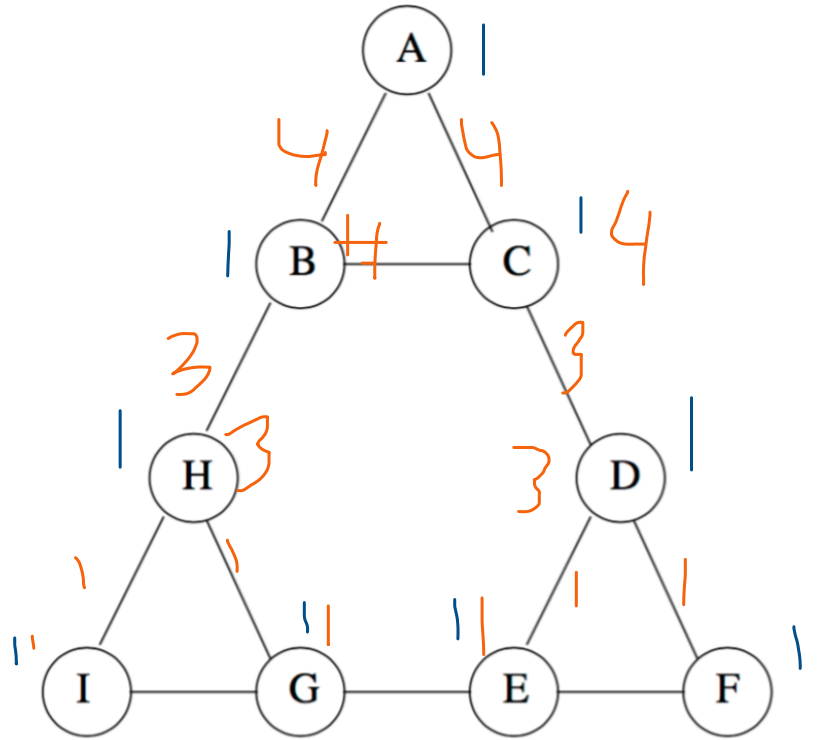




1.

a.

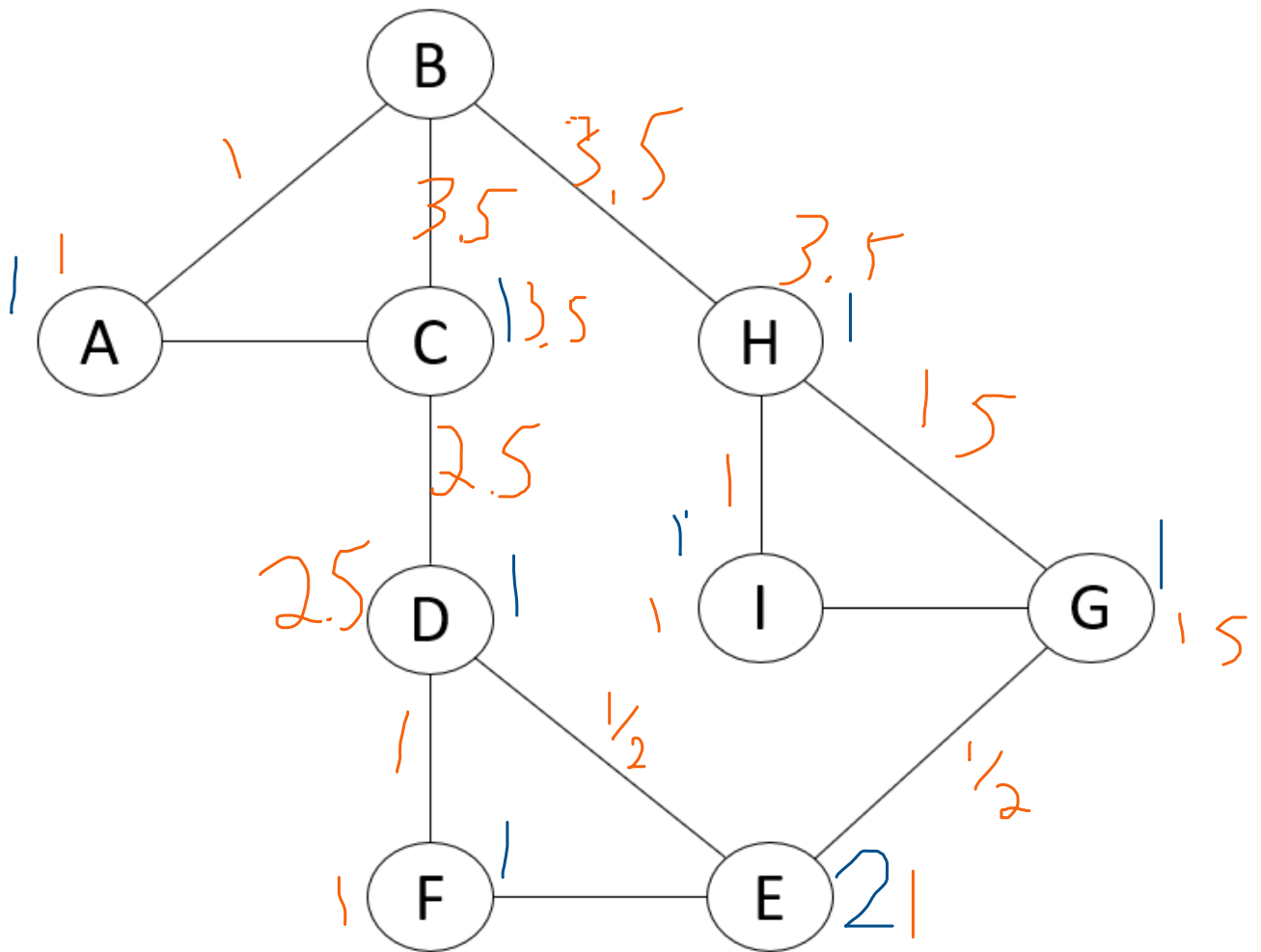


 = number of shortest paths to node

 = number of shortest paths through node/edge

Edge	Credits per edge
AB	4
AC	4
BC	0
BH	3
CD	3
HI	1
HG	1
IG	0
GE	0
DE	1
DF	1
EF	0

b.

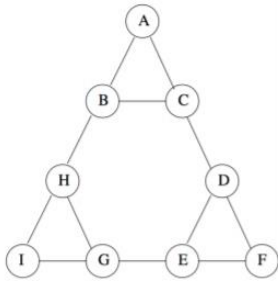


= number of shortest paths to node

= number of shortest paths through node/edge

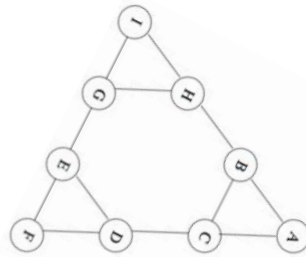
Edge	Credits per edge
BA	1
AC	0
BC	3.5
BH	3.5
CD	2.5
HI	1
HG	1.5
IG	0
DF	1
DE	0.5
FE	0
EG	0.5

c. I and F are symmetric to A



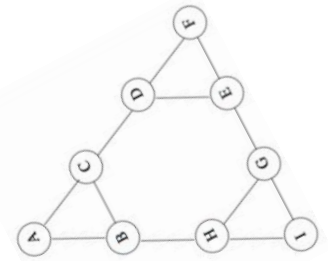
A:

Edge	Credits per edge
AB	4
AC	4
BC	0
BH	3
CD	3
HI	1
HG	1
IG	0
GE	0
DE	1
DF	1
EF	0



I:

Edge	Credits per edge
IG	4
IH	4
GH	0
GE	3
BH	3
EF	1
ED	1
FD	0
CD	0
BC	1
BA	1
AC	0



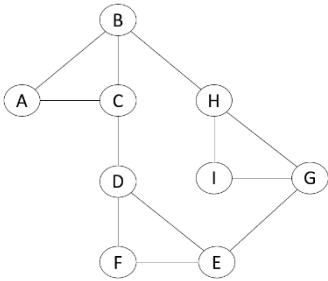
F:

Edge	Credits per edge
FD	4
FE	4
DE	0
DC	3
EG	3
CA	1
CB	1
AB	0
BH	0
GH	1
GI	1
HI	0

C, H, G, E, and D are symmetric to B

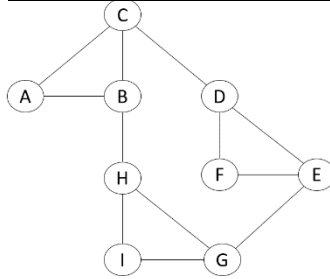
B:

Edge	Credits per edge
BA	1
AC	0
BC	3.5
BH	3.5
CD	2.5
HI	1
HG	1.5
IG	0
DF	1
DE	0.5
FE	0
EG	0.5



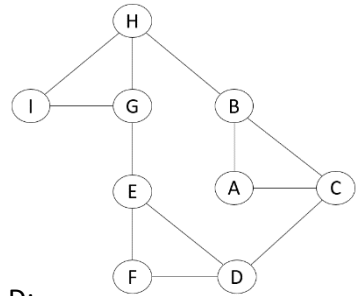
C:

Edge	Credits per edge
AC	1
AB	0
BC	3.5
CD	3.5
BH	2.5
DF	1
DE	1.5
EF	0
HI	1
HG	0.5
IG	0
GE	0.5



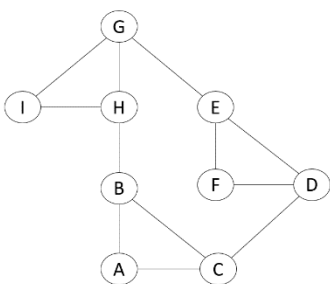
H:

Edge	Credits per edge
HI	1
IG	0
HG	3.5
HB	3.5
GE	2.5
BA	1
BC	1.5
AC	0
EF	1
ED	0.5
FD	0
DC	0.5



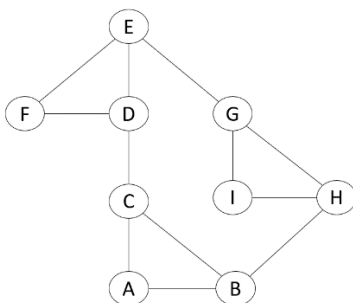
G:

Edge	Credits per edge
GI	1
IH	0
GH	3.5
GE	3.5
HB	2.5
EF	1
ED	1.5
FD	0
BA	1
BC	0.5
AC	0
DC	0.5



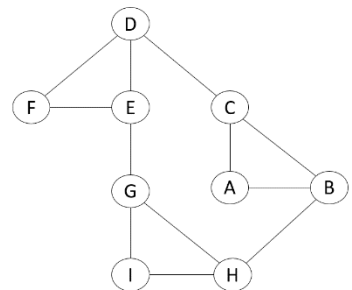
E:

Edge	Credits per edge
EF	1
FD	0
ED	3.5
EG	3.5
DC	2.5
GI	1
GH	1.5
IH	0
CA	1
CB	0.5
AB	0
HB	0.5



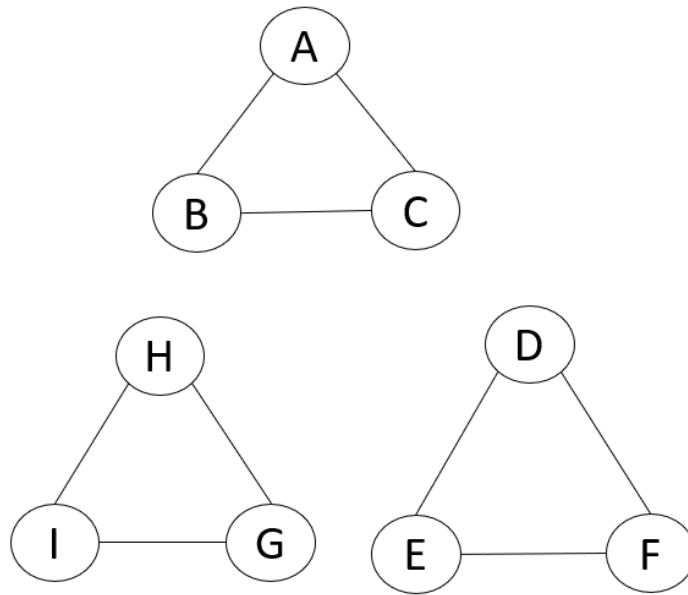
D:

Edge	Credits per edge
DF	1
FE	0
DE	3.5
DC	3.5
EG	2.5
CA	1
CB	1.5
AB	0
GI	1
GH	0.5
IH	0
BH	0.5



Edge	Sum of credits	Betweenness
AB	8	4
AC	8	4
BC	13	6.5
BH	19	9.5
CD	19	9.5
DE	13	6.5
DF	8	4
EF	8	4
EG	19	9.5
GH	13	6.5
GI	8	4
HI	8	4

- d.
Remove all edges with betweenness > 7



2.

a.

$$B_{ij} = A_{ij} - \frac{d_i d_j}{2m}$$

$$\sum_j B_{ij} = \sum_j A_{ij} - \frac{d_i d_j}{2m} = \sum_j A_{ij} - \sum_j \frac{d_i d_j}{2m}$$

$$\sum_j B_{ij} = (\text{Sum of \# edges leaving } i) - d_i \sum_j \frac{d_j}{2m}$$

$$\sum_j B_{ij} = d_i - d_i (1)$$

$$\boxed{\sum_j B_{ij} = 0}$$

$$B_{ij} = A_{ij} - \frac{d_i d_j}{2m}$$

$$\sum_i B_{ij} = \sum_i A_{ij} - \frac{d_i d_j}{2m} = \sum_i A_{ij} - \sum_i \frac{d_i d_j}{2m}$$

$$\sum_i B_{ij} = (\text{Sum of \# edges entering } j) - d_j \sum_i \frac{d_i}{2m}$$

$$\sum_i B_{ij} = d_j - d_j (1)$$

$$\boxed{\sum_i B_{ij} = 0}$$

b.

A =

	A	B	C	D	E	F
A	0	1	1	0	0	0
B	1	0	1	0	0	0
C	1	1	0	1	0	0
D	0	0	1	0	1	1
E	0	0	0	1	0	1
F	0	0	0	1	1	0

$$B_{ij} = A_{ij} - \frac{d_i d_j}{2m}$$

B =

	A	B	C	D	E	F
A	-2/7	5/7	4/7	-3/7	-2/7	-2/7
B	5/7	-2/7	4/7	-3/7	-2/7	-2/7
C	4/7	4/7	-9/14	5/14	-3/7	-3/7
D	-3/7	-3/7	5/14	-9/14	4/7	4/7
E	-2/7	-2/7	-3/7	4/7	-2/7	5/7
F	-2/7	-2/7	-3/7	4/7	5/7	-2/7

c. Using MATLAB to get eigen values and eigen vectors:

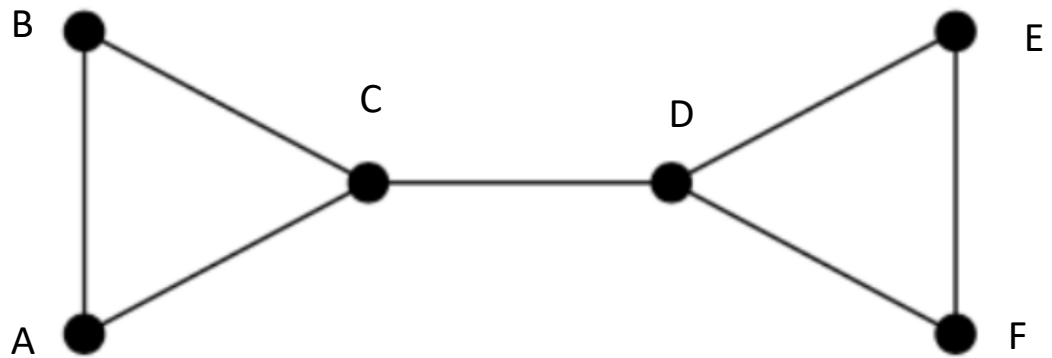
Largest eigen value = 1.7321

Corresponding eigen vector =

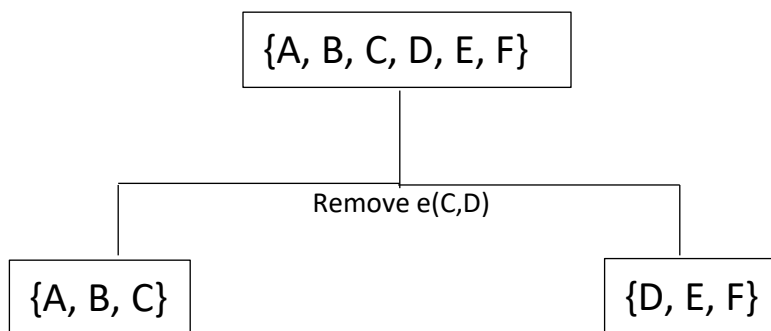
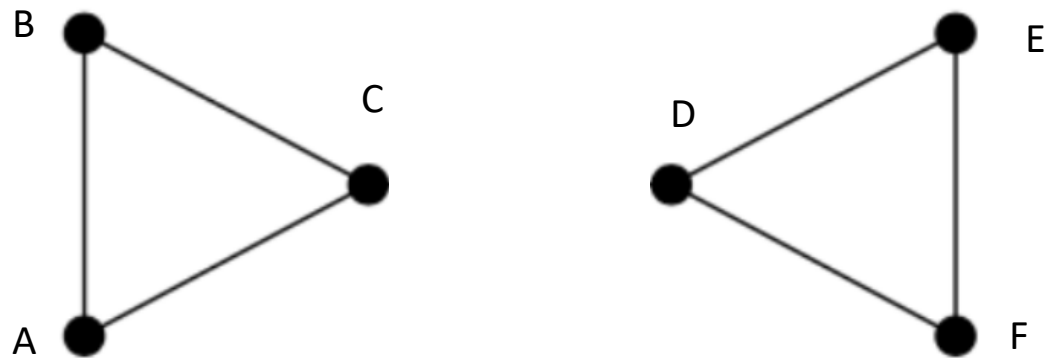
-0.444
-0.444
-0.3251
0.3251
0.444
0.444

d.

Original Network:



After split:



3.

a.

karate has 34 nodes
karate has 78 edges
karate has 2.40819964349 avg path length
karate has 0.570638478208 avg clustering

dolphins has 62 nodes
dolphins has 159 edges
dolphins has 3.3569539926 avg path length
dolphins has 0.258958246055 avg clustering

jazz has 198 nodes
jazz has 2742 edges
jazz has 2.23504076296 avg path length
jazz has 0.617450702154 avg clustering

- b. repository name: homework-4-terrylu_kinaanpatel
- c. repository name: homework-4-terrylu_kinaanpatel
- d. repository name: homework-4-terrylu_kinaanpatel

e. Problem 3 part e

For the karate dataset ,

If we use betweenness based clustering,

Number of cluster found is 2

Modularity score is 74.8974358974

Time to complete is 0.0754609107971 seconds

If we use modularity based clustering

Number of cluster found is 4

Modularity score is 82.4102564103

Time to complete is 0.0586040019989 seconds

If we use spectral clustering

Number of cluster found is 3

Modularity score is 64.0256410256

Time to complete is 0.0482380390167 seconds

For the dolphin dataset

If we use betweenness based clustering,

Number of cluster found is 5

Modularity score is 193.962264151

Time to complete is 0.574102878571 seconds

If we use modularity based clustering

Number of cluster found is 3

Modularity score is 185.106918239

Time to complete is 0.207121133804seconds

If we use spectral clustering

Number of cluster found is 3

Modularity score is 136.0

Time to complete is 0.154905080795seconds

Jazz part b :

time to complete was 187.324069023 seconds

number of cluster found is 39

modularity score is 2918.47921225

jazz part c

2 is the num of clusters

2637.38657914 is the modularity score

time to complete was 2.40068697929 seconds

jazz part d

3 is the num of clusters

2602.32822757 is the modularity score

time to complete was 3.78681612015 seconds

The best algorithm is **modularity based clustering** because it is consistently the either the fastest or barely the second fastest algorithm, and it has the best or second best modularity score each time