسش شبیه سازی ۱۵

```python
import matplotlib.pyplot as plt
from numpy import *
import networkx as nx
def connected_no_friends(n, p):
    c = [] # true, false for connected
    l = [] # num no friends
    for _ in range(100):
        g = nx.gnp_random_graph(n, p)
        c.append(nx.is_connected(g))
        num = 0
        for i in range(n):
            if len(g.adj[i]) == 0:
                num += 1
        l.append(num)
    return (sum(c)/len(c), sum(l)/(len(l)*n))


def p(n):
    return (4)/(n)

x = []
y1 = []
y2 = []
for i in range(10, 151, 10):
    x.append(i)
    temp = connected_no_friends(i, p(i))
    y1.append(temp[0])
    y2.append(temp[1])

plt.plot(x, y1)
plt.show()
plt.plot(x, y2)
plt.show()
```

شکل ۲۰: شبیه سازی شماره۱۵ و نتایج آن



شکل ۲۱: نمودار شبیه سازی ۱۵

در این بخش با آمدن n در مخرج و افزایش آن،سبب تغییر روند میشود و میبینیم که نمودار همبند بودن نزولی و نمودار عدم داشتن دوست ، صعودی میشود.

پرسش تئوری ۲۶

j راس مشخص داریم که باید مولفه همبندی باشند.حال به تعریف مولفه همبندی میپردازیم.

مولفه هم بندی:

بین هر دو راس راهی وجود داشته باشد که به هم متصل باشند و به هیچ راس دیگری متصل نباشند.
با استفاده از تعریف بالا اگر میخواهیم $K_i^j = 1$ باشد،پس راسj داریم که به n-j راس دیگر هیچ راهی نداشته باشد q^{n-j}

پس احتمال آن برای هر راس در مولفه که به هیچ کدام از n-j راس دیگر یال نداشته باشند برابر است با $q^{(n-j)j}$ چون j راس داریم. راس های این مولفه به راس های دیگر یال نداشته باشد پس سقف این احتمال همین میشود.

پرسش تئوری ۲۷

پرسش تئوری ۲۸

پرسش تئوری ۲۹