Problem 1

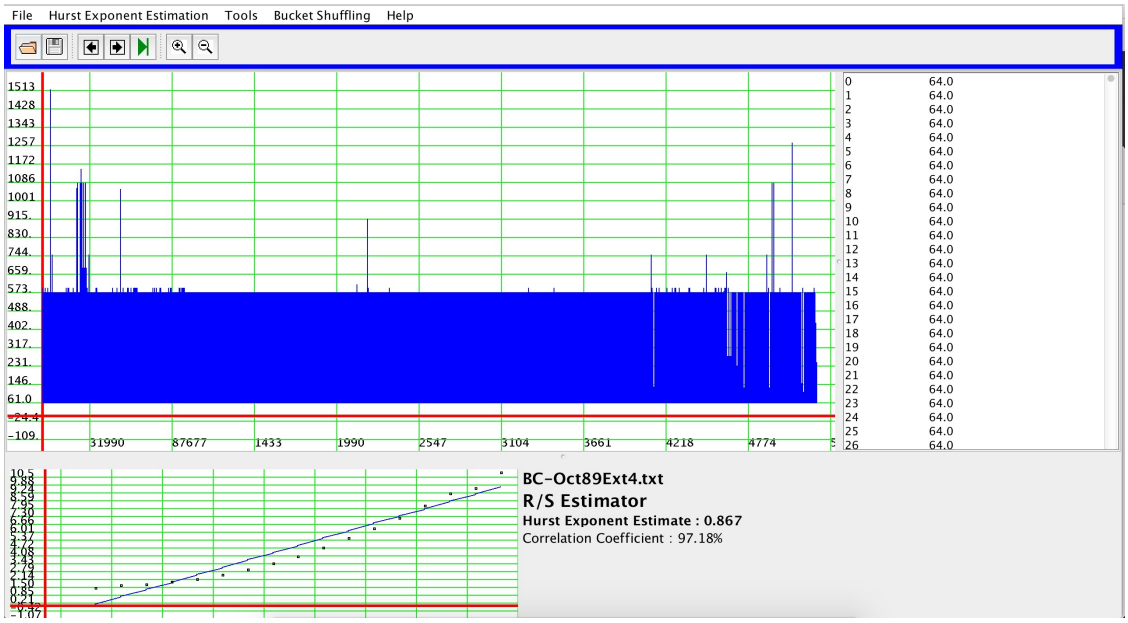
Starting from two real world traces, namely BC-Oct89Ext4.1xt which is obtained from packet size distribution in LAN traffic.
NoC_10x10_400_8KCache_4fDPac.txt which is obtained from data reply requests in on-chip traffic, perform the following statistical tests and comment on the results you get:

a) Download from the Canvas and run the SELFIS software to estimate the Hurst exponents of the packet size in BC-Oct89Ext4.txt and data reply requests in NoC_10x10_400_8KCache_4fDPac.txt (timeseries) via the R/S and variance-time methods. Discuss the sources of discrepancy between Hurst exponents obtained with these two methods.

Solution:

We first evaluate the packet size distribution of the BC-Oct89Ext4 benchmark.

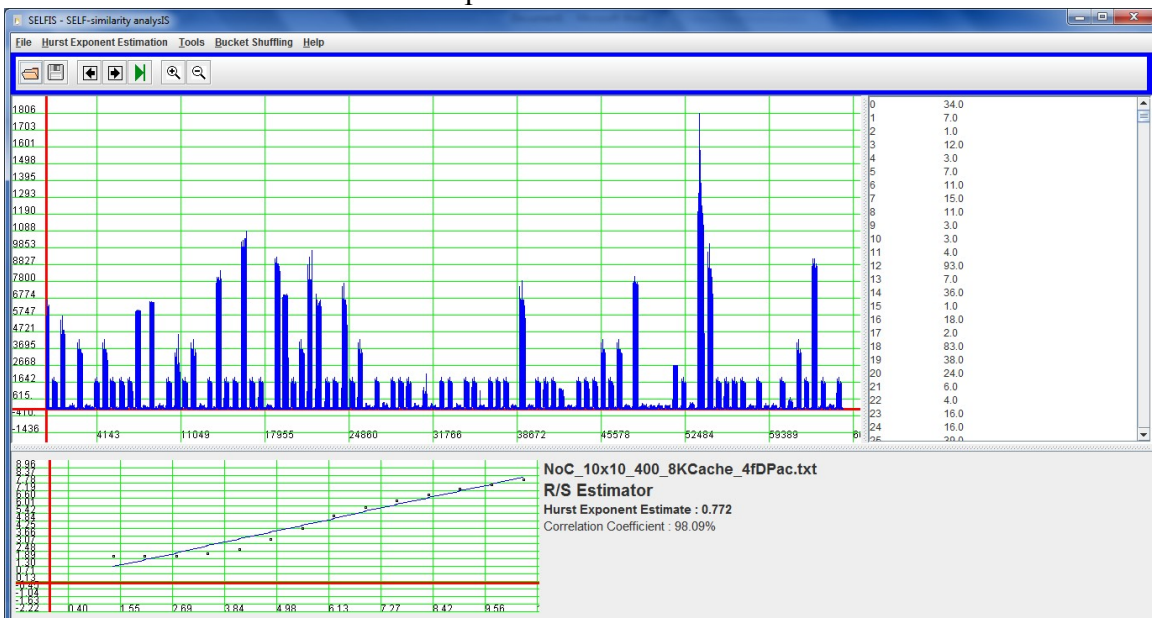
For the BC-Oct89Ext4 benchmark, the R/S method evaluates the Hurst parameter as: 0.867



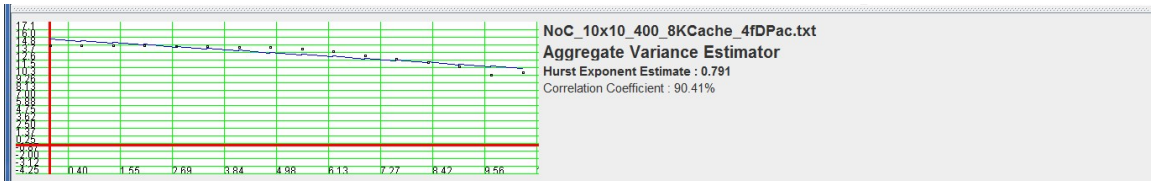
The variance time method reports the Hurst parameter as: Hurst parameter 0.901

For the NoC_10x10_400_8KCache_4fDPac benchmark, the R/S method and variance time method are reported as follows:

The R/S method estimates the Hurst parameter to be 0.772



The variance time method reports the Hurst parameter to be 0.791



The sources of discrepancy between Hurst exponents obtained of these two methods is mainly due to the two methods are applicable to stationary stochastic process.

The two traces files here have limited samples (not strictly stationary process) and the curve fitting to estimate the slope will generate some errors. Overall, I think the two

methods generate very close Hurst parameter and the difference is not too large.

b) Download from Canvas and run the Matlab code for detrended fluctuation analysis (DFA) on the packet size distribution in *BC-Oct89Ext4.txt* and data reply requests in *NoC_10x10_400_8KCache_4fDPac.txt* (time series).

Solution: For NoC traffic, the DFA analysis results is shown in the Fig.1, the Hurst parameter reported is 0.7918 which is very close to the results of variance time methods.

For the LAN traffic, the packet size distribution using DFA analysis is reported as in Fig. 2. The Hurst parameter reported is 1.0061. The R/S method has a closer estimation compared with Variance time method in The LAN traffic analysis.

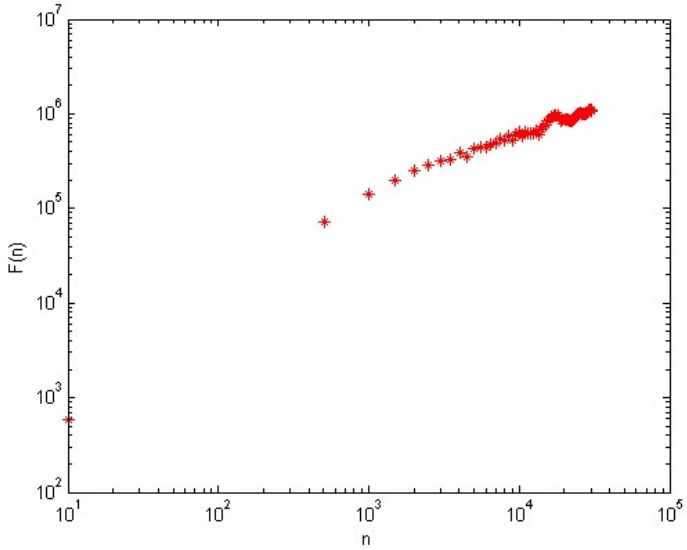


Fig. 1: NOC inter arrival time DFA analysis.

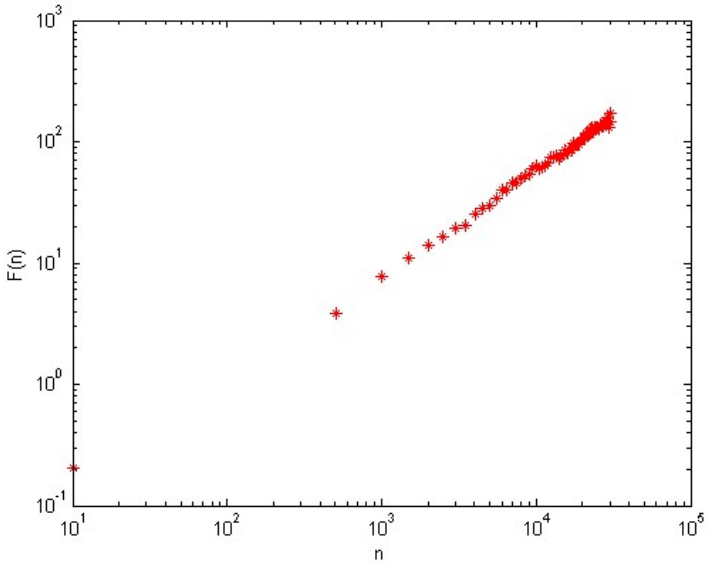


Fig. 2: BCT packet size distribution with DFA analysis.

Problem 2

The goal of this problem is to explore LinkedIn ego-network structure using provided data. The data contains 250 ego-networks which have user attributes and their relationships (with types). Each student is randomly assigned an ego-network. We define the ego-user to be the person (i.e., node) whose ego-network is analyzed. Each ego- network dataset comes with the following information:

a. Complete edge list for the ego-network
   Example:
   U0 U0 U1
   U0 U0 U2
   …
   U1046 U1046 U1172
   U1046 U1046 U1173

   Here, a record "U0 U0 U1" means that U0 is connected to U1 in U0's ego-network.

b. Relationship between ego users' and their friends
   Example:
   U3297 U3299 L4
   U10305 U10351 L2
   U10305 U10369 L2

   Here, the record "U10305 U10369 L2" means that the type of relationship between U10305 and U10369 is L2. Of note, there can possibly exist 8 types relationships (total) in your ego-network.

c. Profile attributes (location, college, employer) for the users
   Example for college.txt
   U346
   2
   C0
   C1

   Here, U346 is UserID, 2 is the number of values associated with the attribute, e.g., U346 attended two colleges (C0 and C1).

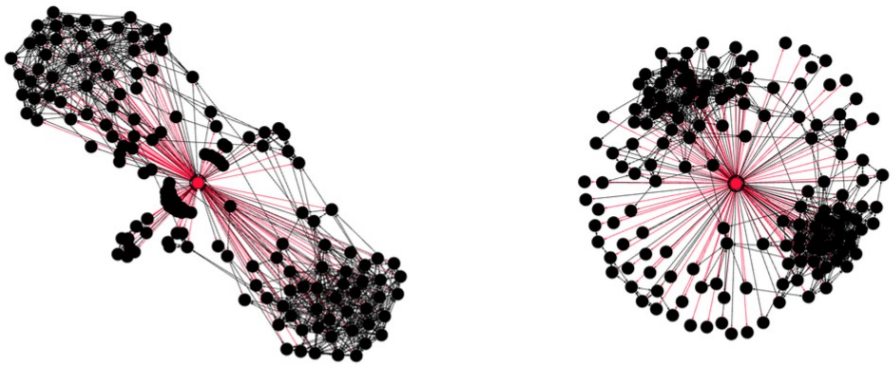
For more details of the dataset, please refer to https://arxiv.org/abs/1309.4157 and http://wwwconference.org/proceedings/www2014/proceedings/p819.pdf.

In this problem, we want you to perform an in-depth analysis of your assigned ego-network. The following steps will allow you to understand the structure of ego-network in detail.

1. Preliminary analysis

a. **Visualization.** Visualize your data using at least two layout techniques. Choose the edge to be directed. Color the ego-user in your network and the edges connecting to it in red. Report the number of nodes and edges in your ego-network.

Solution:[1] The visualizations of the ego-network are shown under layout techniques of ForceAtlas and Yifan Hu Proportional in Figure 3(a)(b) respectively, ego-user and edges connecting to the ego-user are colored in red. There are 157 nodes and 1788 edges in the ego-network.

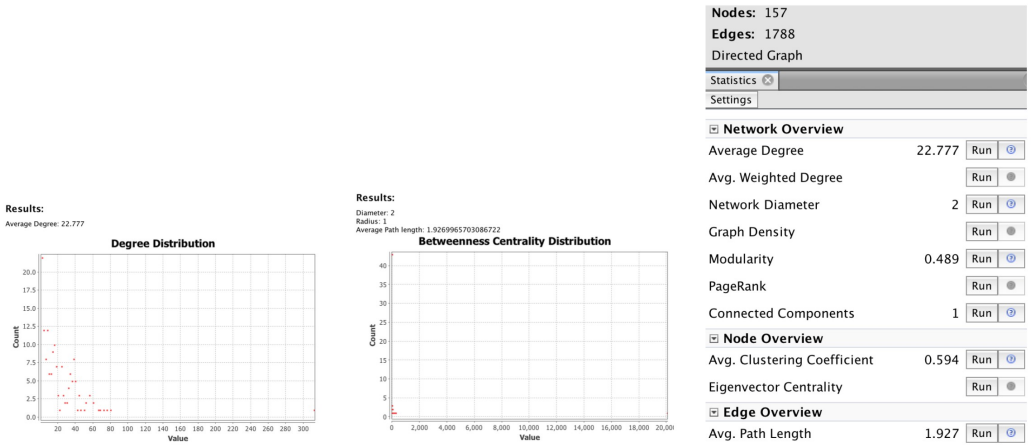


(a) Force Atlas layout. (b) Yifan Hu Proportional layout.

Fig. 3: Visualization of the ego-network under different layout techniques.

b. **Network analysis.** Report degree distribution, node degree, clustering coefficient, average path length, betweenness centrality distribution, and number of communities.

Solution: Degree and betweenness centrality distribution are shown in Fig 4(a) and (b), and other statistics are shown in Fig 4(c), where the average degree is 22.777, clustering coefficient is 0.594, average path length is 1.927, and there are 3 communities under resolution 1 without using edge weight.



(a) Degree distribution. (b) Betweenness centrality distribution. (c) Other statistics.

Fig. 4: Network characteristics.

2. One step further

a. **Popularity**. Who is the most popular person in your ego-network? Explore node degree and betweenness centrality in your ego-network with the ego-user removed. Does the most popular person score high on both metrics? Highlight the most popular people in your network (again, the ego-user should be removed in this case).

[1] All the solutions are based on UserID "U2100".

Solution: The ego-network with the ego-user removed is shown in Fig 5. The most popular person in terms of degree is the one with label *U2196* (colored in red) whose degree is 50, which means he is connected to other 25 friends in the ego-network, the betweenness centrality associated with node *U2196* is 228.617335. The most popular person in terms of betweenness centrality is node *U2202* (colored in blue) whose betweenness centrality is 744.867958, and its degree is 16. We can see the most popular person has both high degree and betweenness centrality, which means the popular person has connections to many others and he is an important connector for any pair of persons to get to know each other.

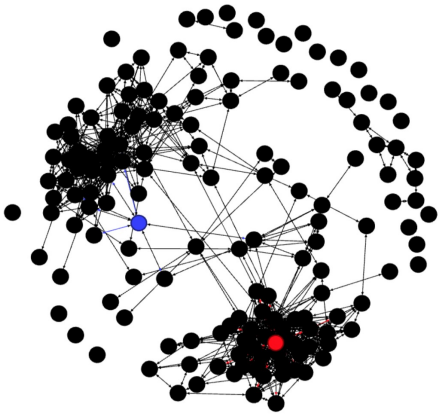


Fig. 5: ego-network with the ego-user removed.

b. **Community structure**. In this part, we want you to visualize communities using different methods:

i) Use the modularity-based community detection embedded in Gephi to detect the communities in your ego-network. Visualize and color communities using different colors.

ii) Explore the type of relationships among users in your assigned ego-network. Group users together if their edge type (relationship) is the same. For instance, in example b. in the beginning of this problem, U10305, U10351 and U10369 should belong to the same group L2, while U3297 and U3299 should belong to another group L4. We call these groups the ground truth circles[2]. Visualize your ground truth circles and color them.

iii) Compare the results from ii) with the communities detected by Gephi in i) and discuss similarities/ differences. Are there people who act as bridges between different ground truth circles? If yes, re-size and visualize these nodes (users) using bigger weights.

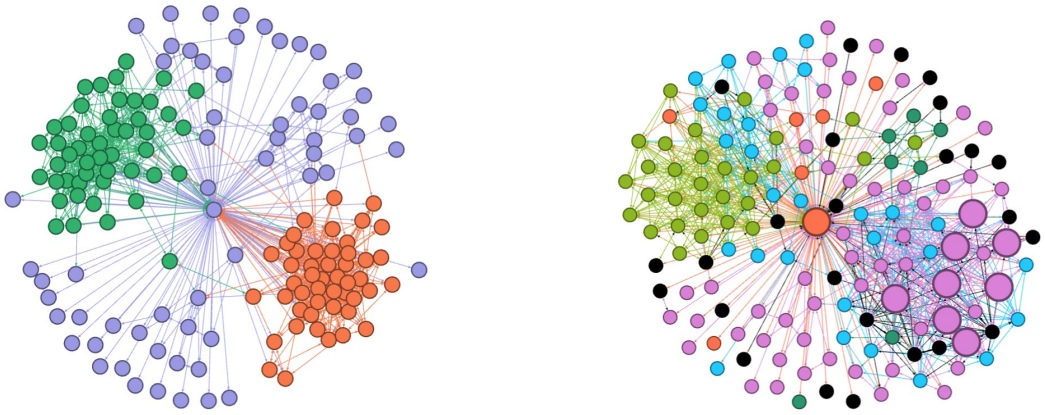
Hint: To visualize these circles, import a spreadsheet (.csv file containing Id, Label, and circle of each node) into Gephi and color nodes by choosing the attribute "circle". A snapshot of such a spreadsheet is shown below:

circle_feature.csv

```
1  Id,Label,circle
2  U2133,U2133,5
3  U2100,U2100,5
4  U2105,U2105,5
5  U2156,U2156,5
6  U2164,U2164,5
7  U2130,U2130,5
8  U2194,U2194,5
9  U2144,U2144,5
```

[2] Note that although there may exist 8 types of relationship among users, some type of relationships may be missing in your assigned ego-network, and some edges may not be labeled as well, so just work on whatever is available for your ego-network.

Solution: Communities detected by Gephi is shown in Figure 6(a), and communities based on edge labels (ground truth circle) is shown in Figure 6(b). Modularity based community detection gives some prior knowledge on the ground truth circles, but it cannot detect nodes that have affiliation to multiple circles. It takes only the network structure into account, while the network ground truth circles are formed by different aspects (common properties), and may overlap with each other. Since modularity based community detection does not consider nodes' profile, the obtained communities from Gephi are not reliable. There are people who act as bridges (have affiliation to multiple communities) in the ground truth circles, these people are visualized with bigger size of node in Figure 6(b).



(a) Community detection based on modularity.

(b) Community detection based on edge label.

Fig. 6: Visualization of the ego-network using different communities.

c. **Clustering**. Compute the average clustering coefficient for each ground truth circle in your network and compare it to the average clustering coefficient of the entire ego- network. Do you see any significant differences?

Solution: The clustering coefficient of each of the circle is shown in Table I. The clustering coefficient of the entire ego-network is 0.594. There are both circles with clustering coefficients higher and lower than that of the entire ego-network. Some of the circles are highly connected, such us Circle 6 and 8, but Circle 5 and 7 are loosely connected compared with the entire network.

	clustering coefficient
circle 1	0.408
circle 2	NA
circle 3	NA
circle 4	NA
circle 5	0.35
circle 6	0.653
circle 7	0.295
circle 8	0.657

TABLE I: Clustering coefficient of each of the ground truth circle.

d. **Homophily**[3]. Explore various features (e.g., college, employer, location) of users in

[3] Homophily (i.e., "love of the same") is the tendency of individuals to associate and bond with similar others.

different ground truth circles. Use entropy[4] to measure the homophily of each circle under different features, and then compare it with the average clustering coefficient of each circle.

Report your results in a table as shown below. Replace '*' with the value of clustering coefficients or entropy measured homophily; fill in the corresponding row(s) as 'NA' if your ego-network does not have certain circle(s).

	clustering coefficient	*Homo college*	*Homo employer*	*Homo location*
circle 1	*	*	*	*
circle 2	*	*	*	*
circle 3	NA	NA	NA	NA
circle 4	*	*	*	*
circle 5	*	*	*	*
circle 6	*	*	*	*
circle 7	*	*	*	*
circle 8	*	*	*	*

Solution: Table II shows the entropy of each circle calculated by college, employer, and location feature respectively. We can see that circle with high clustering coefficients usually has lower entropy values, which means that such circle is homogenous.

	clustering coefficient	*Homophily college*	*Homophily employer*	*Homophily location*
circle 1	0.408	1.5962	6.5668	2.5130
circle 2	NA	NA	NA	NA
circle 3	NA	NA	NA	NA
circle 4	NA	NA	NA	NA
circle 5	0.35	3.8634	7.3564	3.1139
circle 6	0.653	1.7335	5.7876	1.7907
circle 7	0.295	2.2516	4.5236	1.8424
circle 8	0.657	1.3788	4.0235	0.5436

TABLE II: Entropy measured homophily of each circle using different features.

3. Getting even deeper

a. How homogeneous are communities in your ego-network? Do they tend to share similar characteristics? Do similar people belong to the same communities?

Solution: The circles are homogenous as shown in Table II; people in the same circle are likely to have similar characteristics (profiles), especially when the circle are highly connected (e.g. circles group similar people).

b. How interconnected is your LinkedIn ego-network? Is there a high degree of overlap between communities in your ego-network?

Solution: In the ground truth circle affiliation, there are 9 persons belonging to multiple circles; they are shown in Figure 6(b) with bigger size of nodes. Given the small size of the ground truth circles and the huge size of the ego-network, it is reasonable to say that the LinkedIn ego-network is highly interconnected.

Problem 3 Fraction dimension calculation

Solution: According to the paper reference, we assume the relationship of the box radius and size is

L= 2r + 1 (L is the box size while r is the box radius) Running MEMB

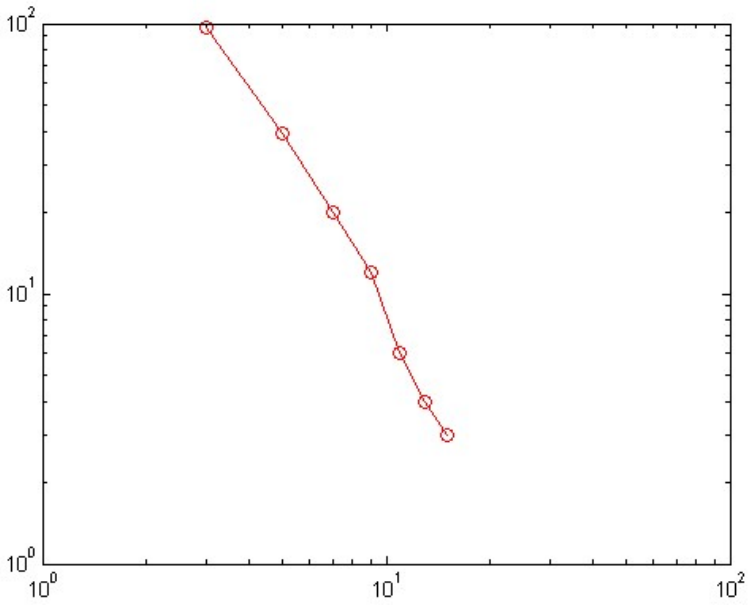
r= [1, 2, 3, 4, 5, 6,7]

L=[3,5,7,9,11,13,15]

The number of boxes is

M=[97, 39, 20, 12, 6, 4, 3];

The plot of the loglog scale L,M relationship is



After fitting the slop of the curve above, we can see the fractional dimension is: 2.2060 (or 1.4983 if polyfit the only the first 4 radius, i.e., r=1-4). This shows that the network exhibits self-similar behavior.

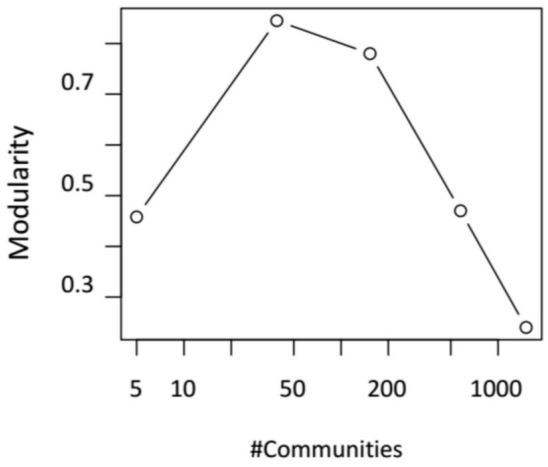
Problem 4: Communities in research collaboration

In this question, you will analyze the community structures in research collaboration. First, download the arXiv collaboration network used in Homework 1 (arXiv_lcc.gml). Second, use the community detection algorithm provided in Gephi (https://gephi.org/) to identify communities in this network. Use the range of resolutions (a user-specified parameter) in [0.001, 10] to obtain the resulting modularity and number of communities. Note that, in contrast to the case mentioned in class, a lower resolution in Gephi's community detection algorithm corresponds to a lower number of communities.

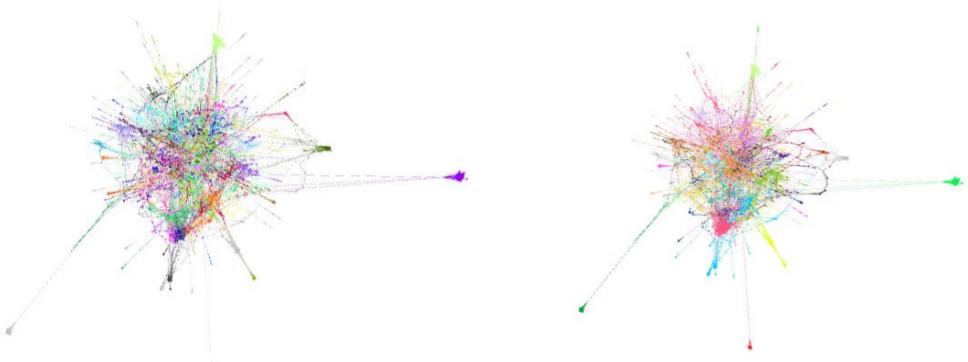
Plot the modularity as a function of number of communities. Visualize the network at resolutions of 0.1 and 0.75.

Solution:

a)



b) Visualizations should be intuitive due to the amount of clustering denoted by color.



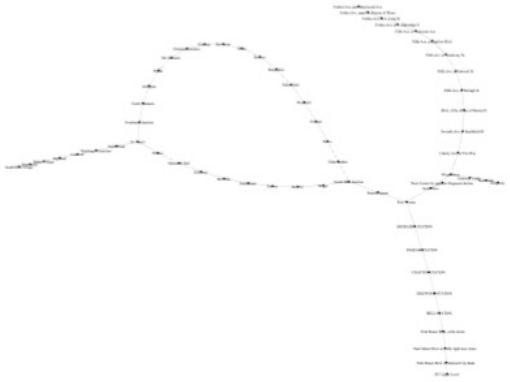
Modularity Resolution = 0.1

Modularity Resolution = 0.75

Problem 5: Pitt Transportation Network

a) Visualize the transportation network, referencing the 'Station Name' field for node identification.

Solution:



b) Analyze the 'small-worldness' of the network you obtain by calculating the average path length, nodes degree, and clustering coefficient.

Solution:
Average degree = 2.032
Average path length = 11.785
Clustering coefficient = 0.026
This network doesn't reflect small-world behavior with a low clustering coefficient due to only having one triangle present in the network and a high average path length. Not many hubs present.

c) Insert several long-distance links (do not connect neighboring nodes) to create 'triangles' within the network in order to reduce the path length by at least 30%. Comment on your results and explain them within the context of 'small-world' networks.

Solution:
By adding several long distance links we are able to reduce the average path length and with more triangles present in the network, the clustering coefficient is reduces. Increasing the amount of links will lead to increased small-world behavior. (looking for path length < 8.3)