Network Midterm

Chenrui Xu

I Background

Community detection is one of the most important areas in network analysis. It has many kinds of applications in the real world. For example, to detect some potential group of people, detect a cluster of things one person might be interested in so that the system can do the recommendation or some other applications in IT company. According to the paper Fast unfolding of communities in large networks, we can use a very fast way to make network data into clusters/communities via using modularity knowledge.

Modularity is one measure of the structure of networks or graphs. The range of it can be from -1 to 1 to represent the possibility network can be divided into modules. While dealing with billions of data, it is necessary to find out one way to make clusters converge fast. That is the method we are going to talk about today.

II Task I

Network
Chenrui Xu

$$Q=\frac{1}{2m}\sum_{ij}\left(A_{ij}-\frac{k_ik_j}{2m}\right)\cdot\delta(c_i,c_j)$$

in statistic, we calculate variance like this

$$Var=\frac{\sum(X_i-\bar{X})^2}{n}\xrightarrow{\text{a little transform}}\frac{\sum(X_i-\bar{X})}{n}$$

in the network, $n=2m$.

① A_{ij} can be seen as the real value. (real weight)

② $\left(\frac{k_ik_j}{2m}\right)$ is the expected value (weight) $E(X)=\sum X\cdot P(X)$.

Prove $\left(\frac{k_ik_j}{2m}\right)$: the meaning is the expected weight of node i/j

so the $P(i)=\frac{k_i}{2m}$ → degree of i ; → total degree (m edges, but undirected, counted twice)

$P(j)=\frac{k_j}{2m}$

$P_{ij}=P(i)\cdot P(j)$ assuming independent.

$E(ij)=2m\cdot P_{ij}=\left(\frac{k_ik_j}{2m}\right)$ proved.

③ $\delta(c_i,c_j)$: the constrain that only i,j be in same cluster the whole formula.

we can ignore it after little transformation:

Given i,j in same cluster.

$$Q=\frac{1}{2m}\sum_{ij}\left(A_{ij}-\frac{k_ik_j}{2m}\right)$$

so all terms in formula clear. back to look at $\frac{\sum(X_i-\bar{X})}{n}$

$X_i=A_{ij}$ $\quad P_{ij}=\bar{X}=\frac{k_ik_j}{2m}$ $\quad n=2m$

so $\frac{\sum_{ij}\left(A_{ij}-\frac{k_ik_j}{2m}\right)}{2m}$ (Given i,j in same cluster)

In random graph model, nodes and edges are generated randomly, isolatedly

Q is easy to understand.

It measure the distance (how different it is) between $\boxed{\text{the expected weight of } ij}$ and real weight of ij in same cluster.

①

Q can be seen as how different (I will consider it as one format of variance with no L1/L2 regularity) the real model and random graph model is in terms of existence of edges. That is why Q is the statistics of modularity.

$$Q=\sum_{c=1}^{n_c}\left(\frac{l_c}{m}-\left(\frac{d_c}{2m}\right)^2\right)$$

Network
Chenmi Xu

This is another formula to calculate Q.

The previous idea is going through all i,j combination pairs.

for this one, we sum value of each cluster together.

As I mentioned in the first page, the expected value is based on the random graph model, so Q actually measures the difference between how real edges and statistic edges lay out, that is why the definition of modularity is the value $(-1,1)$ to show the propobility network can be clusters.

Back to second formula, it can be seen as a change format of first formula.

Given i,j in same cluster.

$$Q=\sum_{i,j}\left(A_{ij}-\frac{k_ik_j}{2m}\right)\cdot\frac{1}{2m}$$

$$\Rightarrow Q=\sum_{c=1}^{n_c}\left(\sum_{i,j}\left(A_{ij}-\frac{k_ik_j}{2m}\right)\cdot\frac{1}{2m}\right)\quad(\text{given } i,j\in C)$$

$$=\sum_{c=1}^{n_c}\left(\sum_{i,j\in C}\left(\frac{1}{2m}\cdot A_{i,j}-\frac{k_ik_j}{4m^2}\right)\right)$$

$$=\sum_{c=1}^{n_c}\left(\frac{\left(\sum_{i,j\in C}A_{i,j}\right)\cdot\frac{1}{2}}{m}-\frac{\sum_{i,j\in C}k_ik_j}{4m^2}\right).$$

Notice here ① $\sum_{i,j\in C}A_{i,j}$ is the sum of edges in cluster C, but counted twice.

so $\frac{1}{2}\sum_{i,j\in C}A_{i,j}$ is the total number of edges joining vertices of cluster C.

we use the term $l_c=\frac{1}{2}\sum_{i,j\in C}A_{i,j}$.

② $\sum_{i,j\in C}k_ik_j=\sum_{i\in C}\sum_{j\in C}(k_ik_j)=\sum_{i\in C}\left(k_i\sum_{j\in C}k_j\right)=\sum_{i\in C}k_i(d_c)$.

we use d_c to represent ~~the~~ all degrees of nodes in cluster C.

so $\sum_j k_j$ is a constant (k_j is the degree of one node j)

$$\Rightarrow\sum_{i,j\in C}k_ik_j=(d_c)\cdot(d_c).$$

Here is the process why two equations equals.

Network
Chenrui Xu

so $Q = \sum_{c=1}^{n_c} \left(\frac{l_c}{m} - \frac{d_c \cdot d_c}{4m^2} \right) = \sum_{c=1}^{n_c} \left(\frac{l_c}{m} - \left(\frac{d_c}{2m}\right)^2 \right)$.

Then, I will explain the physical meaning of the second formula.

According to page one. I recalled expectation $E(X) = \sum X \cdot P(X)$

and transformed variance $\frac{\sum(X_i - \overline{X})}{n}$.

it can be transformed again into $\sum \left(\frac{X_i}{n} - \frac{\overline{X}}{n} \right)$

looks similar right now, right?

① $\frac{d_c}{2m}$. (as total degree is double counted), so the propability the the degree belongs to cluster C is $\frac{d_c}{2m}$.

what if $\left(\frac{d_c}{2m}\right)^2$: it means both two degree (in paper called half edges) can be seen as a whole edge in cluster C.

so $P(C_i) \cdot P(C_j)$ $(i, j \in C)$

② $\frac{l_c}{m}$: the ratio of edges in cluster C to the total number of edges in network

so when pluging $\left(\frac{d_c}{2m}\right)^2$ and $\frac{l_c}{m}$ into $\sum \left(\frac{X_i}{n} - \frac{\overline{X}}{n} \right)$.

the whole things can be super meaningful to show the expected possibility for the network to be a clusters

III Task II

$$\Delta Q=\left(\frac{\Sigma_{in}+2k_{i,in}}{2m}-\left(\frac{\Sigma_{tot}+k_i}{2m}\right)^2\right)-\left(\frac{\Sigma_{in}}{2m}-\left(\frac{\Sigma_{tot}}{2m}\right)^2-\left(\frac{k_i}{2m}\right)^2\right)$$

The idea of Δ is that given an isolated node i (which means it belongs to no-cluster), so if we put node i into a cluster. no other modularity will change.

The only thing we need to consider is the increase of Q in the cluster.

Given $Q=\frac{1}{2m}\left(\sum_{i,j}\left(A_{ij}-\frac{k_ik_j}{2m}\right)\right)\delta(c_i,c_j)$.

if i not in cluster C. $j \in C$.

so $Q_1=\frac{1}{2m}\left(\sum_{i,j}\left(A_{i,j}-\frac{k_ik_j}{2m}\right)\right)$ $j \in C$.

① A_{j_1,j_2} : As $j_1, j_2 \in C$

so A_{j_1,j_2} is the weight inside cluster C. $\Rightarrow \Sigma_{in}$.

$k_{i,in}$ can represent the sum of weights of links from i to nodes in C but in this case, is 0.

② $\left(\frac{k_ik_j}{2m}\right)$ is the expected weights of i to j.

probability $\left(\frac{k_i}{2m}\right)\left(\frac{k_j}{2m}\right)$ it can be seen as $\left(\left(\frac{\Sigma_{tot}+k_i}{2m}\right)\right)^2$.

Σ_{tot} is the sum of weights of links incident to nodes in C

k_i is the degree of node i (sum of weights)

so the expected incident half edge probability of i. and nodes in C is $\frac{\boxed{\Sigma_{tot}+k_i}}{2m}$.

so the probability there's a edge between i/node in C or edge between nodes in C is

$$\left(\frac{\Sigma_{tot}+k_i}{2m}\right)^2.$$

we get rid of parenthese.

$$\left(\frac{\Sigma tot}{2m}\right)^2 + \left(\frac{k_i}{2m}\right)^2 + 2\times\left(\frac{\Sigma tot}{2m}\right)\left(\frac{k_i}{2m}\right).$$

as i not in cluster C $\left(\frac{\Sigma tot}{2m}\right)$ and $\left(\frac{k_i}{2m}\right)$ independent so it's zero (can not happen at same time).

so $Q_1 = \left[\frac{\Sigma in}{2m} - \left(\frac{\Sigma tot}{2m}\right)^2 - \left(\frac{k_i}{2m}\right)^2\right]$.

same idea. if. i in cluster C.

as it's double counted

$$Q_2 = \left[\frac{\Sigma in + (\Sigma_i in)\times 2}{2m} - \left(\frac{\Sigma tot}{2m}\right)^2 - \left(\frac{k_i}{2m}\right)^2 - \frac{2\Sigma tot \cdot k_i}{4m^2}\right]$$

meaning less in Q_1, but meaningful in Q_2.

$$= \left[\frac{\Sigma in + 2(\Sigma_i in)}{2m} - \left(\frac{\Sigma tot + k_i}{2m}\right)^2\right]$$

Overall, ΔQ is the change of modularity.

= Modularity after merging − Modularity before merging.

$$\text{So. } \Delta Q = \underbrace{\left[\frac{\Sigma in + 2(\Sigma_i in)}{2m} - \left(\frac{\Sigma tot + k_i}{2m}\right)^2\right]}_{Q_2} - \underbrace{\left[\frac{\Sigma in}{2m} - \left(\frac{\Sigma tot}{2m}\right)^2 - \left(\frac{k_i}{2m}\right)^2\right]}_{Q_1}$$

It can also be simplified.

$$\Delta Q = \frac{\Sigma_i in}{m} - 2\left(\frac{\Sigma tot}{2m}\right)\left(\frac{k_i}{2m}\right) = \frac{2\Sigma_i in \cdot m - \Sigma tot \cdot k_i}{2m^2}$$

What we need to consider is only the change if an isolated node get into cluster. So the change is the modularity after minus modularity before. It can also be simplified into three terms formular.

IV Task III

The minimum modularity of unweighted network is -0.5. So here are my networks:

```
netmat=rbind(c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1),
             c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
             c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
             c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
             c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
             c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
             c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
             c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
             c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
             c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
             c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
             c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
             c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
             c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
             c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
             c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
             c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
             c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
             c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
             c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
             c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
             c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
             c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
             c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
             c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
             c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
             c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
             c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
             c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
             c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
             c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
             c(1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)
```

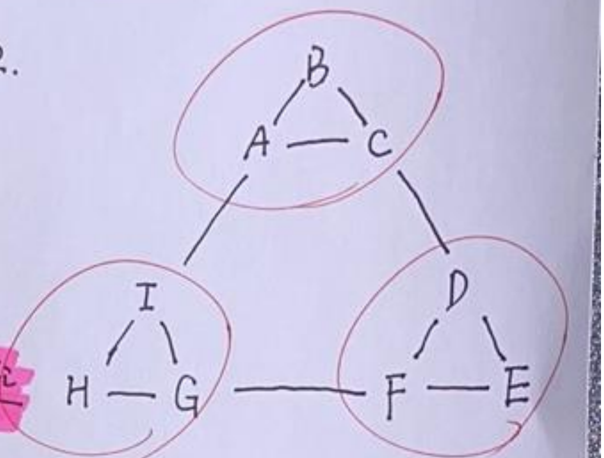
This is a 32 nodes sparsity network with only one edge

Here is the plot of it. As we can see from the plot, only node 1 and node 32 have one edge.
The modularity of the network is -0.5, satisfied the question.

V Task IV

The total weight of network m. is 12.

$\sum_{i,j} A_{ij} = 2m = 24$.



$$\Delta Q = \left(\frac{\Sigma_{in} + 2k_{i,in}}{2m} - \left(\frac{\Sigma_{tot}+k_i}{2m}\right)^2\right) - \left(\frac{\Sigma_{in}}{2m} - \left(\frac{\Sigma_{tot}}{2m}\right)^2 - \left(\frac{k_i}{2m}\right)^2\right) = \frac{2(\Sigma_{i,in})\cdot m - \Sigma_{tot}\cdot k_i}{2m^2}$$

Step1

$A \to B.$ $\left(\frac{0+2}{24} - \left(\frac{3+2}{24}\right)^2\right) - \left(\frac{0}{24} - \left(\frac{2}{24}\right)^2 - \left(\frac{3}{24}\right)^2\right) = \frac{1}{16}$

$A \to C$: $\frac{2\times 12\times 1 - 3\times 3}{2\times 12\times 12} = \frac{15}{2\times 4\times 3\times 12} = \frac{5}{96}$

$A \to I$: $\frac{2\times 12\times 1 - 3\times 3}{2\times 12\times 12} = \frac{5}{96}$

$A \Rightarrow$ other. <0 because there are no edges between them.

~~AB is max so AB can be cluster.~~

$B \to C.$ $\frac{2\times 1\times 12 - 2\times 3}{2\times 12^2} = \frac{1}{16}$

$C \to D$ $\frac{2\times 12\times 1 - 3\times 3}{2\times 12\times 12} = \frac{5}{96}$

$C \to A$ $= \frac{5}{96}$

Why do not we consider no edge between node/cluster

If no direct edge between nodes $\Sigma_{i,in}$ always be 0 (0 − degree) <0 for sure

$\boxed{\text{Same}}$ process in terms of $\begin{cases} H \to I \\ H \to G \\ G \to I \\ \text{etc. } \ldots \end{cases}$

So. after first phase. (AB). (DE). (GH). go into cluster. (second phase).

Step2:

$C \to AB.$ $\frac{2\times 12\times 2 - 5\times 3}{2\times 12\times 12} = \frac{33}{2\times 12\times 12} \approx 0.1146$

$I \to AB$ $\frac{2\times 12\times 1 - 5\times 3}{2\times 12\times 12} = \frac{9}{2\times 12\times 12} < 0.1146.$

other nodes no edges to (AB) cluster. so (ABC) can combine.

same logic (DEF) / (HIG).

Step 3: $I \to (ABC)$: $\frac{2\times 12\times 1 - 3\times(2+3+3)}{2\times 12\times 12} = 0$ not necessary.

$\boxed{\text{Same}}$ $D \to (ABC)$.

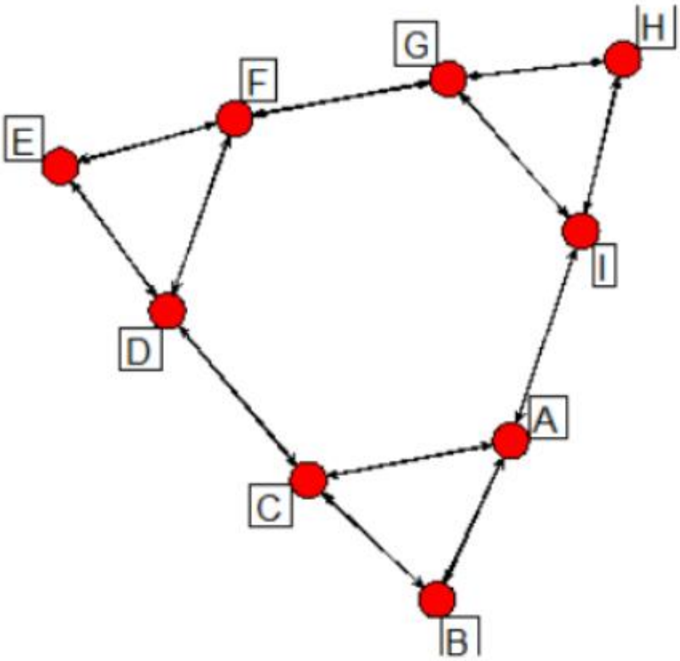
other no edges to cluster (ABC).

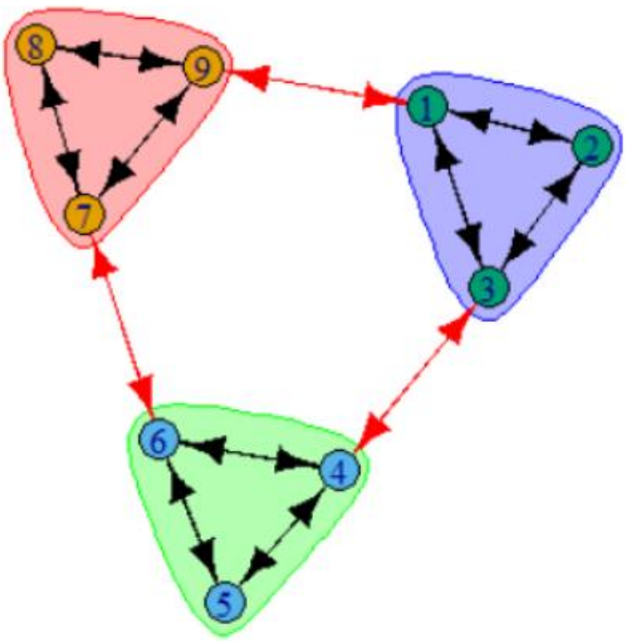
So: (ABC)-(HIG)-(DEF) are the final result of cluster.

6

In part 4, we only used three steps to make the network converge. Each step has two phases. There are also many calculations same, so I didn't show them up.

The modularity of the final network is 0.4166666. And here is the graph of it:





This one showed how it divided into three clusters. The result same with what I wrote on paper.

VI Appendix

````
---

title: "Network_Midterm"

author: "Chenrui Xu"

date: "2021/3/14"

output: html_document

---

```{r}

library(igraph)

library(intergraph)

library(UserNetR)

library(statnet)

```

```{r}

set.seed(999)

netmat=rbind(c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1),

             c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),

             c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),

             c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),

             c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),

             c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
````
````

```
c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),
```

```
                    c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),

                    c(1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)

                    )

net1=network(netmat,matrix.type='adjacency',directed=F)

```
```

```{r}

inet <- asIgraph(net1)

cw=cluster_walktrap(inet)

modularity(inet,V(inet))

plot(cw,inet)

```
```

```{r}


netmat3=rbind(c(1,200))

net3=network(netmat3,type="edgelist")

inet3 <- asIgraph(net3)

cw3=cluster_walktrap(inet3)

modularity(inet3,V(inet3))

plot(cw3,inet3)

```
```
```

Part 4

```{r}
netmat2<-rbind(c(0,1,1,0,0,0,0,0,1),
               c(1,0,1,0,0,0,0,0,0),
               c(1,1,0,1,0,0,0,0,0),
               c(0,0,1,0,1,1,0,0,0),
               c(0,0,0,1,0,1,0,0,0),
               c(0,0,0,1,1,0,1,0,0),
               c(0,0,0,0,0,1,0,1,1),
               c(0,0,0,0,0,0,1,0,1),
               c(1,0,0,0,0,0,1,1,0))
rownames(netmat2)<-c("A","B","C","D","E","F","G","H","I")
colnames(netmat2)<-c("A","B","C","D","E","F","G","H","I")
net2=network(netmat2,type="adjacency")
```

```{r}
plot(net2,displaylabels=T,vertex.cex=3,vertex.col="red",boxed.labels=T,labels.pos=4)
```

```{r}
inet2=asIgraph(net2)
cw2=cluster_walktrap(inet2)
```

modularity(cw2)

plot(cw2,inet2)

```
```