به نام خدا

دانشگاه صنعتی امیرکبیر (پلی تکنیک تهران)

دانشکده مهندسی کامپیوتر

تحلیل شبکه‌های پیچیده

تمرین دوم

دانشجو: رضا ساجدی

۴۰۰۱۳۱۰۷۲

استاد: دکتر چهرقانی

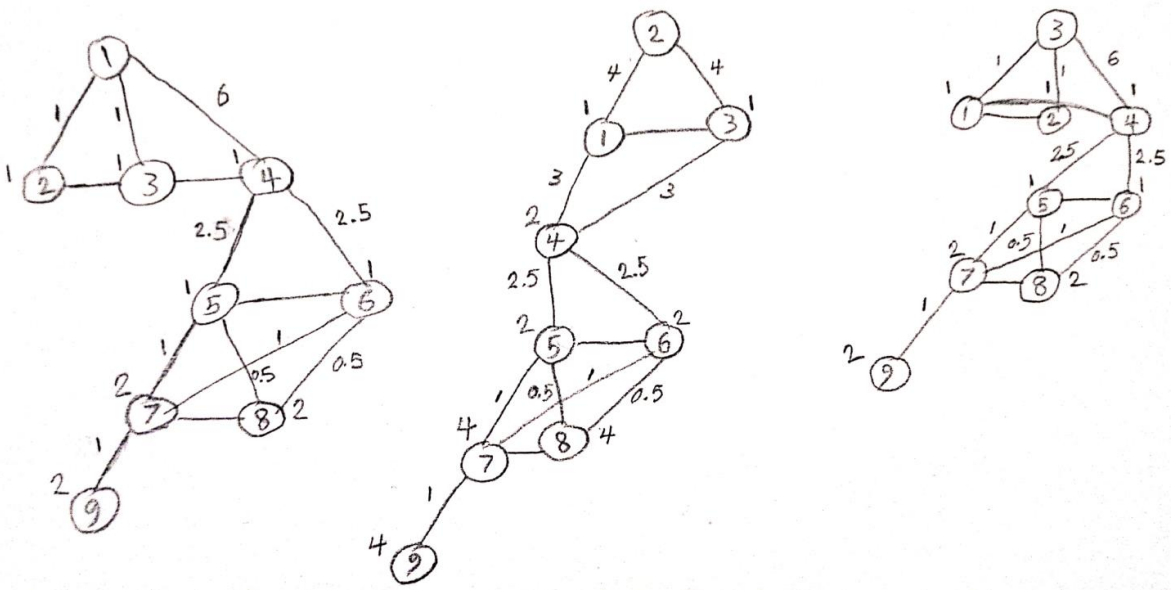
پاییز ۱۴۰۱

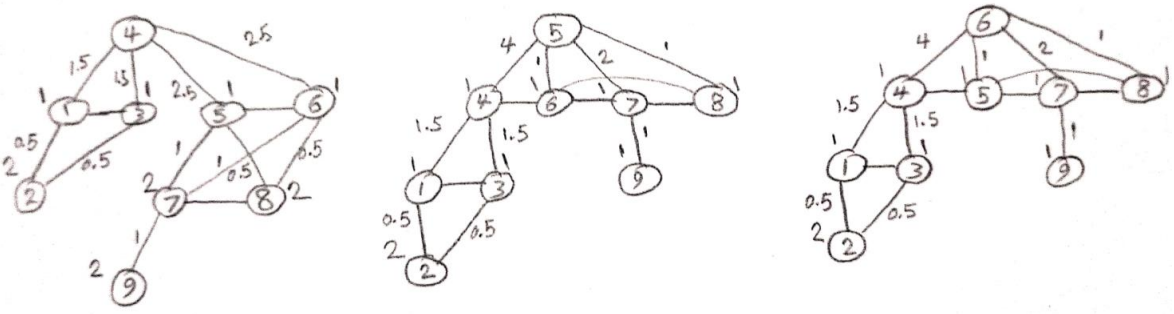
فهرست مطالب

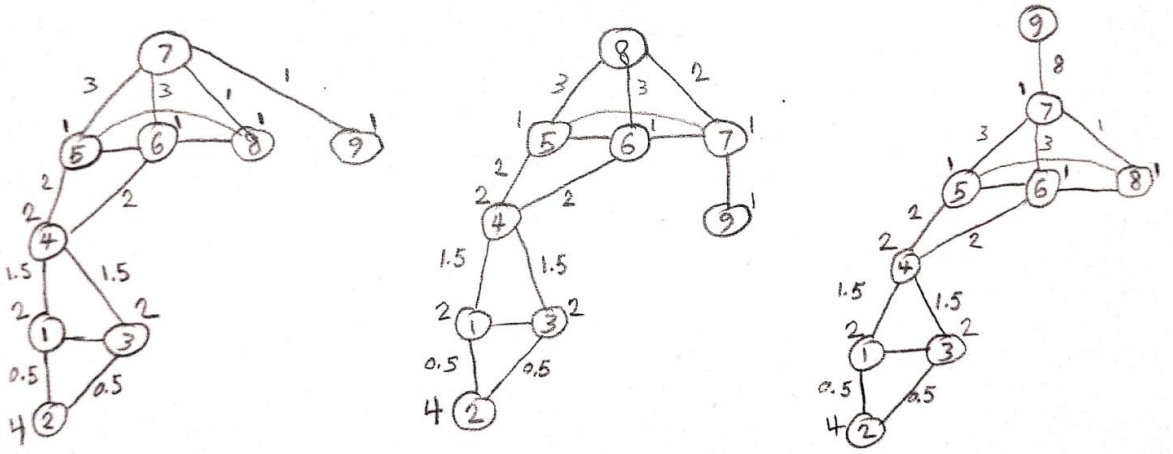
سؤال اول

قسمت الف

برای محاسبه EBC، از هر رأس یک بار BFS می‌زنیم و در مرحله Forward تعداد کوتاه‌ترین مسیرها از رأس شروع تا سایر رأس‌ها را شمارش می‌کنیم. در مرحله Backward نیز رأس‌ها را از آخر به اول پیمایش می‌کنیم و جریان‌هایی با اندازه یک از هر رأس عبور می‌دهیم تا جریان عبوری از هر یال مشخص شود. در نهایت مقادیر جریان در گراف‌های ایجاد شده را برای هر یال جمع می‌کنیم و مقدار حاصل، مرکزیت هر یال (تعداد کوتاه‌ترین مسیرهای عبوری از هر یال) را نشان می‌دهد.







(1,2): (1+4+0.5+0.5+0.5+0.5+0.5+0.5)/2 = 4

(1,3): (1+1)/2 = 1

(1,4): (6+3+1.5+1.5+1.5+1.5+1.5+1.5)/2 = 9

(2,3): (4+1+0.5+0.5+0.5+0.5+0.5+0.5)/2 = 4

(3,4): (3+6+1.5+1.5+1.5+1.5+1.5+1.5)/2 = 9

(4,5): (2.5+2.5+2.5+2.5+4+2+2+2)/2 = 10

(4,6): (2.5+2.5+2.5+2.5+4+2+2+2)/2 = 10

(5,7): (1+1+1+1+2+3+3)/2 = 6

(5,8): (0.5+0.5+0.5+0.5+1+3)/2 = 3

(5,6): (1+1)/2 = 1

(6,7): (1+1+1+1+2+3+3)/2 = 6

(6,8): (0.5+0.5+0.5+0.5+1+3)/2 = 3

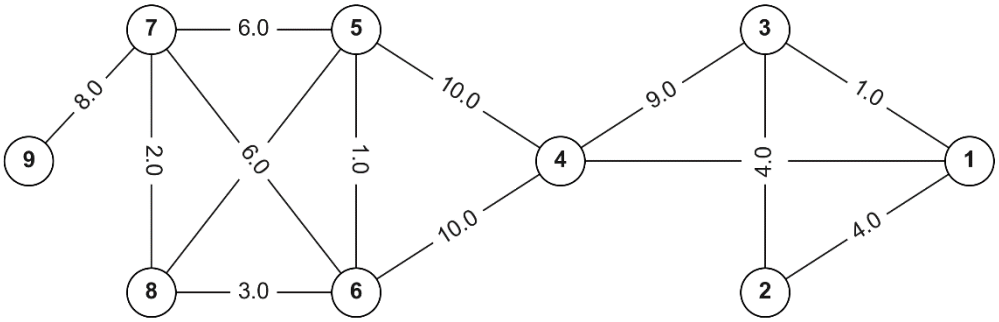
(7,9): (1+1+1+1+1+1+1+1+8)/2 = 8

(7,8): (1+2+1)/2 = 2

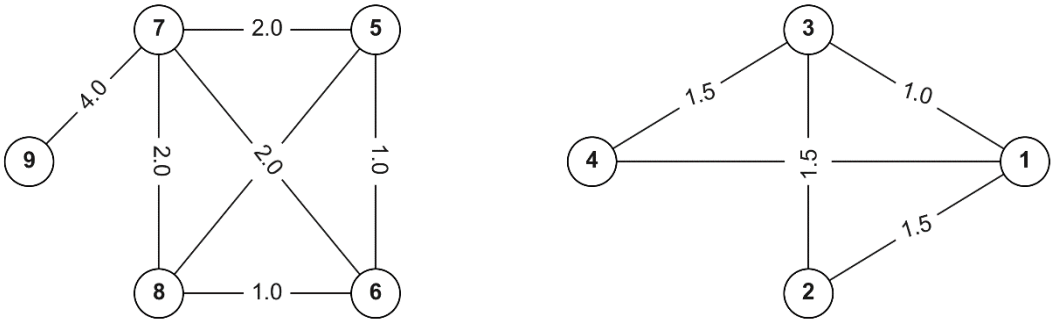
برای مشخص کردن Community های موجود در این گراف، مقدار آستانه EBC را برابر با ۱۰ درنظر گرفته‌ایم و برهمین اساس دو Community به‌دست می‌آید. این مورد، مشابه گام اول الگوریتم Girvan-Newman است که یال‌ها با بیشترین مرکزیت را حذف می‌کند.

برای اطمینان از صحت محاسبات، این فرآیند را با استفاده از کتابخانه NetworkX نیز انجام داده‌ایم و نتایج کاملاً یکسان و صحیح بوده است.

```
(1, 2): 4.0
(1, 3): 1.0
(1, 4): 9.0
(2, 3): 4.0
(3, 4): 9.0
(4, 5): 10.0
(4, 6): 10.0
(5, 7): 6.0
(5, 8): 3.0
(5, 6): 1.0
(6, 7): 6.0
(6, 8): 3.0
(7, 9): 8.0
(7, 8): 2.0
```

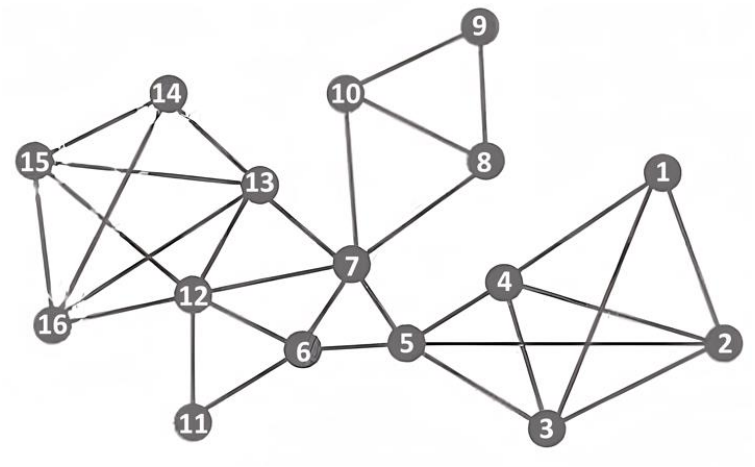


Step 1



قسمت ب

ابتدا رأس‌ها را شماره‌گذاری می‌کنیم:



حال همه کلیک‌های ماکزیمال موجود در این گراف را با به‌کارگیری الگوریتم بازگشتی پیدا کرده و آنها را نام‌گذاری می‌کنیم:

```
Q1 = {1, 2, 3, 4}
Q2 = {2, 5, 3, 4}
Q3 = {7, 8, 10}
Q4 = {7, 13, 12}
Q5 = {7, 6, 12}
Q6 = {7, 6, 5}
Q7 = {9, 8, 10}
Q8 = {11, 12, 6}
Q9 = {14, 16, 13, 15}
Q10 = {15, 16, 12, 13}
```

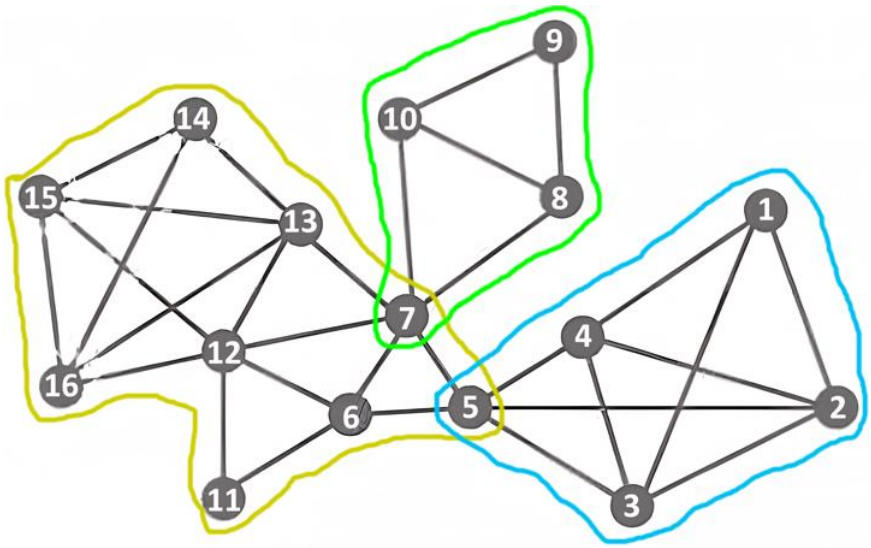
در مرحله بعد، ماتریس همپوشانی را برای کلیک‌های یافت شده محاسبه می‌کنیم. هر خانه از این ماتریس نشان می‌دهد که هر جفت کلیک، چه تعداد رأس مشترک دارند.

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
Q1	4	3	0	0	0	0	0	0	0	0
Q2	3	4	0	0	0	1	0	0	0	0
Q3	0	0	3	1	1	1	2	0	0	0
Q4	0	0	1	3	2	1	0	1	1	2
Q5	0	0	1	2	3	2	0	2	0	1
Q6	0	1	1	1	2	3	0	1	0	0
Q7	0	0	2	0	0	0	3	0	0	0
Q8	0	0	0	1	2	1	0	3	0	1
Q9	0	0	0	1	0	0	0	0	4	3
Q10	0	0	0	2	1	0	0	1	3	4

حال مقدار K را برابر با ۳ درنظر گرفته و ماتریس آستانه را محاسبه می‌کنیم. برای محاسبه ماتریس آستانه، خانه‌هایی که مقدارشان کوچکتر از K-1 است را صفر و سایر خانه‌ها را یک می‌کنیم. خانه‌هایی که مقدار یک دارد، نشان‌دهنده همجوار بودن جفت کلیک مربوط به آن است. (Adjacent k-cliques)

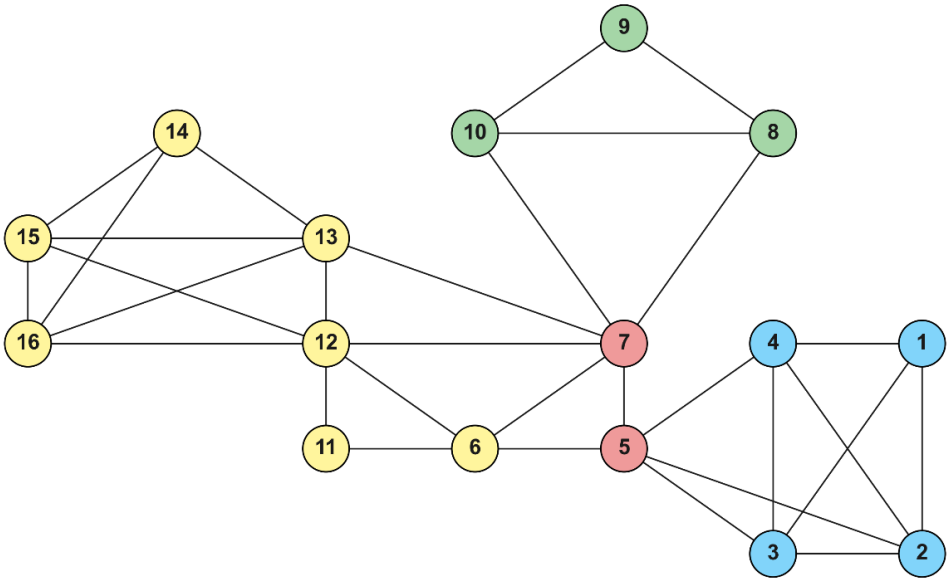
	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
Q1	1	1	0	0	0	0	0	0	0	0
Q2	1	1	0	0	0	0	0	0	0	0
Q3	0	0	1	0	0	0	1	0	0	0
Q4	0	0	0	1	1	0	0	0	0	1
Q5	0	0	0	1	1	1	0	1	0	0
Q6	0	0	0	0	1	1	0	0	0	0
Q7	0	0	1	0	0	0	1	0	0	0
Q8	0	0	0	0	1	0	0	1	0	0
Q9	0	0	0	0	0	0	0	0	1	1
Q10	0	0	0	1	0	0	0	0	1	1

درنهایت با استفاده از ماتریس آستانه، مؤلفه‌های همبند گراف را مشخص می‌کنیم و جوامع نمایان می‌شود. هر مؤلفه همبند از این گراف، یک جامعه است. این جوامع ممکن است با یکدیگر همپوشانی داشته باشند.

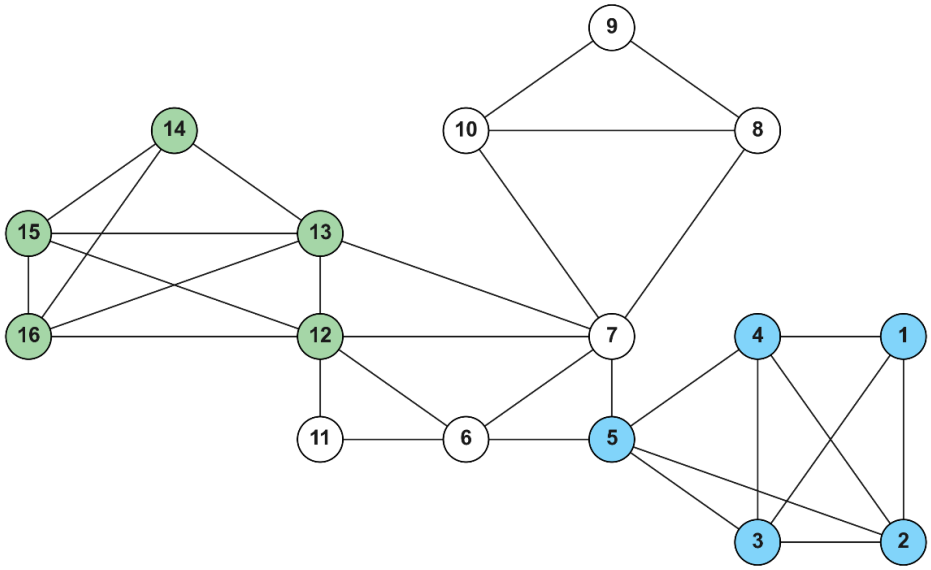


برای اطمینان از صحت جواب حاصل، این مورد را با استفاده از کتابخانه NetworkX نیز انجام داده‌ایم و نتایج کاملا یکسان و صحیح بوده است.

اگر مقدار K را برابر با ۳ درنظر بگیریم، جوامع به‌صورت زیر به‌دست می‌آید. رنگ قرمز، نشان‌دهنده رأس‌هایی است که به بیش از یک جامعه تعلق دارند.



اگر مقدار K را برابر با ۴ درنظر بگیریم، جوامع به‌صورت زیر به‌دست می‌آید. رنگ سفید، نشان‌دهنده رأس‌هایی است که به هیچ جامعه‌ای تعلق ندارند.



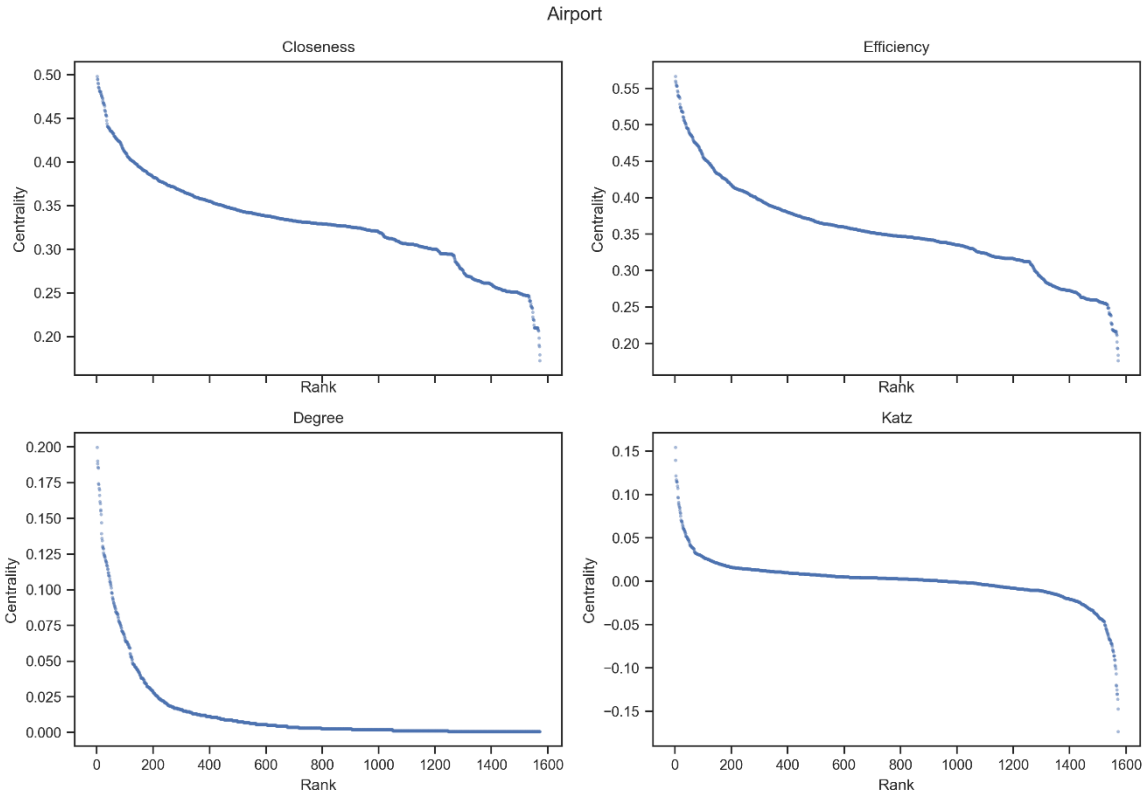
سؤال دوم

قسمت الف

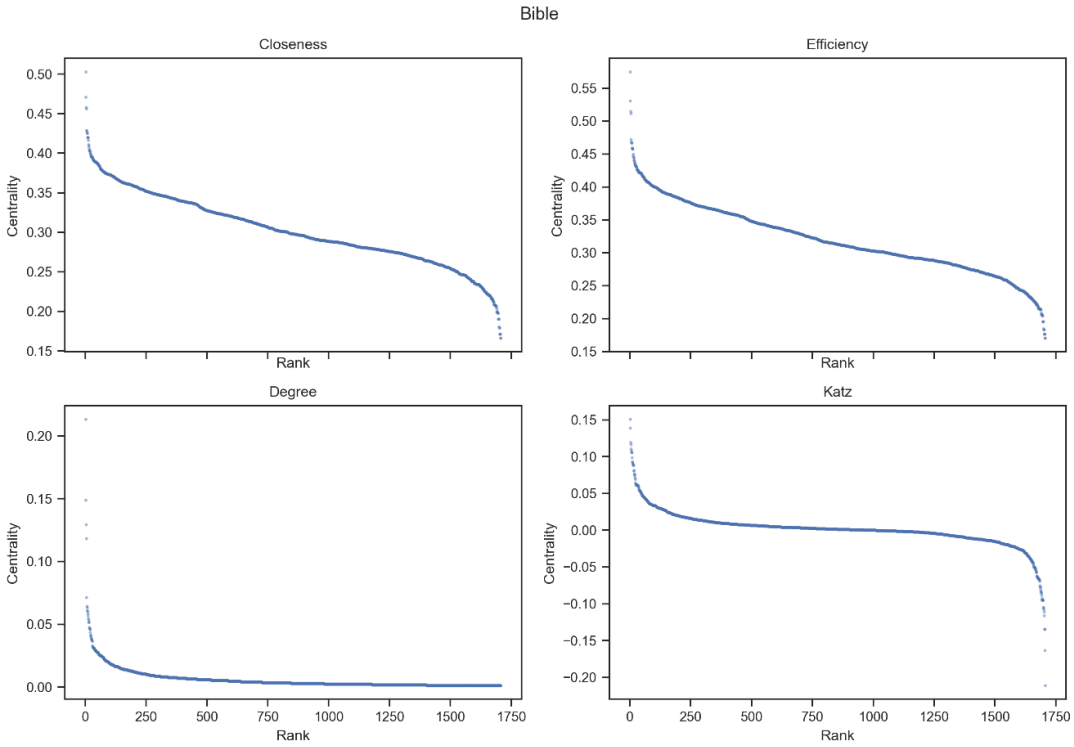
تنها برای خواندن گراف از فایل از کتابخانه NetworkX استفاده شده است و در پیاده‌سازی معیارهای مرکزیت، نقشی نداشته است. مقادیر به‌صورت نزولی در هر فایل ذخیره شده است.

قسمت ب

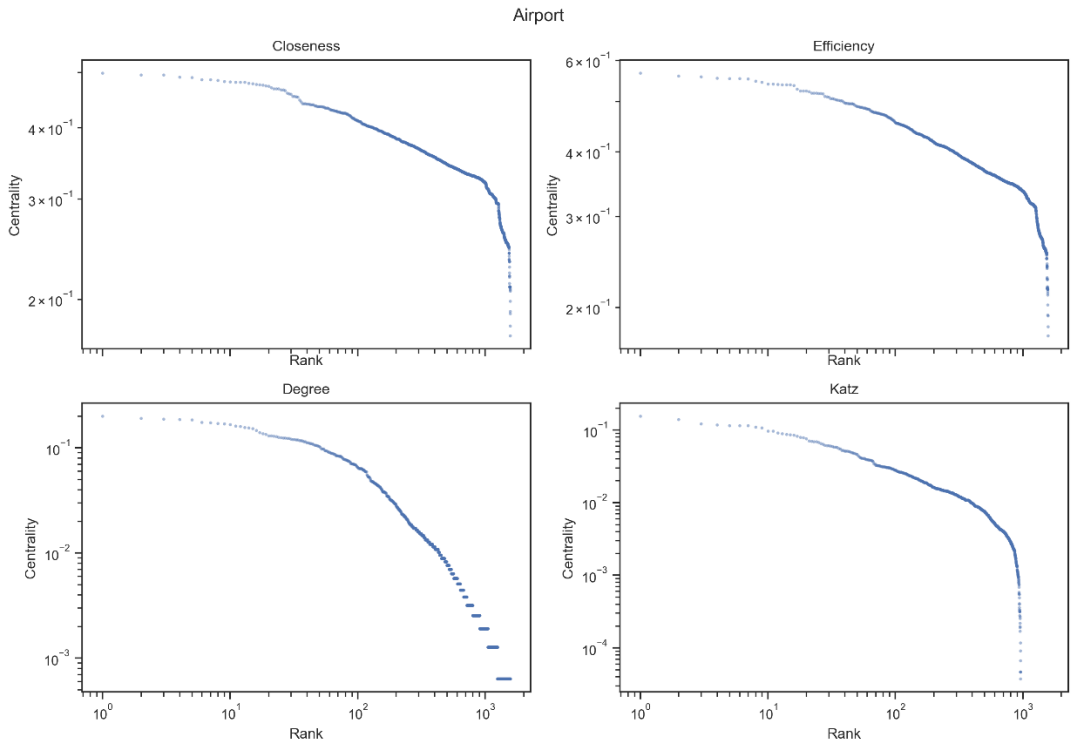
نمودارها برای هر مجموعه‌داده در دو مقیاس معمولی و لگاریتمی رسم شده است. محور افقی نمودارها، یک بار به‌عنوان رتبه گره‌ها و بار دیگر به‌عنوان شناسه گره‌ها درنظر گرفته شده است. گره‌ها برحسب مقدار مرکزیتشان به‌صورت نزولی مرتب شده‌اند.



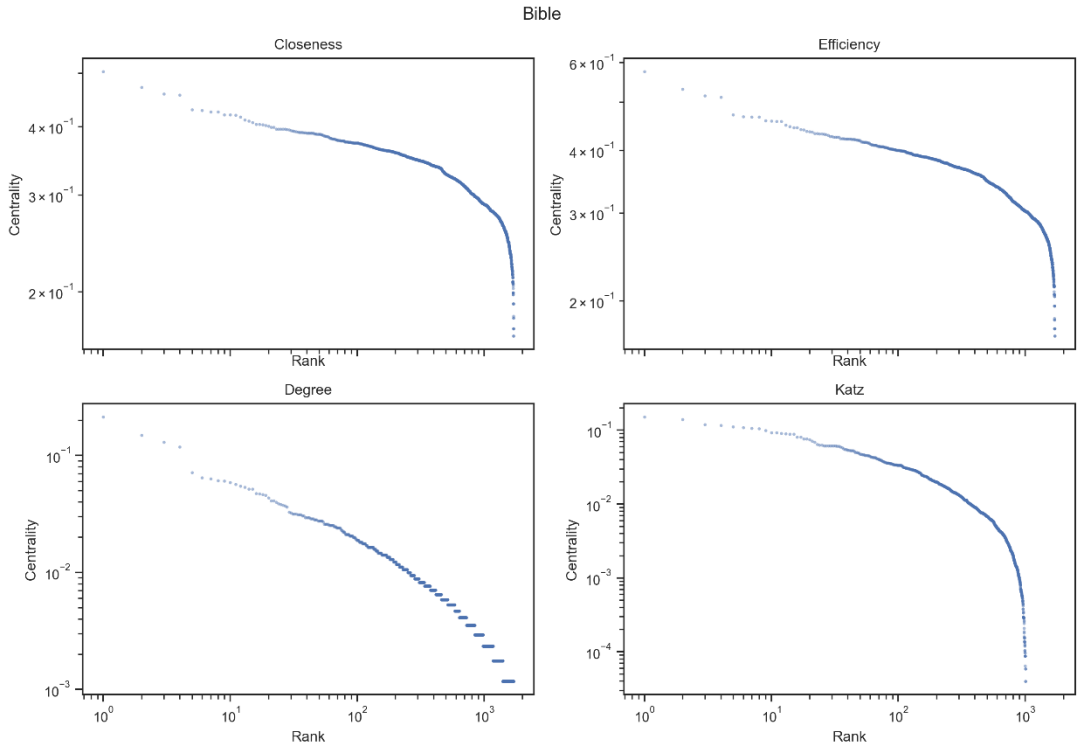
شکل ۱: نمودار مرکزیت برحسب رتبه برای گراف Airports



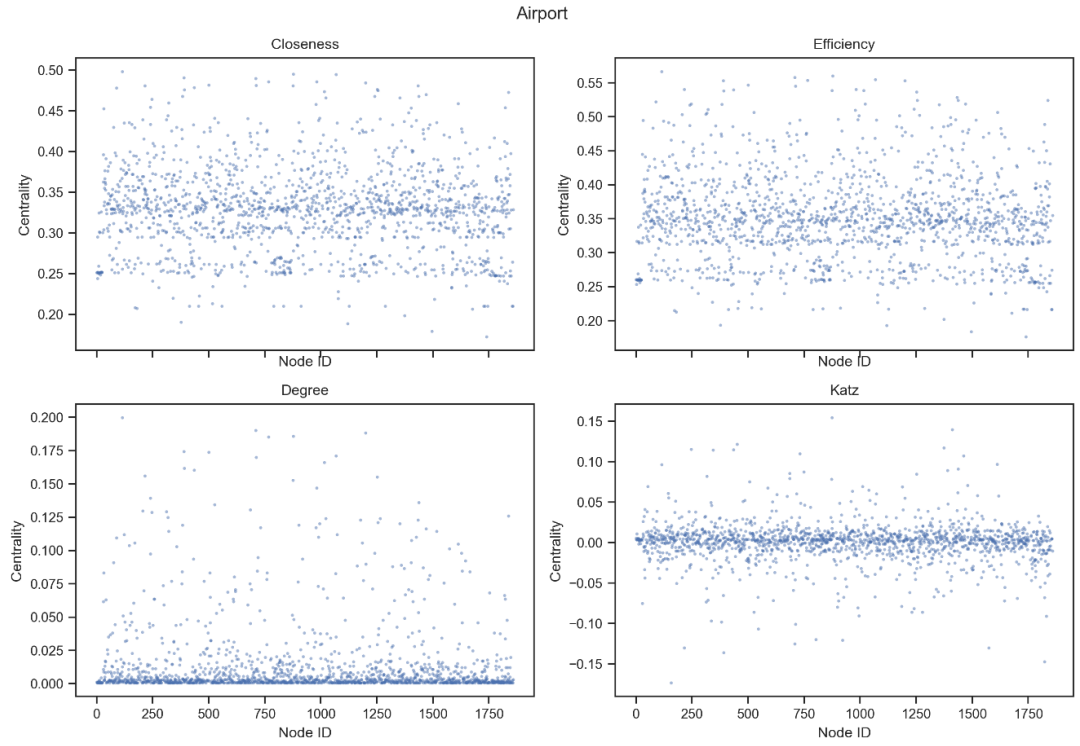
شکل ۲: نمودار مرکزیت برحسب رتبه برای گراف Bible



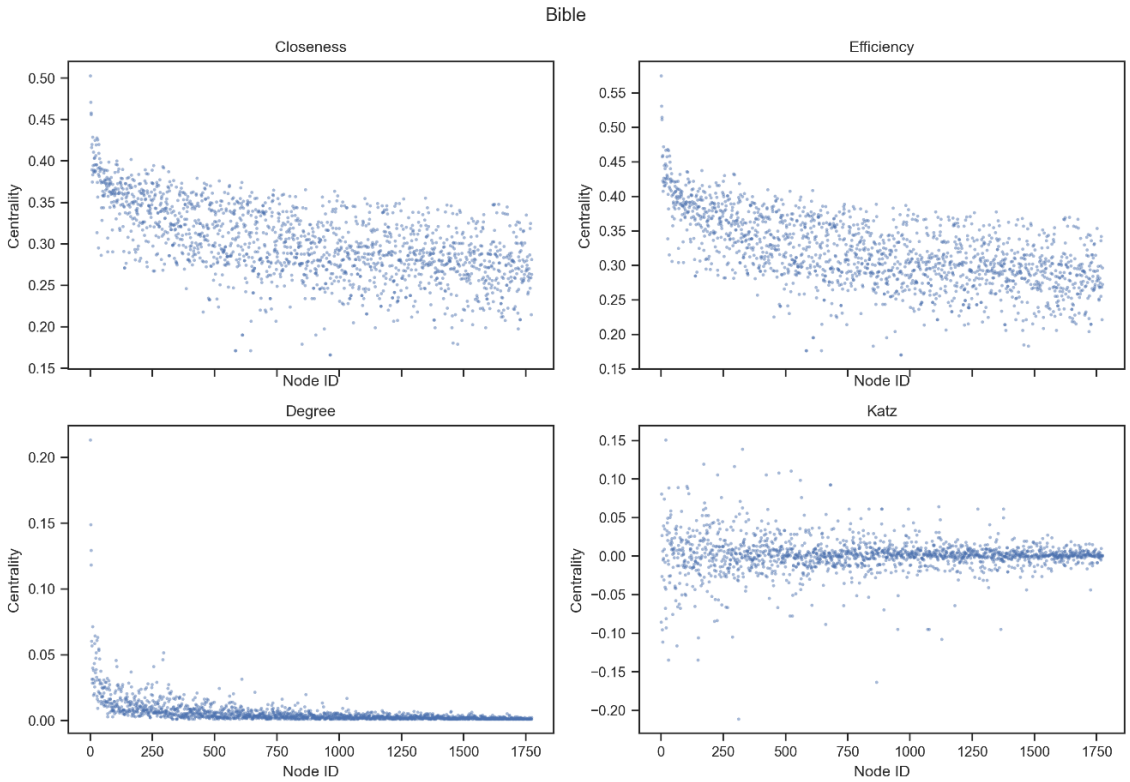
شکل ۳: نمودار مرکزیت برحسب رتبه در مقیاس لگاریتمی برای گراف Airports



شکل ۴: نمودار مرکزیت برحسب رتبه در مقیاس لگاریتمی برای گراف Bible



شکل ۵: نمودار مرکزیت برحسب شناسه گره برای گراف Airports



شکل ۶: نمودار مرکزیت برحسب شناسه گره برای گراف Bible

قسمت ج

دو معیار closeness و efficiency دارای فرم یکسان و مقادیر مشابه هستند. اگر مقدار مرکزیت را تابعی برحسب رتبه درنظر بگیریم، فرم آنها شبیه تابع logit می‌شود. تابع logit، وارون تابع sigmoid است.

معیار katz نیز فرمی مشابه تابع logit داشته است؛ اما بازه مقادیر آن در مقایسه با دو معیار قبل متفاوت است. این معیار می‌تواند مقادیر منفی هم داشته باشد، اما دو معیار قبل تنها مقادیر نامنفی را شامل می‌شوند. در این معیار، تعداد زیادی از گره‌ها مقادیر یکسانی داشته‌اند اما دو معیار قبل، مقادیر متفاوت‌تری به گره‌های مختلف اختصاص داده‌اند.

معیار degree نیز فرمی شبیه power law داشته است.

قسمت د

Airports

Rank	Closeness	Efficiency	Degree	Katz
1	0.4984 (Atlanta GA: Hartsfield-Jackson Atlanta International)	0.5668 (Atlanta GA: Hartsfield-Jackson Atlanta International)	0.1999 (Atlanta GA: Hartsfield-Jackson Atlanta International)	0.1547 (Las Vegas NV: McCarran International)
2	0.4953 (Los Angeles CA: Los Angeles International)	0.5601 (Los Angeles CA: Los Angeles International)	0.1903 (Washington DC: Washington Dulles International)	0.1396 (South Bend IN: South Bend International)
3	0.4948 (Minneapolis MN: Minneapolis-St Paul International)	0.5579 (Washington DC: Washington Dulles International)	0.1884 (Chicago IL: Chicago O'Hare International)	0.1217 (Panama City FL: Northwest Florida Beaches International)
4	0.4906 (Denver CO: Denver International)	0.5548 (Minneapolis MN: Minneapolis-St Paul International)	0.1859 (Los Angeles CA: Los Angeles International)	0.1171 (Roanoke VA: Roanoke Blacksburg Regional Woodrum Field)
5	0.4900 (Washington DC: Washington Dulles International)	0.5536 (New York NY: John F. Kennedy International)	0.1852 (New York NY: John F. Kennedy International)	0.1152 (Baltimore MD: Baltimore/Washington International Thurgood Marshall)
6	0.4858 (New York NY: John F. Kennedy International)	0.5534 (Chicago IL: Chicago O'Hare International)	0.1744 (Denver CO: Denver International)	0.1148 (Detroit MI: Detroit Metro Wayne County)
7	0.4856 (Orlando FL: Orlando International)	0.5531 (Denver CO: Denver International)	0.1738 (Newark NJ: Newark Liberty International)	0.1147 (Charleston/Dunbar WV: Yeager)
8	0.4844 (Chicago IL: Chicago O'Hare International)	0.5469 (Newark NJ: Newark Liberty International)	0.1712 (Minneapolis MN: Minneapolis-St Paul International)	0.1099 (Indianapolis IN: Indianapolis International)
9	0.4819 (Newark NJ: Newark Liberty International)	0.5451 (Houston TX: George Bush Intercontinental/Houston)	0.1700 (Houston TX: George Bush Intercontinental/Houston)	0.1071 (San Juan PR: Luis Munoz Marin International)
10	0.4813 (Houston TX: George Bush Intercontinental/Houston)	0.5406 (Orlando FL: Orlando International)	0.1661 (Miami FL: Miami International)	0.0970 (Toledo OH: Toledo Express)

Bible

Rank	Closeness	Efficiency	Degree	Katz
1	0.5029 (israel)	0.5748 (israel)	0.2134 (israel)	0.1509 (benjamin)
2	0.4710 (judah)	0.5308 (judah)	0.1489 (judah)	0.1391 (elam)
3	0.4582 (jerusalem)	0.5148 (david)	0.1295 (david)	0.1194 (shem)
4	0.4563 (david)	0.5116 (jerusalem)	0.1184 (jerusalem)	0.1164 (shemaiah)
5	0.4289 (egypt)	0.4719 (egypt)	0.0715 (egypt)	0.1104 (uz)
6	0.4279 (ephraim)	0.4678 (ephraim)	0.0645 (benjamin)	0.1083 (aram)
7	0.4254 (manasseh)	0.4670 (benjamin)	0.0633 (manasseh)	0.1057 (arphaxad)
8	0.4251 (benjamin)	0.4664 (manasseh)	0.0610 (ephraim)	0.1054 (meshech)
9	0.4202 (joseph)	0.4590 (moses)	0.0604 (saul)	0.0986 (lud)
10	0.4201 (moses)	0.4585 (joseph)	0.0586 (philistines)	0.0926 (hul)

قسمت ه

نتایج حاصل از دو معیار closeness و efficiency و مقدار مرکزیت اختصاص‌یافته به گره‌ها بسیار مشابه بوده است و می‌توان آنها را تقریباً یکسان و معادل درنظر گرفت. نتایج حاصل از معیار degree نیز شباهت زیادی به دو معیار قبل داشته است؛ اما مقادیر مرکزیت آن متفاوت است.

نتایج حاصل از معیار katz با سه معیار دیگر کاملاً متفاوت بوده است و گره‌های متفاوتی را مهم درنظر گرفته است.

مهم بودن یک گره، مفهومی نسبی است و در کاربردهای مختلف ممکن است تعریف متفاوتی داشته باشد. در صورت سؤال، تعریفی از اهمیت گره و یا کاربرد مورد استفاده ارائه نشده است؛ بنابراین نمی‌توان گفت کدام معیار در شناسایی گره‌های مهم موفق‌تر بوده است. هرکدام از این چهار معیار مزایا و معایبی دارند و در کاربردهای مختلف استفاده می‌شوند.

سه معیار اول، گره‌های معروف و شناخته شده که فاصله کمتری با سایر گره‌ها دارند یا پیوندهای مستقیم بیشتری با گره‌های دیگر دارند را مهم درنظر می‌گیرند.

معیار katz، تأثیر نسبی یک گره را با اندازه‌گیری تعداد همسایه‌های مستقیم (گره‌های درجه یک) و همچنین تمام گره‌های دیگر در شبکه که از طریق این همسایه‌های فوری به گره مورد نظر متصل می‌شوند، محاسبه می‌کند. این معیار بیشتر در تحلیل گراف‌های جهت‌دار فاقد دور مناسب است و از کاربردهای آن می‌توان تخمین وضعیت نسبی یا تأثیر بازیگران در شبکه اجتماعی یا سیستم رتبه‌بندی بصری برای تیم‌های ورزشی نام برد.

سؤال سوم

برای خوشه‌بندی در این سؤال، مجموعه‌داده فرودگاه‌ها انتخاب شده است.

قسمت الف

تنها برای خواندن گراف از فایل و ساخت ماتریس مجاورت از کتابخانه NetworkX استفاده شده است.

```
Adjacency Matrix:
[[0 1 1 ... 0 0 0]
 [1 0 1 ... 0 0 0]
 [1 1 0 ... 0 0 0]
 ...
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]]

Degree Matrix:
[[ 2  0  0 ...  0  0  0]
 [ 0 40  0 ...  0  0  0]
 [ 0  0  9 ...  0  0  0]
 ...
 [ 0  0  0 ...  1  0  0]
 [ 0  0  0 ...  0  1  0]
 [ 0  0  0 ...  0  0  1]]

Laplacian Matrix:
[[ 2 -1 -1 ...  0  0  0]
 [-1 40 -1 ...  0  0  0]
 [-1 -1  9 ...  0  0  0]
 ...
 [ 0  0  0 ...  1  0  0]
 [ 0  0  0 ...  0  1  0]
 [ 0  0  0 ...  0  0  1]]
```

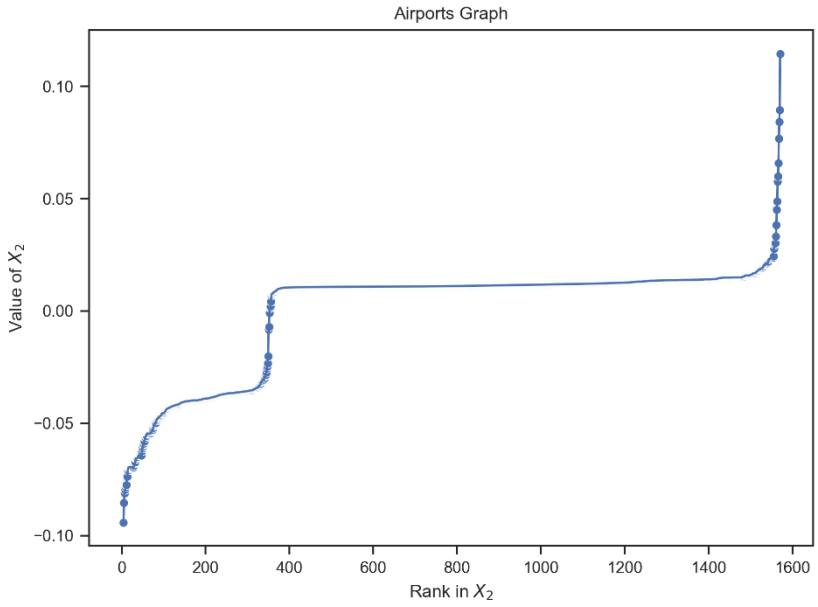
قسمت ب

مقادیر ویژه با استفاده از کتابخانه NumPy محاسبه شده است.

```
Smallest Eigenvalue: 0.0
Corresponding Eigenvector:
[[0.02522166]
 [0.02522166]
 [0.02522166]
 ...
 [0.02522166]
 [0.02522166]
```

```
 [0.02522166]]

Second Smallest Eigenvalue: 0.21795319159317073
Corresponding Eigenvector:
[[-0.06043121]
 [-0.05436919]
 [-0.05332205]
 ...
 [ 0.01377518]
 [ 0.01483884]
 [ 0.02437507]]
```



قسمت ج

خوشه‌بندی با استفاده از الگوریتم k-means و کتابخانه scikit-learn انجام شده است.

قسمت د

معیار Modularity با استفاده از رابطه زیر محاسبه می‌شود:

$$Q(G,S) = \frac{1}{2m}\sum_{ij}\left(A_{ij} - \frac{k_i k_j}{2m}\right)\delta(s_i, s_j)$$

که اگر هر دو رأس i و j در یک خوشه باشند، $\delta(s_i, s_j)$ مقدار ۱ را دارد و در غیر این صورت مقدارش صفر است. برای افزایش سرعت محاسبات، می‌توان آن را به رابطه زیر تبدیل کرد:

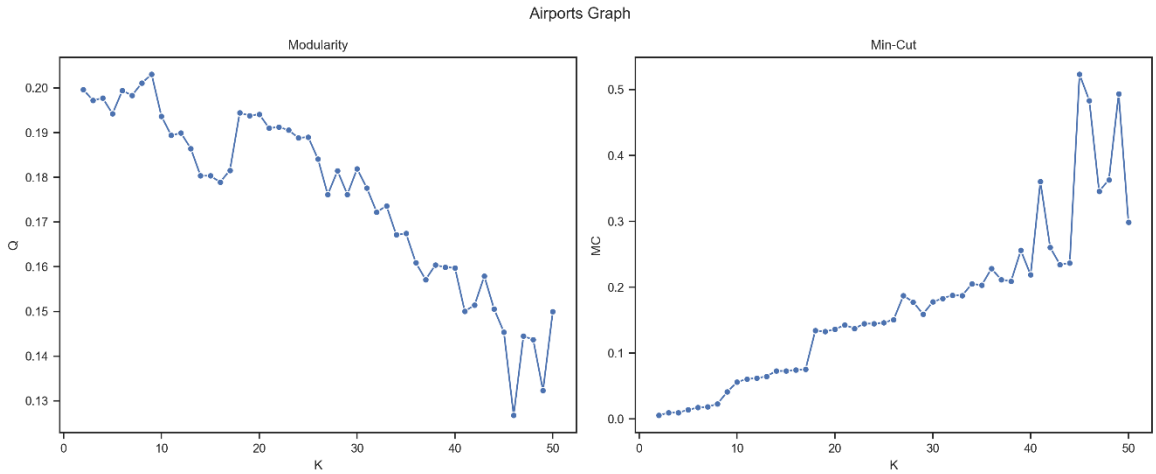
$$Q(G,S) = \sum_{s \in S} \left[\frac{E_s}{m} - \left(\frac{k_s}{2m} \right)^2 \right]$$

که E_s تعداد یال‌های درون‌خوشه‌ای و k_s مجموع درجه رأس‌های موجود در خوشه s را نشان می‌دهد. معیار Min-Cut نیز با استفاده از رابطه زیر محاسبه شده است:

$$MC(G,S) = \frac{1}{2m} \sum_{s \in S} \sum_{i \in s, j \notin s} w_{ij}$$

نمودارهای زیر مقادیر هر دو معیار را با تغییر تعداد خوشه‌ها (K) برای الگوریتم k-means نشان می‌دهد. زمانی که مقدار ۹ را برای هایپرپارامتر K درنظر بگیریم، بیشترین میزان Modularity را داریم. معیار Min-Cut نیز زمانی که تعداد خوشه‌ها برابر با ۲ باشد، کمترین مقدار را دارد.

```
Best K for Modularity: 9 , Q: 0.20302801991656394
Best K for Min-Cut: 2 , MC: 0.005460671546415708
```



سؤال چهارم

تنها برای خواندن گراف از فایل از کتابخانه NetworkX استفاده شده است و در پیاده‌سازی الگوریتم اصلی نقشی نداشته است. محاسبه بردارهای ویژه با پیاده‌سازی روش توانی انجام شده است.

مجموعه‌داده wiki-vote شامل یک گراف جهت‌دار است، اما آن را به‌صورت غیرجهت‌دار درنظر می‌گیریم:

```
Graph with 7115 nodes and 100762 edges
```

نتایج حاصل از پیاده‌سازی و اجرای الگوریتم مطابق ذیل است:

```
Results for the Fast Modularity Optimization Algorithm:
Number of detected communities: 3
Modularity: 0.3998
```

جوامع کشف شده، در فایل communities.txt ذخیره شده است. در این فایل برای هر جامعه، شناسه گره‌هایی که در آن قرار دارند مشخص شده است. به‌منظور مقایسه نتایج، دو مورد از الگوریتم‌های تشخیص جامعه که در کتابخانه NetworkX موجود است نیز بر روی گراف اجرا شده است:

```
Results for the NetworkX Greedy Algorithm:
Number of detected communities: 55
Modularity: 0.3402


Results for the NetworkX Louvain Algorithm:
Number of detected communities: 29
Modularity: 0.4264
```

نتایج نشان می‌دهد که مقدار Modularity حاصل از اجرای الگوریتم Louvain بیشتر بوده و در نتیجه عملکرد بهتری داشته است. پس از آن FMO و Greedy به‌ترتیب در جایگاه دوم و سوم قرار می‌گیرند.