باسمه تعالی

دانشگاه صنعتی شریف
دانشکده مهندسی برق

پروژه درس آمار و احتمال مهندسی

متاگراف

استاد درس

دکتر محمّدعلی مدّاح‌علی

دانشجویان

ملیکا رجبی ۹۹۱۰۱۶۰۸

سید معین الدین مکیان ۹۹۱۰۲۲۶۴

بهمن ماه ۱۴۰۰

فهرست مطالب

۱ بلاتشبیه مقدمّه

در این پروژه با گراف تصادفی[1] (ER graph) سروکار داریم که در طول پروژه آن را به اختصار G می‌نامیم. در این بخش به معرفی تابعی می‌پردازیم که این گراف را با دریافت n (تعداد رأس‌ها) و p (احتمال به هم وصل بودن هردو رأس است) تشکیل می‌دهد.

در تابع `make_random_graph`، ابتدا با استفاده از کتابخانه networkx گرافی با n رأس تولید می‌کنیم و سپس با استفاده از توابع `random_pairs` و `all_pairs` که تعریف کرده‌ایم، ابتدا یال‌هایی که طبق احتمال p تولید می‌شوند را تولید می‌کنیم(با استفاده از yield) و سپس به گراف اولیه اضافه می‌کنیم.

همچنین به دلیل استفاده زیاد از میانگین در طول پروژه تابعی برای محاسبه میانگین یک لیست تعریف می‌کنیم که در ابتدا همه اعضای آن لیست را با هم جمع می‌کند و سپس بر طول آن لیست تقسیم می‌کند.

(قطعه کد ۱):

```
import networkx as nx
import numpy as np

def all_pairs(nodes):
    for i, u in enumerate(nodes):
        for j, v in enumerate(nodes):
            if i>j:
                yield u, v

def random_pairs(nodes, p):
    for edge in all_pairs(nodes):
        if flip(p):
            yield edge

def flip(p):
    return np.random.random() < p

def make_random_graph(n, p):
    G = nx.Graph()
    nodes = range(n)
    G.add_nodes_from(nodes)
    G.add_edges_from(random_pairs(nodes, p))
    return G

def average(lst):
    return sum(lst) / len(lst)
```

البته می‌توانستیم از تابع `networkx.erdos_reyni_graph` که از توابع آماده در کتابخانه network است، استفاده کنیم ولی توابعی که نوشتیم سرعت بسیار بالاتری نسبت به تابع این کتابخانه دارد.

[1] The ER graph $G(n, p)$ contains n nodes, and each pair of nodes is connected by an edge with probability p

۲ درخت دوستی بنشان!

۱.۲ پرسش تئوری ۱

ابتدا مسئله را به صورت گرافی با n راس شبیه سازی می‌کنیم به طوری که هر رأس نشان دهنده‌ی یک نفر در شبکه و هر یال نشان دهنده‌ی رابطه ی دوستی بین دو رأس مجاورش است. می‌دانیم که با احتمال p بین هر دو راس در این گراف یک یال وجود دارد. حالا می‌خواهیم احتمال صحیح بودن تمام روابط دوستی یا به عبارتی صحیح بودن جای تمام یال‌ها را به دست بیاوریم.
همان طور که در سوال گفته شده، درکل m رابطه‌ی دوستی بین این n نفر وجود دارد. پس باید دقیقا m یال در گراف وجود داشته باشد. همچنین میدانیم که این گراف در کل می تواند $\binom{n}{2}$ یال داشته باشد بنابراین:

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

۲.۲ پرسش تئوری ۲

در این بخش برخلاف بقیه بخش ها؛ وجود یال‌ها با احتمال p و به تعداد دلخواه نیست. بلکه از آنجایی که هکر تعداد دقیق m یعنی تعداد یال ها را می‌داند؛ تنها باید از بین همه‌ی یال‌های ممکن؛ m تا را انتخاب کند. این فرایند به $\binom{\binom{n}{2}}{m}$ حالت امکان پذیر است.
همچنین می‌دانیم که از بین همه‌ی این حالات تنها یک حالت مدنظر است که همان حالت درست روابط دوستی است.
درنتیجه احتمال خواسته شده برابر است با:

$$\frac{1}{\binom{\binom{n}{2}}{m}} = \frac{1}{\binom{\frac{n(n-1)}{2}}{m}}$$

۳.۲ پرسش تئوری ۳

در این قسمت از ما خواسته شده احتمال اینکه هکر بتواند دقیقا $\frac{1}{5}$ روابط دوستی یا به عبارتی $\frac{1}{5}$ تعداد یال‌ها را درست تشخیص دهد را محاسبه کنیم.
پس همانند پرسش تئوری ۱ عمل می‌کنیم با این تفاوت که ابتدا باید $\frac{m}{5}$ یال مدنظرمان را از بین m یال انتخاب کنیم.
پس احتمال خواسته شده برابر می‌شود با:

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

البته باید توجه کنیم که چون صورت سوال درمورد بقیه‌ی روابط بین افراد توضیح دقیقی نداده، در اینجا ما اینگونه درنظر گرفتیم که فقط و فقط همان ۲۰ درصد را دوست و بقیه را غیردوست تشخیص دهد.

۴.۲ پرسش شبیه سازی ۱

در یک اجرای نمونه کد این بخش ۱۶۷۰/۸ بدست آمد که با مقدار واقعی m بسیار فاصله دارد.

۵.۲ پرسش تئوری ۴

همانطور که گفته شد با احتمال p رابطه‌ی دوستی بین افراد برقرار است. این یعنی p برابرِ $\binom{n}{2}$ یال ممکن؛ رابطه‌ی دوستی تشخیص داده می شود:

$$\binom{n}{2}p = \binom{1000}{2} \times \frac{34}{10000} = 1683.3$$

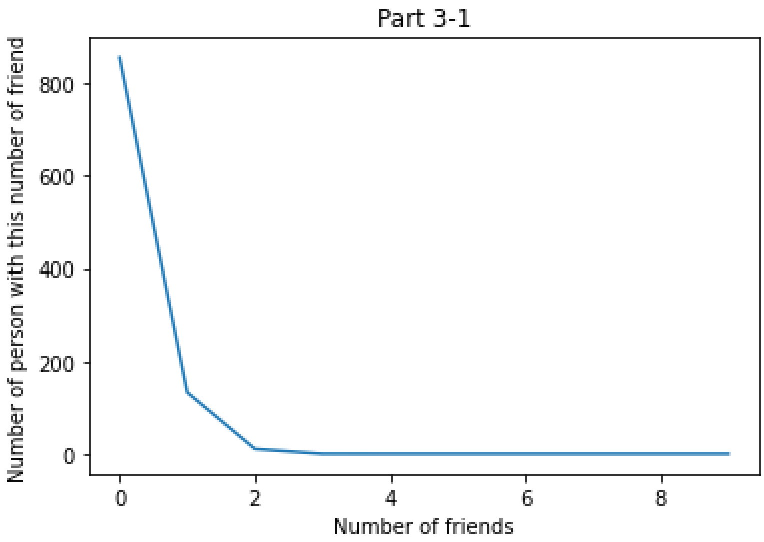
به طور کلی هم، بدیهی است زمانی این مقدار میانگین به مقدار واقعی m نزدیک می شود که رابطه‌ی زیر بین m, n, p برقرار باشد:

$$\binom{n}{2}p = m \longrightarrow \frac{n(n-1)}{2}p = m$$

۳ خلوت‌گزیده را به تماشا چه حاجت است؟

۱.۳ پرسش شبیه سازی ۲

تعداد متوسط افراد اجتماعی در یک اجرای نمونه کد این بخش ۱۴۴ بدست آمد.



شکل ۱: شبیه سازی دوم

۲.۳ پرسش تئوری ۵

تعداد دوست‌های هر فرد؛ درواقع همان درجه‌ی رأس متناظر با آن فرد است. پس برای به دست آوردن میانگین تعداد دوست‌های هر فرد باید میانگین درجه‌های گراف را به دست بیاوریم.
برای این کار ابتدا تعداد یال‌ها را دوبرابر کرده و سپس بر تعداد رئوس تقسیم می‌کنیم:

$$\frac{m \times 2}{n} = \frac{\binom{n}{2} \times 2p}{n} = (n-1)p$$
$$= 999 \times 16 \times 10^{-5} = 0.15984$$

۳.۳ پرسش تئوری ۶

می‌دانیم اگر تعداد دوستان فردی از میانگین بیشتر باشد؛ آن فرد اجتماعی و درغیر این صورت؛ گوشه گیر است. در این قسمت از ما خواسته شده احتمال اجتماعی بودن فرد را پیدا کنیم. یعنی احتمال اینکه تعداد دوستان فرد انتخابی بیشتر از میانگین باشد. همچنین میانگین تعداد دوست‌های هر فرد هم طبق پرسش تئوری ۵ برابر ۰/۱۵۹۸۴ است:

$$X : \text{تعداد دوست های فرد مورد نظر} \longrightarrow \mathbb{P}[X \geq 0.15984]$$

چون تعداد دوست عددی صحیح است؛ پس این احتمال برابر عبارت زیر است:

$$\mathbb{P}[X \geq 1] = 1 - \mathbb{P}[X < 1] = 1 - \mathbb{P}[X = 0]$$

حالا فرض می‌کنیم تعداد دوست‌های هر فرد از توزیع پواسون با پارامتر λ پیروی می‌کند.
می‌دانیم که پارامتر λ نشان دهنده‌ی میانگین توزیع پواسون است. پس همانطور که در پرسش تئوری ۵ به دست آوردیم؛ مقدار آن برابر ۰/۱۵۹۸۴ است و داریم:

$$X \sim Poisson(\lambda = 0.15984)$$

$$1 - \mathbb{P}[X = 0] = 1 - \frac{\lambda^0}{0!}e^{-\lambda} = 1 - e^{-0.15984} = 0.14772$$

حالا می‌خواهیم امید ریاضی تعداد افراد اجتماعی را به دست آوریم. برای این کار طبق فرمول داریم:

$$\begin{aligned} \mathbb{E}[X] &= \sum_{i=1}^{n} \mathbb{P}[X \geq 1] \\ &= n \times 0.14772 \xrightarrow{n=1000} 147.72 \rightarrow \lceil 147.72 \rceil = 148 \end{aligned}$$

۴ هواداران کویش را چو جان خویشتن دارم؟

۱.۴ پرسش شبیه سازی ۳

در یک اجرا نمونه کد این بخش میانگین روابط دارای خاصیت تراگذری ۴۴۵۰/۶ و میانگین تعداد روابط دارای خاصیت رقابت نیز ۱۱۵۳۵۵۳/۰ بدست آمد.

۲.۴ پرسش شبیه سازی ۴

در این بخش با $n = 2000$ و $p = 0.2$ با سه بار تکرار نمونه کد این بخش اعداد ۱۰۶۷۷۶۸۳/۰ برای میانگین تعداد روابط با خاصیت تراگذری و عدد ۱۵۹۸۸۶۲/۳۳۳۳۳۳۳۳۳ برای میانگین تعداد روابط با خاصیت رقابت بدست آمد.

۳.۴ پرسش تئوری ۷

t را برابر تعداد حالاتی که یک مثلث (سه راس) از راس‌های گراف انتخاب کنیم تعریف می‌کنیم:

$$t \equiv \binom{n}{3} \longrightarrow \text{تعداد انتخاب‌های مثلث از گراف}$$

همچنین X را به صورت زیر تعریف می کنیم :

$$X_j = \begin{cases} 1 & \text{سه رأس انتخابی تشکیل مثلث دهند} \\ 0 & \text{سه رأس انتخابی تشکیل مثلث ندهند} \end{cases} \quad (1 \leq j \leq t)$$

حالا به محاسبه‌ی امید ریاضی تعداد دوستی‌هایی که خاصیت تراگذاری دارند می‌پردازیم:

$$\mathbb{E}[\text{تعداد دوستی با خاصیت تراگذاری}] = \mathbb{E}[\sum_{j=1}^{t} X_j] \xrightarrow{\text{خطی بودن } \mathbb{E}} \sum_{j=1}^{t} \mathbb{E}[X_j] = tp^3 = \binom{n}{3} p^3$$

حالا به محاسبه‌ی امید ریاضی دوستی‌هایی می‌پردازیم که خاصیت رقابت دارند. برای این کار ابتدا باید سه رأس را از بین رئوس گراف انتخاب کنیم. سپس از بین این سه رأس انتخابی، یک رأس را به عنوان رأس اصلی درنظر می‌گیریم که به هردو رأس دیگر متصل خواهد شد.
احتمال وجود یال؛ p و عدم وجود یال $1 - p$ است. طبق این توضیحات مقدار خواسته شده به صورت زیر به دست می‌آید:

$$\mathbb{E}[\text{تعداد دوستی با خاصیت رقابت}] = \binom{n}{3} \times \binom{3}{1} \times p^2(1 - p)$$

۴.۴ پرسش تئوری ۸

کسر خواسته شده را پیاده سازی می‌کنیم به طوری که صورت آن تعداد دوستی با خاصیت تراگذاری و مخرج آن مجموع تعداد دوستی‌های با خاصیت تراگذاری یا رقابت باشد:

$$\begin{aligned}
\frac{\mathbb{E}[\text{تراگذری}]}{\mathbb{E}[\text{تراگذری یا رقابت}]} &= \frac{\mathbb{E}[\text{تراگذری}]}{\mathbb{E}[\text{تراگذری}] + \mathbb{E}[\text{رقابت}]} \\
&= \frac{\binom{n}{3}p^3}{\binom{n}{3}p^3 + \binom{n}{3} \times 3 \times p^2(1-p)} \\
&= \frac{p}{p+3-3p} = \frac{p}{3-2p}
\end{aligned}$$

$$\begin{aligned}
\mathbb{E}[\text{تعداد دوستی با خاصیت تراگذری}] &= \binom{3000}{3} \times (10^{-2})^3 \\
&= \frac{3000! \times 10^{-6}}{3!2997!} = \frac{8000 \times 2999 \times 2998 \times 10^{-6}}{6} \\
&= 4495.501
\end{aligned}$$

$$\begin{aligned}
\mathbb{E}[\text{تعداد دوستی با خاصیت رقابت}] &= \binom{3000}{3} \times \binom{3}{1} \times (10^{-2})^2 \times 0.99 \\
&= \frac{3000 \times 2999 \times 2998 \times 3}{6} \times 10^{-4} \times 0.99 \\
&= 1335163.797
\end{aligned}$$

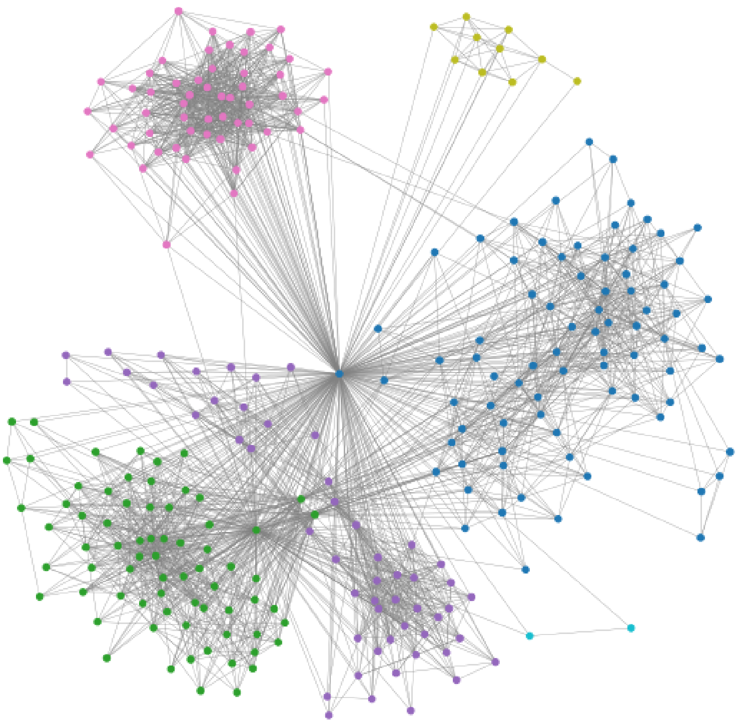
همانطور که مشاهده می‌شود، اعداد به دست آمده از محاسبات تئوری تقریبا برابر اعداد حاصل از شبیه سازی است که نشان دهنده‌ی صحت محاسبات و شبیه سازی است.

۵.۴ پرسش شبیه سازی ۵

میانگین تعداد روابط دوستی میان دوستان یک شخص در یک بار اجرای نمونه کد این بخش ۱۲ هزارم بدست آمد .

۶.۴ پرسش تئوری ۹

اگر با استفاده از دیتابیس شرکت متا که در دنیای واقعی رخ داده است میانگین را محاسبه کنیم نسبت به اینکه با شبیه سازی این کار را انجام دهیم، به عدد کمتری می‌رسیم.
دلیل این موضوع این است که در دنیای واقعی برقراری دوستی بین افراد، به فاکتورهایی مثل محل زندگی، سبک زندگی، ویژگی و خصوصیات افراد در تعامل با یکدیگر و... بستگی دارد. همان طور که در شکل هم مشخص است؛ دوستی‌ها در برخی نواحی تراکم بیشتر و در برخی نواحی تراکم کمتری دارد.
اما در شبیه سازی، همه ی افراد برای اینکه باهم رابطه‌ی دوستی برقرار کنند شرایط مشابه و یکسانی دارند.



همچنین اگر به شکل موجود در اول پروژه هم دقت کنیم می‌بینیم که شباهت زیادی به نقشه‌ی کره‌ی زمین دارد که تاییدی بر تاثیر محل زندگی و ... بر برقراری دوستی بین افراد است.

۷.۴ پرسش تئوری ۱۰

برای به دست آوردن تعداد روابط دوستی بین دوستان یک شخص اینگونه عمل می‌کنیم که ابتدا از بین همه‌ی افراد باقی مانده، دو نفر را انتخاب می‌کنیم و فرض می‌کنیم که هردوی این افراد با فرد اصلی دوست هستند. حالا عدد حاصل را در احتمال وجود رابطه‌ی دوستی بین آن دو شخص هم ضرب می کنیم تا تعداد روابط دوستی بین دوستان یک شخص به دست بیاید:

$$\binom{n-1}{2} \times p^3$$

۵ من از دیار حبیبم نه از بلاد غریب!

۱.۵ پرسش شبیه سازی ۶

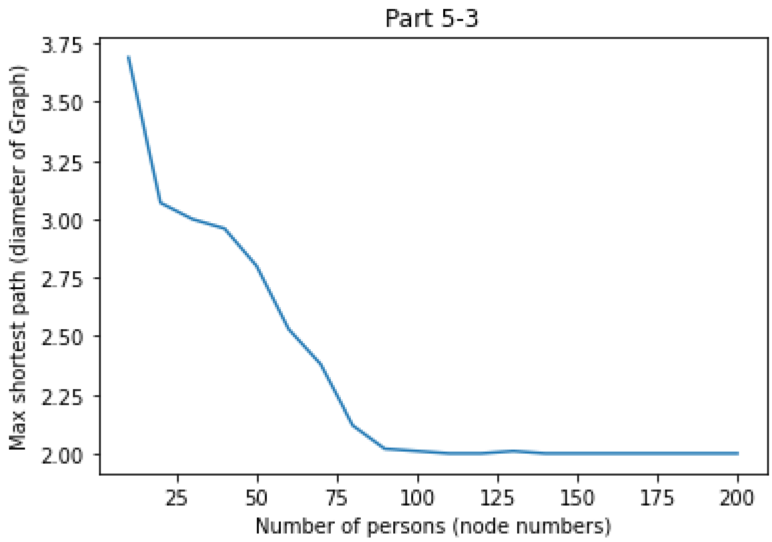
میانگین فاصله دو شخص در متاگراف با اجرای نمونه کد این بخش ۵/۷۶ بدست آمد.

۲.۵ پرسش شبیه سازی ۷

جواب شبیه سازی در این قسمت با اجرای نمونه کد این بخش ۲/۸۳ بدست آمد.

۳.۵ پرسش شبیه سازی ۸

با اجرای نمونه کد این بخش،نمودار تابع میانگین بیشترین فاصله در گراف بر حسب n به صورت (اکیداً) نزولی است و بعد از ۱۰۰ نمودار تقریبا به فرم نمودار ثابت در می‌آید و برابر با ۲ می‌شود. (به عبارت دیگر میانگین بیشترین فاصله بین گره‌های مختلف به ۲ میل می‌کند)



شکل ۲: شبیه سازی هشتم

۴.۵ پرسش تئوری ۱۱

می خواهیم احتمال این را به دست بیاوریم که دو رأس انتخابی همسایه‌ی مشترک نداشته باشند. برای این موضوع باید احتمال این را حساب کنیم که هیچ رأسی در گراف وجود نداشته باشد که همزمان به هردوی این رأس‌ها متصل باشد.
احتمال اتصال همزمان به این دو رأس برابر p^2 است. پس متمم آن می شود $1 - p^2$.
همچنین این موضوع برای همه‌ی رأس‌های دیگر در گراف هم باید برقرار باشد. پس آن را به توان $n-2$ می‌رسانیم:

$$\begin{aligned}\mathbb{P}[I_{u,v} = 1] &= \mathbb{P}[\text{همسایه مشترک نداشته باشند}] \\ &= (1 - \mathbb{P}[\text{یک همسایه مشترک داشته باشند}])^{n-2} \\ &= (1 - p^2)^{n-2}\end{aligned}$$

۵.۵ پرسش تئوری ۱۲

طبق فرمول داریم:

$$\begin{aligned}
\mathbb{E}[X_n] &= \mathbb{E}[\sum_{i=1}^{n} \sum_{j=i+1}^{n} I_{u,v}] \\
&= \sum_{i=1}^{n} \sum_{j=i+1}^{n} \mathbb{E}[I_{u,v}] \\
&= \binom{n}{2} \mathbb{P}[I_{u,v} = 1] \\
&= \binom{n}{2} (1-p^2)^{n-2}
\end{aligned}$$

۶.۵ پرسش تئوری ۱۳

نامساوی مارکف: اگر X یک متغیر تصادفی نامنفی و $a > 0$ باشد:

$$\mathbb{P}(X \geq a) \leq \frac{\mathbb{E}(X)}{a}$$

حالا طبق این نامساوی و با درنظر گرفتن $a = 1$ داریم:

$$\mathbb{P}[X_n \geq 1] \leq \frac{\mathbb{E}(X_n)}{1} \rightarrow \mathbb{P}[X_n \geq 1] \leq \binom{n}{2}[(1-p^2)^{n-2}]$$

سپس هردو عبارت را در $n \rightarrow \infty$ محاسبه می‌کنیم:

$$\lim_{n\rightarrow\infty} \mathbb{P}[X_n \geq 1] \leq \lim_{n\rightarrow\infty} \frac{n(n-1)}{2}(1-p^2)^{n-2}$$

حد عبارت سمت راست برابر ۰ است پس:

$$\lim_{n\rightarrow\infty} \mathbb{P}[X_n \geq 1] = 0 \rightarrow \lim_{n\rightarrow\infty} \mathbb{P}[X_n = 0] = 1$$

همان طور که انتظار می‌رفت با افزایش تعداد رئوس احتمال پیدا شدن حداقل یک جفت رأس که همسایه‌ی مشترک نداشته باشند تقریبا صفر می‌شود و قطعا حداقل یک جفت رأس یافت می‌شود که دارای یک همسایه‌ی مشترک باشد.

۷.۵ پرسش تئوری ۱۴

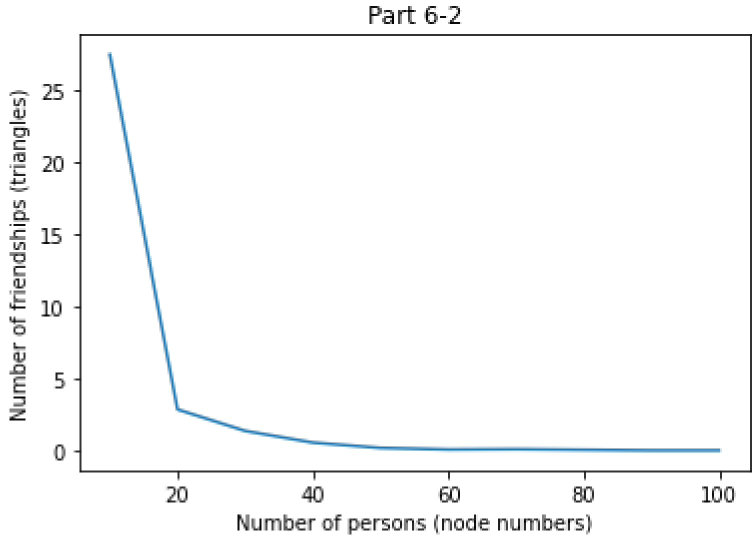
همانطور که در پرسش قبل نتیجه گرفتیم، با افزایش تعداد رئوس حتما حداقل یک جفت راس هستند که همسایه‌ی مشترک دارند. این به این معنی است که حتما هر دو رأس را انتخاب کنیم، یک همسایه‌ی مشترک دارد یعنی حداکثر با یک واسطه به هم می‌رسند. پس قطر گراف حداکثر برابر ۲ خواهد بود و به p وابسته نیست.
از عبارت حداکثر استفاده کردیم چون ممکن است دو رأسی که انتخاب می شود، با یک یال به هم وصل باشند. همچنین همانطور که در شبیه سازی هم دیدیم، جواب‌ها باهم تطابق داشت که نشان دهنده‌ی درستی محاسبات و شبیه سازی است.

۶ وَ إن یَکاد بخوانید!

۱.۶ پرسش شبیه سازی ۹

میانگین تعداد حلقه‌های دوستی، با اجرای نمونه کد این بخش ۶۴۰۶/۳ بدست آمد.

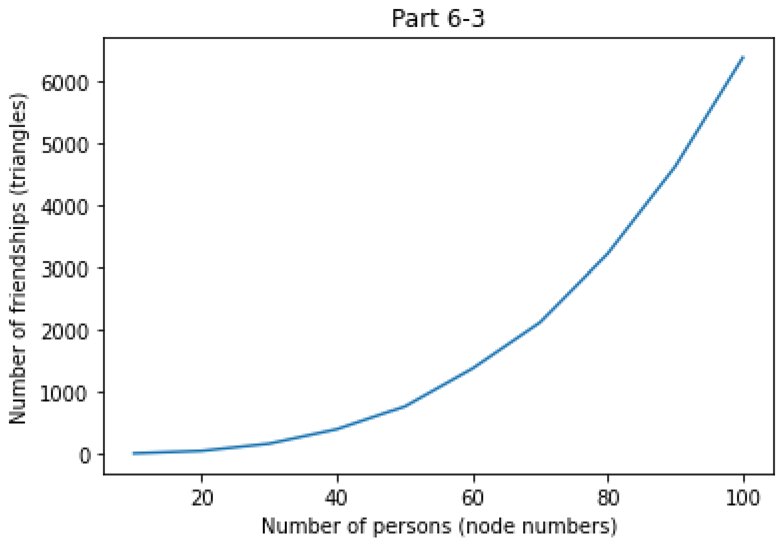
۲.۶ پرسش شبیه سازی ۱۰



شکل ۳: شبیه سازی دهم

با توجه به نمودار درصورتی که p از رابطه داده شده محاسبه شود، تعداد مثلث‌ها به صفر میل می‌کند. (کد این بخش)

۳.۶ پرسش شبیه سازی ۱۱

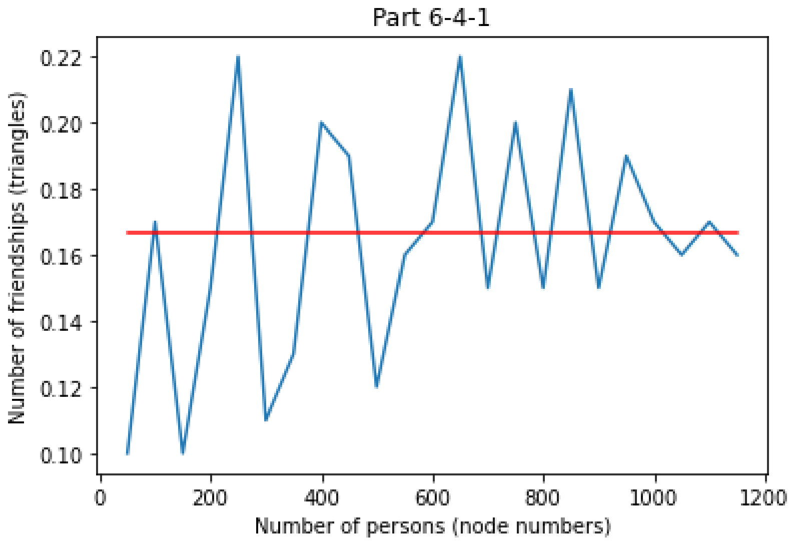


شکل ۴: شبیه سازی یازدهم

در این بخش چون p یک عدد ثابت است و تعداد دوستی‌ها با افزایش تعداد گره‌ها افزایش می‌یابد تعداد حلقه‌های دوستی همگرا نمی‌شوند و به صورت صعودی افزوده می‌شود که این موضوع از روی شکل نیز مشخص است. (کد این بخش)

۴.۶ پرسش شبیه سازی ۱۲

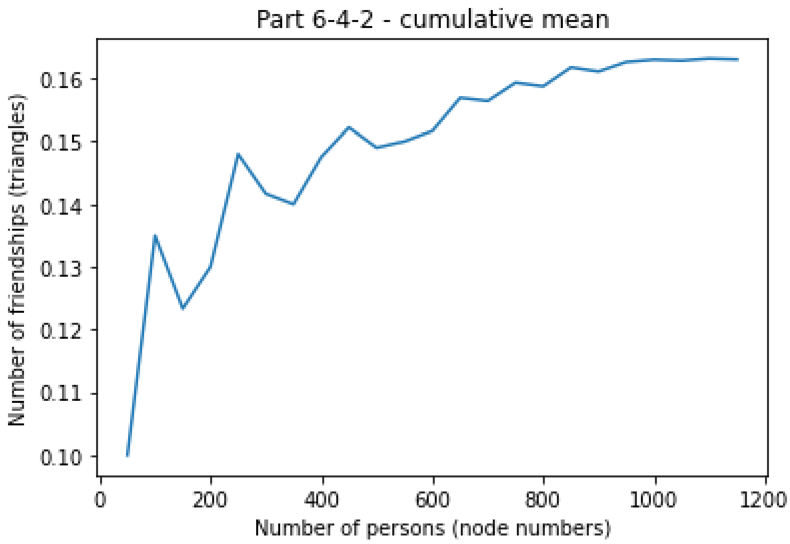
با اجرای نمونه کد این بخش نتیجه زیر بدست می‌آید:



شکل ۵: شبیه سازی دوازدهم

با توجه به نمودار، تعداد حلقه‌های دوستی در حال نوسان حول خطی (مقداری ثابت) هستند و این مقدار اینگونه بدست می‌آید:

$$\lim_{n\to\infty} \mathbb{E}[T_{3,n}|p = \frac{1}{n}] = \lim_{n\to\infty} \binom{n}{3} p^3 = \frac{1}{3!} \simeq 0.17$$



شکل ۶: شبیه سازی دوازدهم - نمودار میانگین تجمعی

نمودار میانگین تجمعی بدست آمده نیز، نتیجه بخش قبل را تأیید می‌کنند و نمودار در $n \to \infty$ به $\frac{1}{3!}$ میل می‌کند.

۵.۶ پرسش تئوری ۱۵

در اینجا می‌خواهیم احتمال اینکه سه رأس انتخابی، تشکیل حلقه بدهند را محاسبه کنیم.
برای اینکه به این هدف برسیم، باید هر سه یالی که می‌تواند بین آن‌ها وجود داشته باشد، موجود باشد.
احتمال وجود یال بین دو راس هم p است. پس احتمال خواسته شده برابر است با:

$$\mathbb{P}[I_{u,v,w} = 1] = p^3$$

۶.۶ پرسش تئوری ۱۶

طبق فرمول، احتمالی که در پرسش تئوری قبل محاسبه کردیم را در تعداد حالات انتخاب سه رأس از n رأس گراف، ضرب می‌کنیم:

$$\mathbb{E}[T_{3,n}] = \binom{n}{3} \times p^3$$

۷.۶ پرسش تئوری ۱۷

طبق نامساوی مارکف و قرار دادن $a = 1$ و همچنین نتیجه‌ای که از پرسش تئوری قبل به دست آوردیم داریم:

$$\mathbb{P}[T_{3,n} \geq 1] \leq \frac{\mathbb{E}[T_{3,n}]}{1} = \binom{n}{3} p^3$$

حالا همان طور که در سوال گفته شده به جای p عبارت $\frac{1}{n^2}$ را قرار می‌دهیم و سپس حد عبارت حاصل را در $n \to \infty$ محاسبه می‌کنیم:

$$\xrightarrow{p=\frac{1}{n^2}} \binom{n}{3} p^3 = \frac{n(n-1)(n-2)}{3!} \times \frac{1}{n^6} \xrightarrow{\lim_{n\to\infty}} 0$$
$$\longrightarrow \lim_{n\to\infty} \mathbb{P}[T_{3,n} \geq 1] = 0 \longrightarrow \lim_{n\to\infty} \mathbb{P}[T_{3,n} = 0] = 1$$

همان طور که دیدیم با افزایش تعداد رئوس احتمال اینکه حداقل یک سری سه تایی، رأس پیدا شوند که باهم تشکیل مثلث بدهند؛ نزدیک به صفر خواهد بود. یعنی تقریبا هیچ سری سه تایی ای رأس یافت نمی‌شود که باهم تشکیل مثلث دهند.

۸.۶ پرسش تئوری ۱۸

برای اثبات قضیه‌ی ذکر شده از نامساوی چبیشف استفاده می‌کنیم:

$$\mathbb{P}(|X - \mu| \geq k) \leq \frac{\sigma^2}{k^2}$$

که μ همان میانگین و σ^2 همان واریانس است.

حالا فرض می کنیم k برابر میانگین یا همان $\mathbb{E}[Y_n]$ است:

$$k = \mathbb{E}[Y_n] = \mu \longrightarrow \mathbb{P}(\underbrace{|Y_n - k| \geq k}_{Y_n \geq 2k \cup Y_n \leq 0}) \leq \frac{Var[Y_n]}{\mathbb{E}[Y_n]^2} \longrightarrow \mathbb{P}(Y_n \geq 2k \cup Y_n \leq 0) \leq \frac{Var[Y_n]}{\mathbb{E}[Y_n]^2}$$

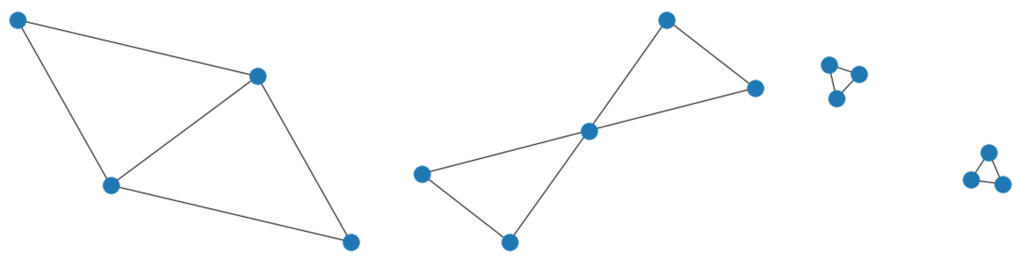
$$\longrightarrow \mathbb{P}(Y_n \geq 2k) + \mathbb{P}(Y_n \leq 0) \leq \frac{Var[Y_n]}{\mathbb{E}[Y_n]^2}$$

حالا حد هر دو طرف تساوی را در $n \to \infty$ محاسبه می‌کنیم. همان طور که در فرض داریم حد عبارت سمت راست برابر صفر است بنابراین:

$$\lim_{n\to\infty} (\mathbb{P}[Y_n \geq 2k] + \mathbb{P}[Y_n \leq 0]) = 0 \xrightarrow{\text{هر دو جمله نامنفی}} \lim_{n\to\infty} \mathbb{P}[Y_n \leq 0] = 0$$
$$\longrightarrow \lim_{n\to\infty} (1 - \mathbb{P}[Y_n \leq 0]) = 1 \to \lim_{n\to\infty} \mathbb{P}[Y_n \geq 1] = 1$$

۹.۶ پرسش تئوری ۱۹

برای ایجاد دو مثلث، سه حالت ایجاد می‌شود که به صورت زیر است:



شکل ۷: پرسش تئوری ۱۹

می دانیم امید ریاضی خواسته شده، برابر مجموع امید ریاضی‌های این سه حالت است:

$$\mathbb{E}[I_{u,v,w} I_{u',v',w'}] = \mathbb{E}_1 + \mathbb{E}_2 + \mathbb{E}_3$$

حالا به محاسبه‌ی هریک از آن‌ها می‌پردازیم:

در حالت اول ابتدا از بین همه‌ی رئوس ۶ رأس مورد نیاز را انتخاب می‌کنیم. سپس باید آن‌ها را در دو گروه سه تایی قرار دهیم که تعداد حالات ممکن برای انجام این کار برابر $\frac{\binom{6}{3}\binom{3}{3}}{2!}$ است. درنهایت هم به ازای شش یال موجود، عبارت را در p^6 ضرب می‌کنیم.

$$\mathbb{E}_1 = \binom{n}{6} \frac{\binom{6}{3}\binom{3}{3}}{2!} \times p^3 \times p^3 = 10p^6 \times \binom{n}{6}$$

درحالت دوم ابتدا باید ۵ رأس مورد نیاز را از بین n رأس انتخاب کنیم. سپس به پنج حالت مختلف، رأسی که در مرکز قرار گرفته و به تمام بقیه‌ی رئوس انتخابی متصل می شود را انتخاب می‌کنیم. حالا رئوس باقی مانده را به دو گروه دوتایی تقسیم کرده که تعداد حالات انجام این کار برابر است با $\frac{\binom{4}{2}\binom{2}{2}}{2!}$ و درآخر عبارت در p^6 ضرب می‌شود.

$$\mathbb{E}_2 = \binom{n}{5}\binom{5}{1}\frac{\binom{4}{2}\binom{2}{2}}{2!} \times p^6 = 15p^6 \times \binom{n}{5}$$

در حالت سوم، به چهار رأس احتیاج داریم. سپس از بین رئوس انتخابی دو رأس را انتخاب می‌کنیم تا یال مشترک بین آن دو ایجاد شود. در نهایت هم عبارت را در p^5 ضرب می کنیم چون درکل ۵ یال وجود دارد.

$$\mathbb{E}_3 = \binom{n}{4}\binom{4}{2} \times p^5 = 6p^5 \times \binom{n}{4}$$

در نتیجه داریم:

$$\rightarrow \mathbb{E}[I_{u,v,w}I_{u',v',w'}] = \binom{n}{6}10p^6 + \binom{n}{5}15p^6 + \binom{n}{4}6p^5$$

۱۰.۶ پرسش تئوری ۲۰

برای محاسبه‌ی عبارت خواسته شده طبق فرمول داریم:

$$\begin{aligned}\mathbb{E}[T_{3,n}^2] &= \mathbb{E}[(\sum_{k=1}^{\binom{n}{3}} I_{u,v,w})^2] \\ &= \sum_{k=1}^{\binom{n}{3}} \mathbb{E}[I_{u,v,w}^2] + 2\sum_{k=1}^{\binom{n}{3}} \mathbb{E}[I_{u,v,w}I_{u',v',w'}]\end{aligned}$$

که با توجه نتایج پرسش‌های تئوری قبلی، برابر است با:

$$\binom{n}{3}p^3 + 2[\binom{n}{6}10p^6 + \binom{n}{5}15p^6 + \binom{n}{4}6p^5] = [20\binom{n}{6} + 30\binom{n}{5}]p^6 + 12\binom{n}{4}p^5 + \binom{n}{3}p^3$$

۱۱.۶ پرسش تئوری ۲۱

می‌دانیم:

$$Var(T_{3,n}) = \mathbb{E}(T_{3,n}^2) - (\mathbb{E}(T_{3,n}))^2$$

جمله‌ی اول در عبارت بالا همان مقدار به دست آمده در پرسش تئوری ۲۰ است. همچنین جمله‌ی دوم نیز طبق پرسش‌های تئوری قبلی برابر $(\binom{n}{3}p^3)^2$ است. بنابراین با جایگذاری c به جای p داریم:

$$20\binom{n}{6}c^6 + 30\binom{n}{5}c^6 + 12\binom{n}{4}c^5 + \binom{n}{3}c^3 - \binom{n}{3}\binom{n}{3}c^6 =$$
$$\frac{20 \times n!c^6}{6!(n-6)!} + \frac{30 \times n!c^6}{5!(n-5)!} + \frac{12 \times n!c^5}{4!(n-4)!} + \frac{n!c^3}{3!(n-3)!} - \frac{n!}{3!(n-3)!} \times \frac{n!c^6}{3!(n-3)!} =$$
$$\frac{c^6}{36}n(n-1)(n-2)(n-3)(n-4)(n-5) + \frac{c^6}{40}n(n-1)(n-2)(n-3)(n-4)$$
$$+\frac{c^5}{2}n(n-1)(n-2)(n-3) + \frac{c^3}{6}n(n-1)(n-2) - \frac{1}{36}c^6n^2(n-1)^2(n-2)^2$$

حالا با استفاده از مقادیر به دست آمده و هم ارزی‌ها در بینهایت، کسر زیر را پیاده سازی کرده و آن را در $n \to \infty$ محاسبه می‌کنیم:

$$\lim_{n\to\infty} \frac{Var[T_{3,n}]}{\mathbb{E}^2[T_{3,n}]} = \lim_{n\to\infty} \frac{\frac{3n^5c^6+60n^4c^5+20n^3c^3}{120}}{\frac{n^6c^6}{36}} = 0$$

۱۲.۶ پرسش تئوری ۲۲

طبق نامساوی چبیشف داریم:

$$\mathbb{P}(|T_{3,n} - \mu| \geq k) \leq \frac{\sigma^2}{k^2}$$

فرض می‌کنیم k برابر میانگین یا همان $\mathbb{E}[T_{3,n}]$ باشد. بنابراین داریم:

$$\mathbb{P}(T_{3,n} \geq 2k) + \mathbb{P}(T_{3,n} \leq 0) \leq \frac{Var[T_{3,n}]}{\mathbb{E}^2[T_{3,n}]}$$

حالا عبارات را در $n \to \infty$ محاسبه می‌کنیم تا به رابطه‌ی زیر برسیم:

$$\lim_{n\to\infty}[\mathbb{P}(T_{3,n} \geq 2k) + \mathbb{P}(T_{3,n} \leq 0)] \leq 0 \to \lim_{n\to\infty}[\mathbb{P}(T_{3,n} \geq 2k) + \mathbb{P}(T_{3,n} \leq 0)] = 0$$
$$\to \lim_{n\to\infty}[P(T_{3,n} \leq 0)] = 0 \to \lim_{n\to\infty} \mathbb{P}[T_{3,n} \geq 1] = 1$$

این موضوع نشان می‌دهد زمانی که تعداد رئوس خیلی زیاد باشد، با احتمال بسیار بالایی حتما سه رأس وجود دارد که تشکیل مثلث دهد.

۱۳.۶ پرسش تئوری ۲۳

$\mathbb{I}$)ابتدا اثبات می‌کنیم که اگر X از توزیع پواسون پیروی کند، عبارت ذکر شده صحیح است:

$$\mathbb{P}[X=k]=\frac{e^{-\lambda}\lambda^k}{k!}\rightarrow \mathbb{E}[X_r]=\frac{\partial^r}{\partial t^r}M_X(t)|_{t=1}\rightarrow \frac{\partial^r}{\partial t^r}\mathbb{E}[t^X]|_{t=1}$$

$$\rightarrow \tfrac{\partial^r}{\partial t^r}[\textstyle\sum_{k=0}^{\infty} t^k\mathbb{P}(X=k)]|_{t=1}=\tfrac{\partial^r}{\partial t^r}[e^{-\lambda}\textstyle\sum_{k=0}^{\infty}\frac{(t\lambda)^k}{k!}]|_{t=1}=\tfrac{\partial^r}{\partial t^r}e^{t\lambda-\lambda}|_{t=1}=\lambda^r e^{t\lambda-\lambda}|_{t=1}=\lambda^r$$

$\mathbb{II}$)حالا عکس قضیه را اثبات می‌کنیم. یعنی اول فرض می‌کنیم عبارت داده شده صحیح است و سپس نتیجه می‌گیریم که متغیر X از توزیع پواسون پیروی می‌کند:

$$\mathbb{E}[X_r]=\lambda^r\rightarrow\frac{\partial^r}{\partial t^r}M_X(t)|_{t=1}=\lambda^r\rightarrow\frac{\partial^r}{\partial t^r}\mathbb{E}[t^X]|_{t=1}=\lambda^r\rightarrow\frac{\partial^r}{\partial t^r}[\sum_{k=0}^{\infty}t^k\mathbb{P}(X=k)]|_{t=1}=\lambda^r$$

$$\xrightarrow[\times e^{\lambda t-t}]{\overbrace{\int.....\int}^{r}\overbrace{dt.....dt}^{r}} e^{\lambda t-\lambda}=\sum_{k=0}^{\infty}t^k\mathbb{P}(X=k)\rightarrow\sum_{k=0}^{\infty}\frac{(\lambda t)^k}{k!}e^{-\lambda}=\sum_{k=0}^{\infty}t^k\mathbb{P}(X=k)$$

$$\rightarrow\sum_{k=0}^{\infty}\frac{(\lambda t)^k}{k!}=\sum_{k=0}^{\infty}e^{\lambda}t^k\mathbb{P}(X=k)$$

$$\rightarrow e^{\lambda}t^k\mathbb{P}(X=k)=\frac{(\lambda t)^k}{k!}\rightarrow\mathbb{P}(X=k)=\frac{e^{-\lambda}\lambda^k}{k!}\rightarrow X\sim Poisson(\lambda)$$

حالا طبق $\mathbb{I}$ و $\mathbb{II}$ نتیجه می‌گیریم که:

شرط لازم و کافی برای آن که متغیر تصادفی X، یک متغییر تصادفی پواسون با پارامتر λ باشد، آن است که داشته باشیم $\mathbb{E}[(X)_r]=\lambda^r$.

۱۴.۶ پرسش تئوری ۲۴

طبق پرسش‌های تئوری قبلی عبارات خواسته شده را قبلا به دست آوردیم. در اینجا تنها کافیست به جای p، $\frac{c}{n}$ قرار داده و عبارات را در $n\rightarrow\infty$ محاسبه کنیم:

$$\mathbb{E}[T_{3,n}]=\binom{n}{3}p^3\xrightarrow[\binom{n}{3}\overset{n\rightarrow\infty}{\sim}\frac{n^3}{6}]{p=\frac{c}{n},n\rightarrow\infty}\frac{n^3}{6}\times\frac{c^3}{n^3}=\frac{c^3}{6}\rightarrow\lambda$$

$$\mathbb{E}[(T_{3,n})_2]=\mathbb{E}[T_{3,n}(T_{3,n}-1)]=\mathbb{E}[T_{3,n}^2-T_{3,n}]=\mathbb{E}[T_{3,n}^2]-\mathbb{E}[T_{3,n}]$$

$$=20\binom{n}{6}p^6+30\binom{n}{5}p^6+12\binom{n}{4}p^5+\binom{n}{3}p^3-\binom{n}{3}p^3$$

$$\xrightarrow{p=\frac{c}{n},n\rightarrow\infty}\frac{c^6n^6}{36n^6}+\frac{c^6n^5}{40n^6}+\frac{c^5n^4}{2n^5}\sim\frac{c^6}{36}\rightarrow\lambda^2$$

۱۵.۶ پرسش تئوری ۲۵

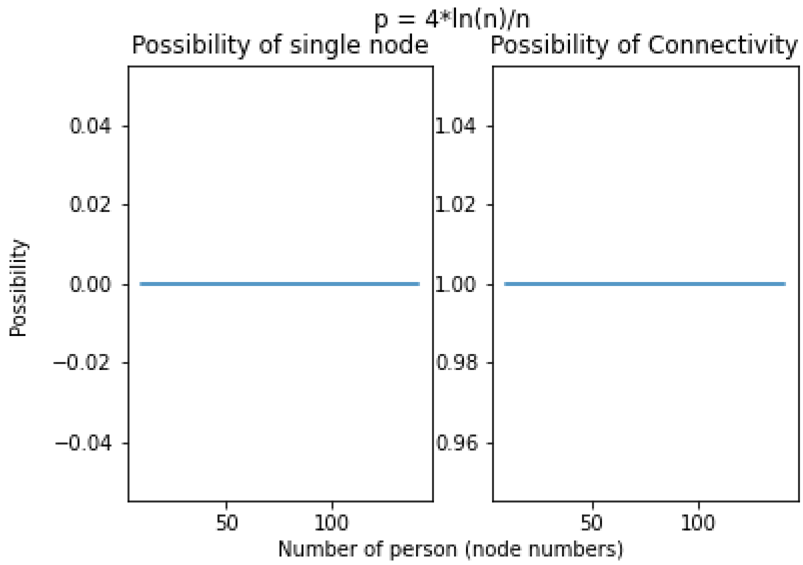
۷ قومی به جد و جهد نهادند وصل دوست، قومی دگر حواله به تقدیر می‌کنند!

۱.۷ پرسش شبیه سازی ۱۳

در اجرای نمونه کد این بخش احتمال همبند بودن ۰ و احتمال اینکه فردی بدون دوست پیدا شود نیز ۱ بدست آمد.

۲.۷ پرسش شبیه سازی ۱۴

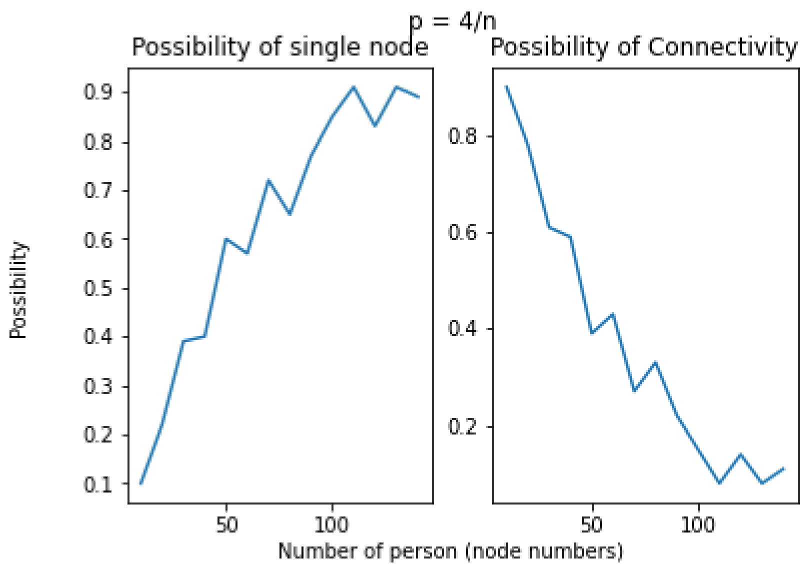
با اجرای نمونه کد این بخش نتیجه زیر بدست می‌آید:



شکل ۸: شبیه سازی چهاردهم

۳.۷ پرسش شبیه سازی ۱۵

با اجرای نمونه کد این بخش نتیجه زیر بدست می‌آید:



شکل ۹: شبیه سازی پانزدهم

۴.۷ پرسش تئوری ۲۶

گراف را به دو بخش با j رأس و $n-j$ رأس تقسیم می‌کنیم. برای اینکه بخش با j رأس جدا از بخش با $n-j$ رأس باشد، باید بین هیچ یک از رأس‌های دو بخش، یالی وجود نداشته باشد.
پس احتمال عدم وجود یال که $1-p$ بود را به توان $j(n-j)$ می‌رسانیم.
همچنین بخش با j رأس باید همبند باشد. یعنی به تعداد یال‌های موجود در آن باید عددمان در p ضرب شود. بنابراین احتمال خواسته شده به صورت زیر است:

$$\mathbb{P}[K_i^j = 1] = p^\alpha (1-p)^{j(n-j)}$$

می‌دانیم که p عددی بین ۰ و ۱ است. پس وقتی به توان α که عددی مثبت است می‌رسد، بین ۰ ، ۱ قرار می‌گیرد:

$$\mathbb{P}[K_i^j = 1] \leq (1-p)^{j(n-j)}$$

۵.۷ پرسش تئوری ۲۷

طبق فرمول داریم:

$$\mathbb{E}[X_j] = \sum_{k=1}^{\binom{n}{j}} \mathbb{E}[K_i^j] \leq \binom{n}{j} (1-p)^{j(n-j)}$$

۶.۷ پرسش تئوری ۲۸

طبق فرمول داریم:

$$\mathbb{E}[X] = \mathbb{E}[\sum_{j=1}^{n-1} X_j] = \sum_{j=1}^{n-1} \mathbb{E}[X_j] \leq \sum_{j=1}^{n-1} [\binom{n}{j} (1-p)^{j(n-j)}] \leq \sum_{j=1}^{n-1} \frac{1}{n^{1.7}} = (n-1)n^{-1.7} \xrightarrow{n\to\infty} 0$$

همانطور که مشاهده کردیم، با افزایش تعداد رئوس تعداد بخش‌های گراف کم می‌شود و به صفر میل می کند.

۷.۷ پرسش تئوری ۲۹

طبق نامساوی مارکف داریم:

$$\mathbb{P}[X \geq 1] \leq \frac{\mathbb{E}[X]}{1} \to \lim_{n\to\infty} \mathbb{P}[X \geq 1] \leq \lim_{n\to\infty} \mathbb{E}[X] = 0 \to \lim_{n\to\infty} \mathbb{P}[X \geq 1] = 0 \to \lim_{n\to\infty} \underbrace{\mathbb{P}[X = 0]}_{g(n,p)\ \text{همبند بودن}} = 1$$

۸ کد سوالات شبیه سازی

۱.۸ شبیه سازی ۱

(قطعه کد ۲):

```
import networkx
import numpy

n = 1000
m = 3000
p = 0.0034

friendships = []

nodes = numpy.arange(n)

for t in range(0,10):

    G = networkx.Graph()
    G.add_nodes_from(nodes)

    for i in range(0,n):
        for j in range(i+1,n):
            s = numpy.random.choice([0,1], p=[1-0.0034, 0.0034])

            if s == 1:
               G.add_edges_from([(nodes[i],nodes[j])])

    friendships.append(G.number_of_edges())

    G.clear()

m_simulation = numpy.mean(friendships)
print(m_simulation)
```

۲.۸ شبیه سازی ۲

(قطعه کد ۳):

```
import matplotlib.pyplot as plt
import networkx as nx

n = 1000
p = 0.00016
L = []
n_plot = 10
friends_n = list(range(0,n_plot))
friendship_data = []
social_person_n = []
n_loop = 10

for i in range (0,n_loop):
  haveFriend = [0]*n_plot
  G = make_random_graph(n, p)
  L.append(nx.Graph.number_of_edges(G)*2/n)
  social_person_sum = 0

  for j in range (0,n):
    for k in range(0,len(friends_n)):
      if(G.degree(j)==k):
        haveFriend[k]+=1
    if(G.degree(j)>L[i]):
      social_person_sum += 1
  social_person_n.append(social_person_sum)
  friendship_data.append(haveFriend)
print(average(social_person_n))

ploting_data= [];
for i in range (0,n_plot):
  temp = []
  for j in range (0,n_loop):
    temp.append(friendship_data[j][i])
  ploting_data.append(average(temp))

plt.plot(friends_n, ploting_data)
plt.xlabel('Number of friends')
plt.ylabel('Number of person with this number of friend')
plt.title('Part 3-1')
plt.show()
```

۳.۸ شبیه سازی ۳

(قطعه کد ۴):

```
import networkx as nx

n = 3000
p = 0.01
loop_n = 5
transitives = []
competitions = []

for i in range (0,loop_n):
  transitive = 0
  competition = 0
  processed = []
  random_graph = make_random_graph(n, p)

  for j in range (0,n):
    transitive += nx.triangles(random_graph,j)

  dictionary = dict(nx.shortest_path_length(random_graph))
  for k in range (0,len(dictionary)):
    res = sum(x == 2 for x in dictionary[k].values())
    competition += res/2

  transitives.append((transitive/3))
  competitions.append(competition)

print(average(transitives))
print(average(competitions))
```

۴.۸ شبیه سازی ۴

(قطعه کد ۵):

```
import networkx as nx

n = 2000
p = 0.2
loop_n = 3
transitives = []
competitions = []

for i in range (0,loop_n):
  transitive = 0
  competition = 0
  processed = []
  random_graph = make_random_graph(n, p)

  for j in range (0,n):
    transitive += nx.triangles(random_graph,j)

  dictionary = dict(nx.shortest_path_length(random_graph))
  for k in range (0,len(dictionary)):
    res = sum(x == 2 for x in dictionary[k].values())
    competition += res/2

  transitives.append((transitive/3))
  competitions.append(competition)

print(average(transitives))
print(average(competitions))
```

۵.۸ شبیه سازی ۵

(قطعه کد ۶):

```
import networkx as nx

n = 1000
p = 0.003
sum_of_friendship = 0

G = make_random_graph(n, p)

for i in range(0,n):
  sum_of_friendship += G.subgraph(G.adj[i]).number_of_edges()

print(sum_of_friendship/n)
```

۶.۸ شبیه سازی ۶

(قطعه کد ۷):

```
import networkx as nx

n = 1000
p = 0.0033

G = make_random_graph(n, p)

sum = 0
num = 0

dictionary = dict(nx.shortest_path_length(G))
for i in range (0,len(dictionary)):
  for j in range (0,len(list(dictionary[i].values()))):
    sum += list(dictionary[i].values())[j]
  num += len(dictionary[i])

print(sum/num)
```

۷.۸ شبیه سازی ۷

(قطعه کد ۸):

```
import networkx as nx

n = 50
p = 0.34
loop_n = 100
data = []

for i in range (0,loop_n):
  G = make_random_graph(n, p)
  if(nx.is_connected(G) == False):
      for j in range(0,n):
        if(G.degree(j)==0):
          G.remove_node(j)
  data.append(nx.algorithms.distance_measures.diameter(G))

print(average(data))
```

۸.۸ شبیه سازی ۸

(قطعه کد ۹):

```
import networkx as nx
import matplotlib.pyplot as plt

n_s = 10
p = 0.34
loop_n = 100
step = 10
n_cycle = 20

x = []
y = []

for i in range (0,n_cycle):
  x.append(n_s)
  data = []
  for j in range (0,loop_n):
    G = make_random_graph(n_s, p)
    G = G.subgraph(max(nx.connected_components(G), key=len))
    data.append(nx.algorithms.distance_measures.diameter(G))
  y.append(average(data))
  n_s += step

plt.plot(x, y)
plt.xlabel('Number of persons (node numbers)')
plt.ylabel('Max shortest path (diameter of Graph)')
plt.title('Part 5-3')
plt.show()
```

۹.۸ شبیه سازی ۹

(قطعه کد ۱۰):

```
import networkx as nx

n = 100
p = 0.34
loop_n = 100
transitives = []

for i in range (0,loop_n):
  random_graph = make_random_graph(n, p)
  transitive = nx.triangles(random_graph)
  transitives.append((sum(transitive.values())/3))

print(average(transitives))
```

۱۰.۸ شبیه سازی ۱۰

(قطعه کد ۱۱):

```
import networkx as nx
import matplotlib.pyplot as plt

n_s = 10
loop_n = 100
step = 10
n_cycle = 10

x = []
y = []

for i in range (0,n_cycle):
  x.append(n_s)
  transitives = []
  p = 60/n_s/n_s
  for j in range (0,loop_n):
    random_graph = make_random_graph(n_s, p)
    transitive = nx.triangles(random_graph)
    transitives.append((sum(transitive.values())/3))
  n_s += step
  y.append(average(transitives))

plt.plot(x, y)
plt.xlabel('Number of persons (node numbers)')
plt.ylabel('Number of friendships (triangles)')
plt.title('Part 6-2')
plt.show()
```

۱۱.۸ شبیه سازی ۱۱

(قطعه کد ۱۲):

```
import networkx as nx
import matplotlib.pyplot as plt

n_s = 10
loop_n = 100
step = 10
n_cycle = 10
p = 0.34

x = []
y = []

for i in range (0,n_cycle):
  x.append(n_s)
  transitives = []
  for j in range (0,loop_n):
    random_graph = make_random_graph(n_s, p)
    transitive = nx.triangles(random_graph)
    transitives.append((sum(transitive.values())/3))
  n_s += step
  y.append(average(transitives))

plt.plot(x, y)
plt.xlabel('Number of persons (node numbers)')
plt.ylabel('Number of friendships (triangles)')
plt.title('Part 6-3')
plt.show()
```

۱۲.۸ شبیه سازی ۱۲

(قطعه کد ۱۳):

```
import networkx as nx
import matplotlib.pyplot as plt

n_s = 50
loop_n = 100
step = 50
n_cycle = 23

transitives = []

x = []
y = []
const = []
yy = []

for i in range (0,n_cycle):
  x.append(n_s)
  p = 1/n_s
  transitives.clear()
  for j in range (0,loop_n):
    random_graph = make_random_graph(n_s, p)
    transitive = nx.triangles(random_graph)
    transitives.append((sum(transitive.values())/3))
  n_s += step
  y.append(average(transitives))
  const.append(1/6)
  yy.append(average(y))

plt.plot(x, y)
plt.plot(x, const,'r-')
plt.xlabel('Number of persons (node numbers)')
plt.ylabel('Number of friendships (triangles)')
plt.title('Part 6-4-1')
plt.show()

plt.plot(x, yy)
plt.xlabel('Number of persons (node numbers)')
plt.ylabel('Number of friendships (triangles)')
plt.title('Part 6-4-2 - cumulative mean')
plt.show()
```

۱۳.۸ شبیه سازی ۱۳

(قطعه کد ۱۴):

```
import networkx as nx

n = 150
p = 0.2
loop_n = 100
connected = 0
single_node = 0


for i in range (0,loop_n):
  G = make_random_graph(n, p)

  m = 0

  if(nx.is_connected(G) == True):
    connected += 1

  for j in range (0,n):
    if(G.degree(j)==0):
      m += 1

  if(m != 0):
    single_node += 1

print(Possibility of single node : +str(single_node/loop_n))
print(Possibility of Connectivity : +str(connected/loop_n))
```

۱۴.۸ شبیه سازی ۱۴

(قطعه کد ۱۵):

```
import networkx as nx
import numpy as np
import matplotlib.pyplot as plt

n_s = 10
loop_n = 100
n_cycle = 14

x = []
y = []
yy = []

for p in range (0,n_cycle):
  x.append(n_s)
  connected = 0
  single_node = 0
  p = 4*np.log(n_s)/n_s
  for i in range (0,loop_n):
    G = make_random_graph(n_s, p)
    m = 0
    if(nx.is_connected(G) == True):
      connected += 1

    for j in range (0,n_s):
      if(G.degree(j)==0):
        m += 1

    if(m != 0):
      single_node += 1

  y.append(single_node/loop_n)
  yy.append(connected/loop_n)
  n_s += 10

fig, (ax1, ax2) = plt.subplots(1, 2)
ax1.plot(x, y)
ax2.plot(x, yy)
ax1.set_title('Possibility of single node')
ax2.set_title('Possibility of Connectivity')
fig.text(0.5, 0.04, 'Number of person (node numbers)', ha='center', va='center')
fig.text(0.001, 0.5, 'Possibility', ha='center', va='center', rotation='vertical')
fig.suptitle('p = 4*ln(n)/n')
plt.show()
```

۱۵.۸ شبیه سازی ۱۵

(قطعه کد ۱۶):

```
import networkx as nx
import numpy as np
import matplotlib.pyplot as plt

n_s = 10
loop_n = 100
n_cycle = 14

x = []
y = []
yy = []

for p in range (0,n_cycle):
  x.append(n_s)
  connected = 0
  single_node = 0
  p = 4/n_s
  for i in range (0,loop_n):
    G = make_random_graph(n_s, p)
    m = 0
    if(nx.is_connected(G) == True):
      connected += 1

    for j in range (0,n_s):
      if(G.degree(j)==0):
        m += 1

    if(m != 0):
      single_node += 1

  y.append(single_node/loop_n)
  yy.append(connected/loop_n)
  n_s += 10

fig, (ax1, ax2) = plt.subplots(1, 2)
ax1.plot(x, y)
ax2.plot(x, yy)
ax1.set_title('Possibility of single node')
ax2.set_title('Possibility of Connectivity')
fig.text(0.5, 0.04, 'Number of person (node numbers)', ha='center', va='center')
fig.text(0.001, 0.5, 'Possibility', ha='center', va='center', rotation='vertical
    ')
fig.suptitle('p = 4/n')
plt.show()
```