**Network Midterm**

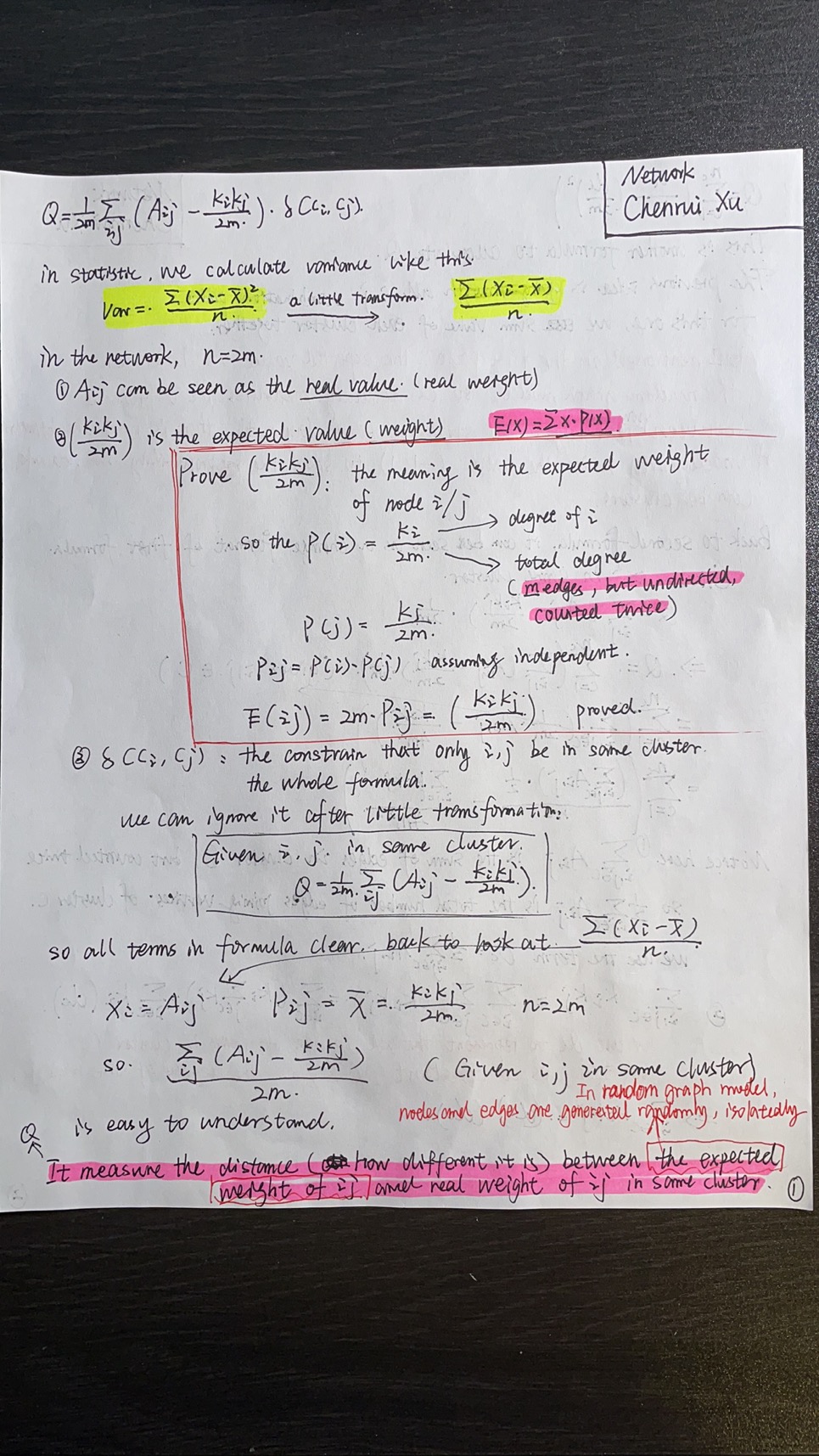
**Chenrui Xu**

**I Background**

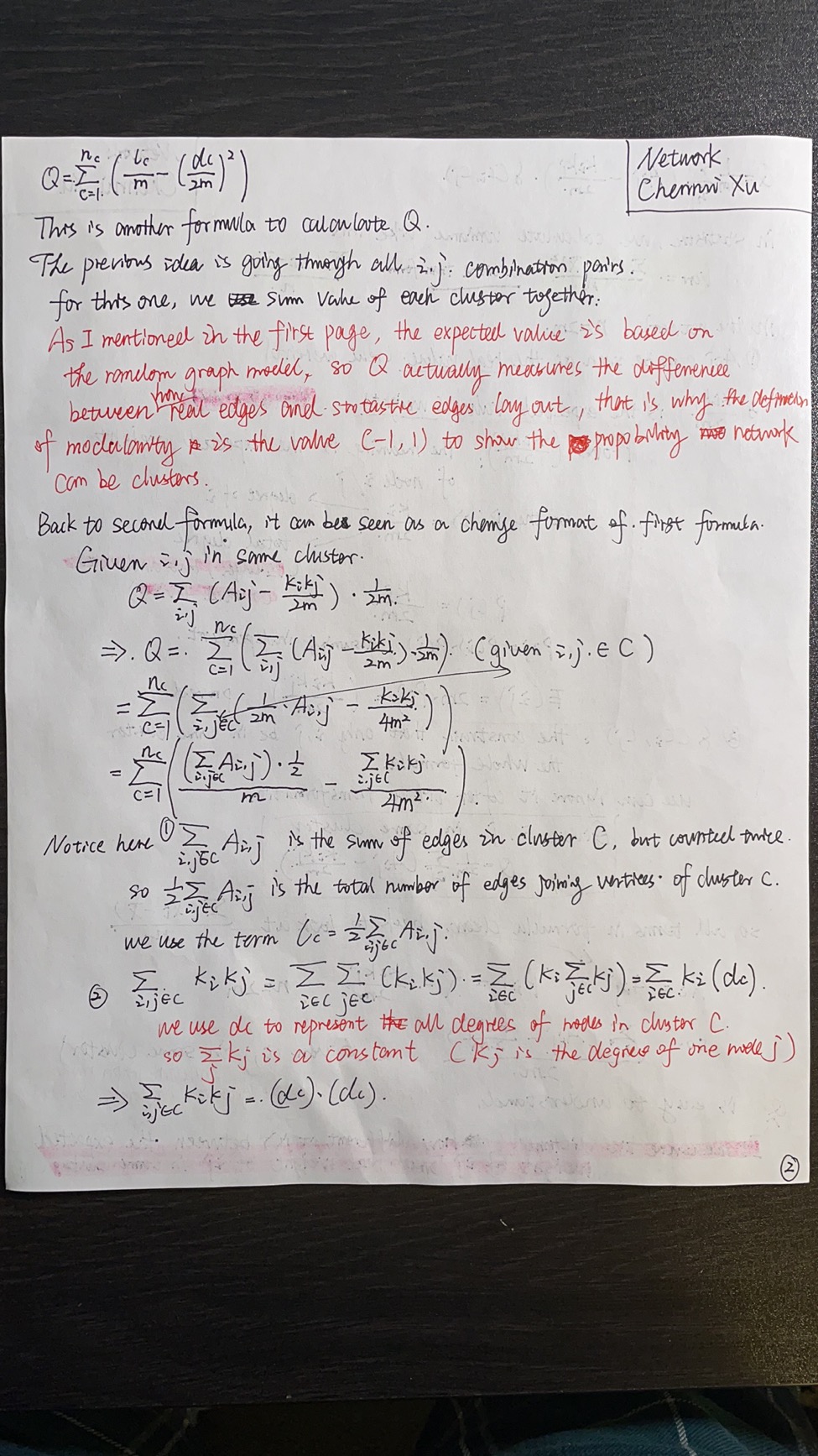
Community detection is one of the most important areas in network analysis. It has many kinds of applications in the real world. For example, to detect some potential group of people, detect a cluster of things one person might be interested in so that the system can do the recommendation or some other 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.

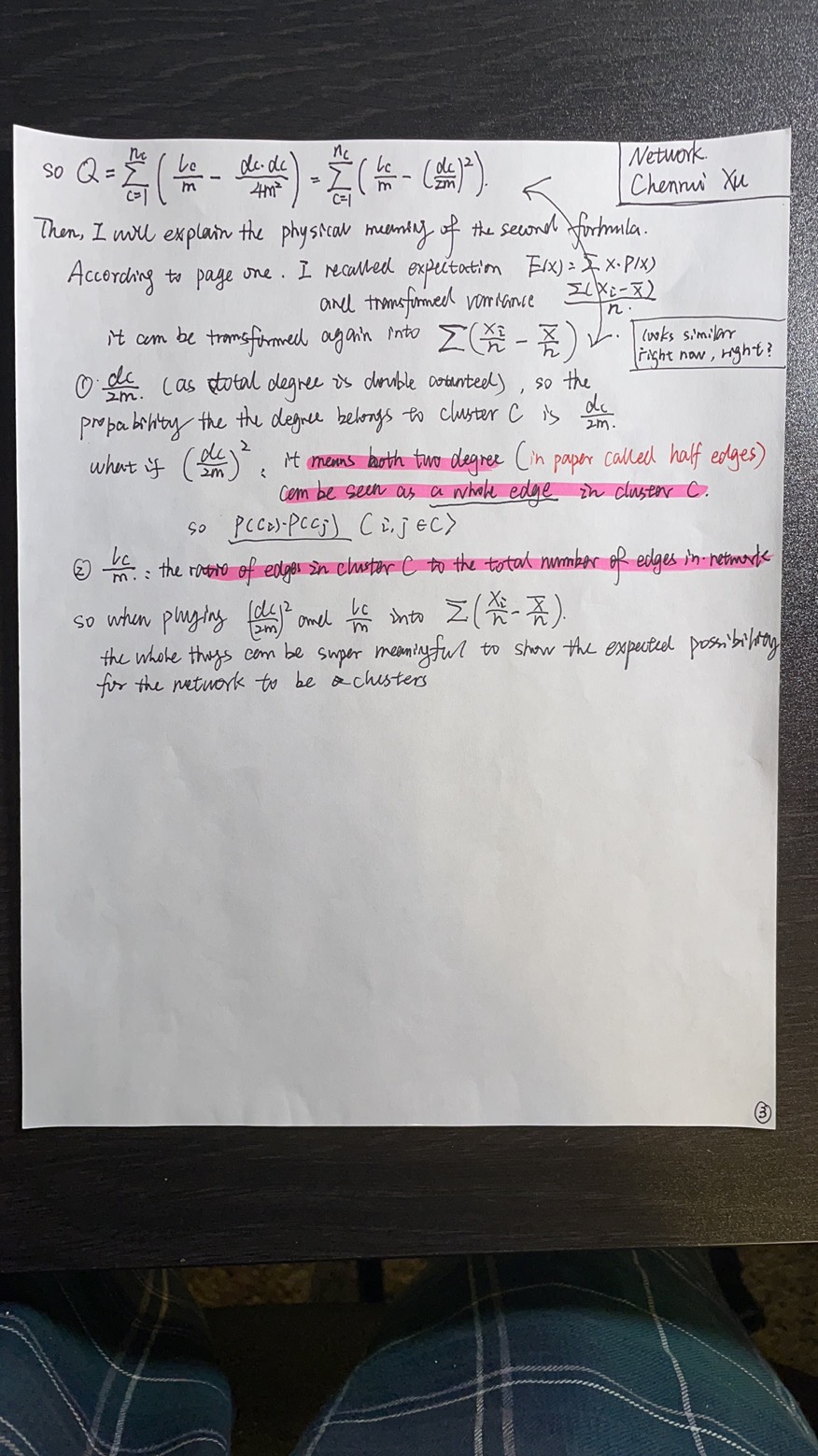
**II Task I**

****

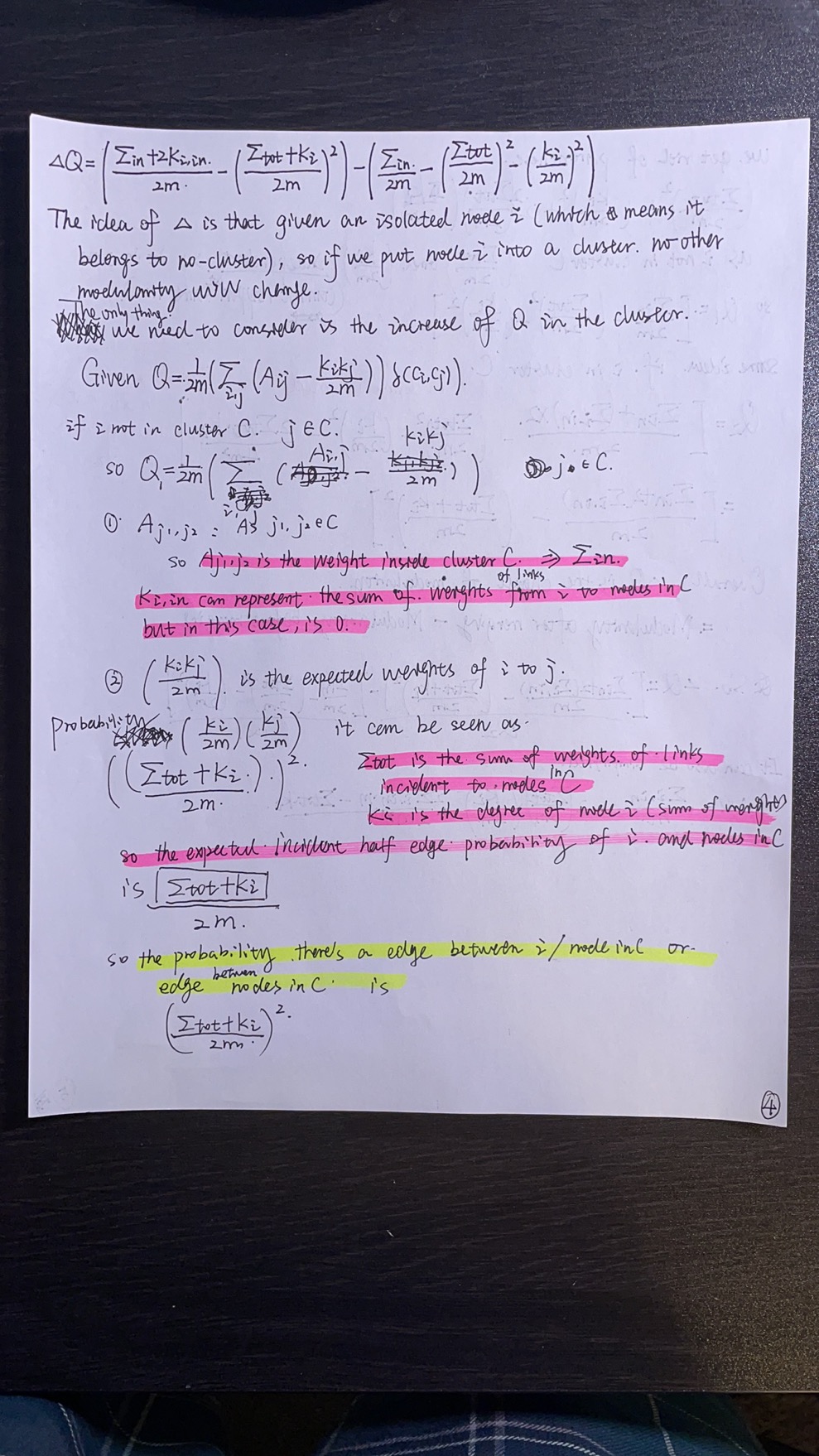
Q can be seen as how different (I will consider it as one format of variance with no L1/L2 regularity) the real model and random graph model is in terms of existence of edges. That is why Q is the statistics of modularity.

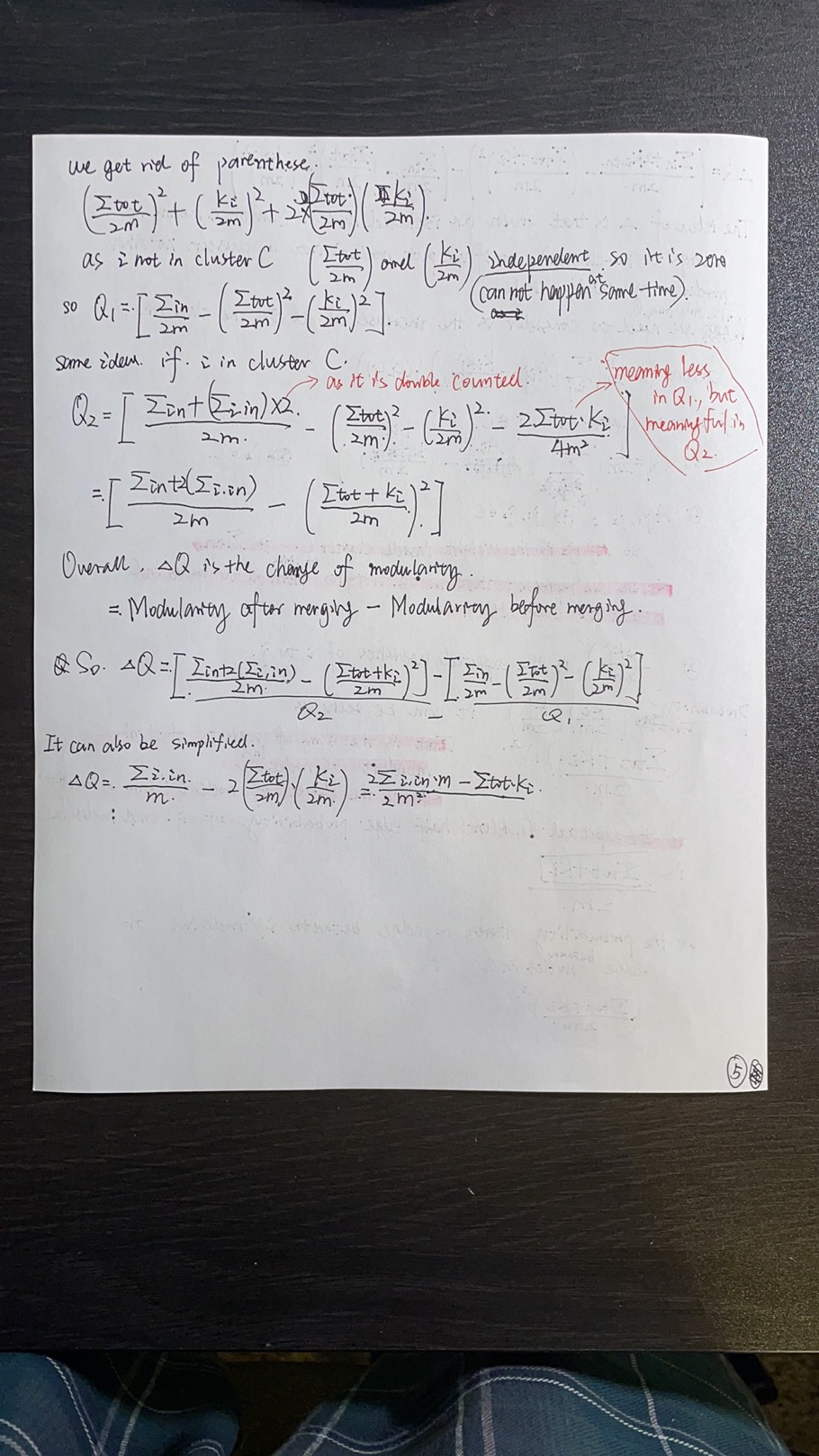
****

Here is the process why two equations equals.

****

**III Task II**

****

****

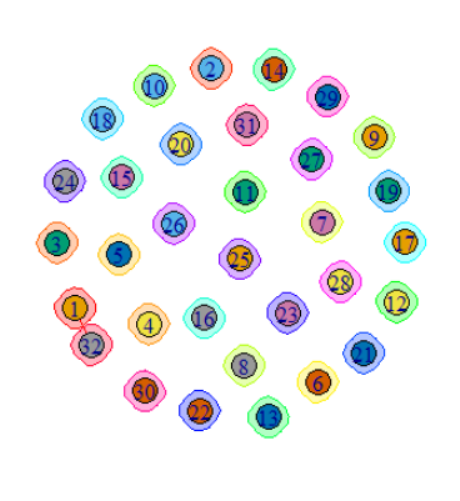
What we need to consider is only the change if an isolated node get into cluster. So the change is the modularity after minus modularity before. It can also be simplified into three terms formular.

**IV Task III**

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



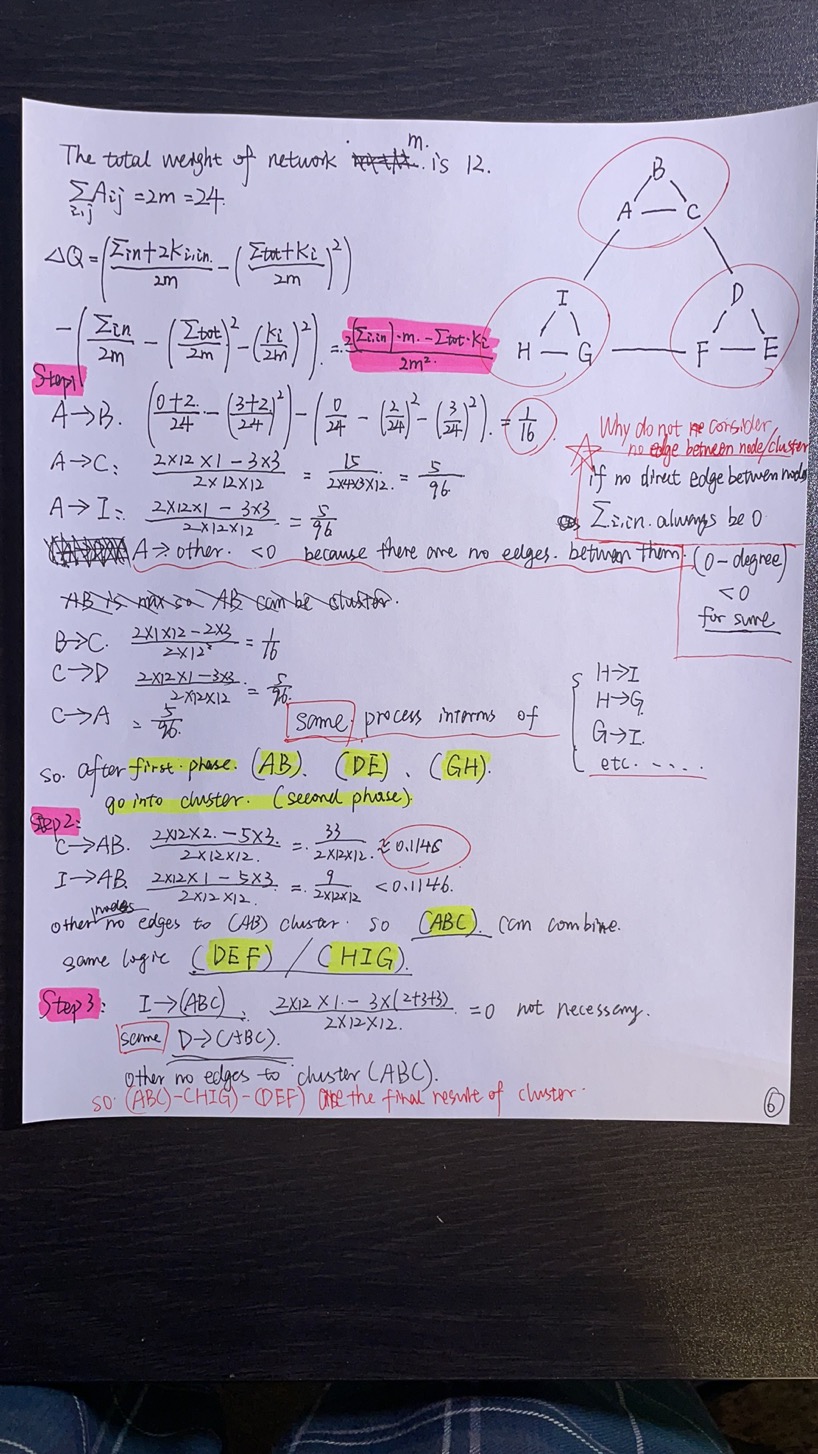
This is a 32 nodes sparsity network with only one edge



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.

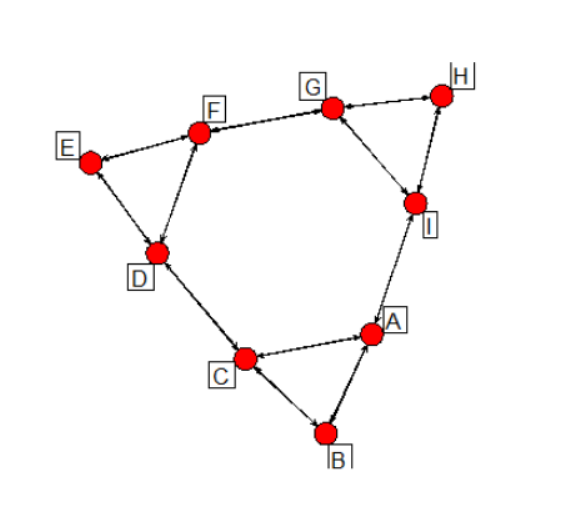
**V Task IV**

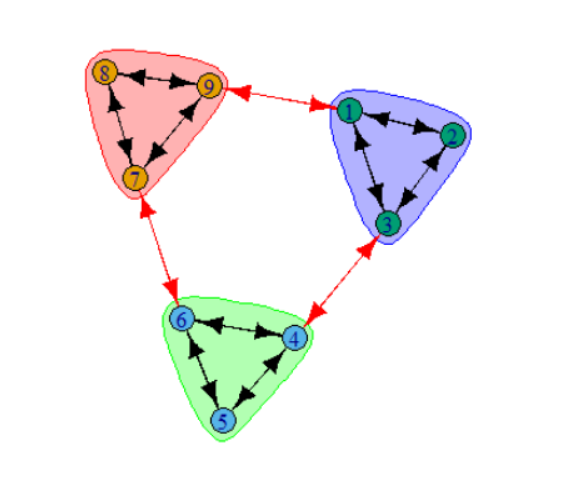
****

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),**

**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)**

**)**

**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)**

**```**