



=number of shortest paths to node

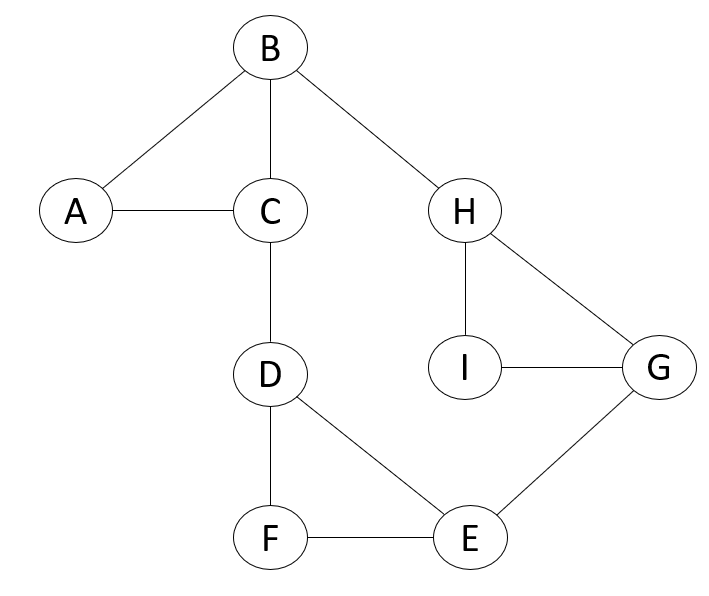


= number of shortest paths through node/edge



|  |  |
| --- | --- |
| 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 |







=number of shortest paths to node

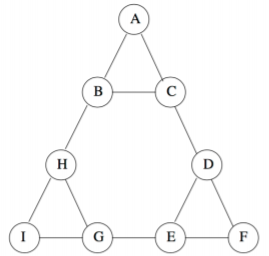


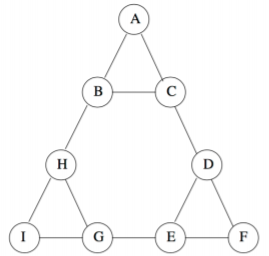
= number of shortest paths through node/edge

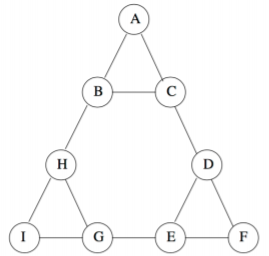


|  |  |
| --- | --- |
| Edge | Credits per edge |
| BA | 1 |
| AC | 0 |
| BC | 3.5 |
| BH | 3.5 |
| CD | 2.5 |
| HI | 1 |
| HG | 1.5 |
| IG | 0 |
| DF | 1 |
| DE | 0.5 |
| FE | 0 |
| EG | 0.5 |

* 1. 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 |

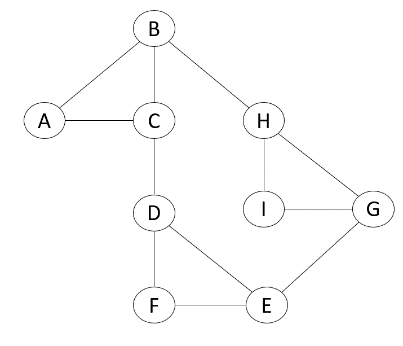
F:

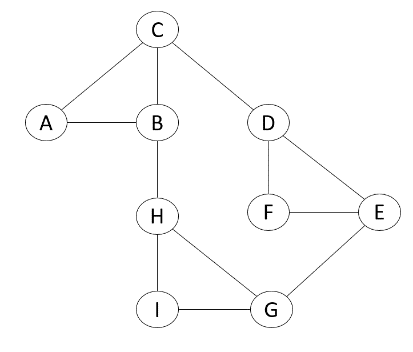
|  |  |
| --- | --- |
| Edge | Credits per edge |
| FD | 4 |
| FE | 4 |
| DE | 0 |
| DC | 3 |
| EG | 3 |
| CA | 1 |
| CB | 1 |
| AB | 0 |
| BH | 0 |
| GH | 1 |
| GI | 1 |
| HI | 0 |

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 |

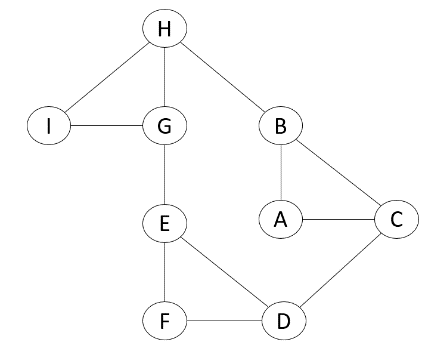


C:

|  |  |
| --- | --- |
| 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 |

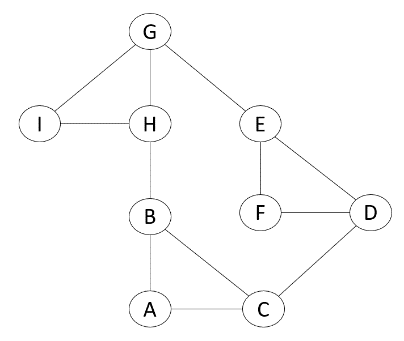
H:

|  |  |
| --- | --- |
| 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 |



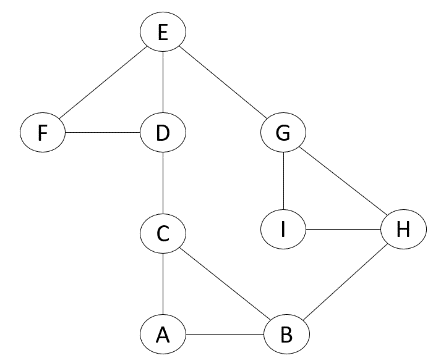
G:

|  |  |
| --- | --- |
| 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 |



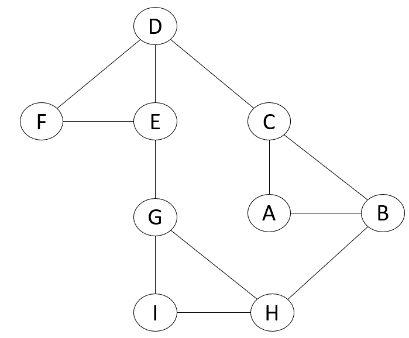
E:

|  |  |
| --- | --- |
| 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:

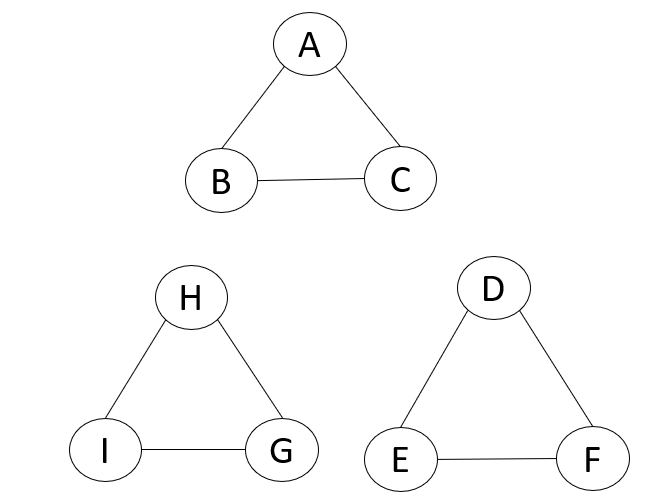
|  |  |
| --- | --- |
| Edge | Credits per edge |
| DF | 1 |
| FE | 0 |
| DE | 3.5 |
| DC | 3.5 |
| EG | 2.5 |
| CA | 1 |
| CB | 1.5 |
| AB | 0 |
| GI | 1 |
| GH | 0.5 |
| IH | 0 |
| BH | 0.5 |



|  |  |  |
| --- | --- | --- |
| 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 |

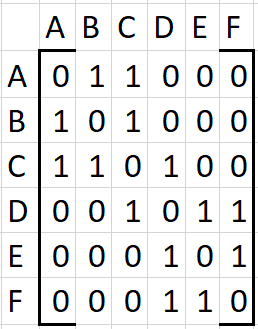


Remove all edges with betweenness > 7

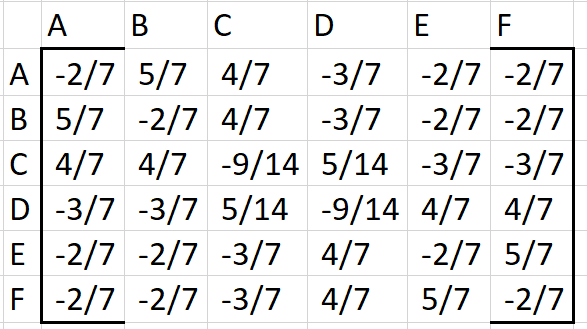




A =



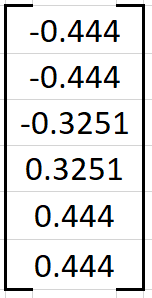
B=



* 1. Using MATLAB to get eigen values and eigen vectors:

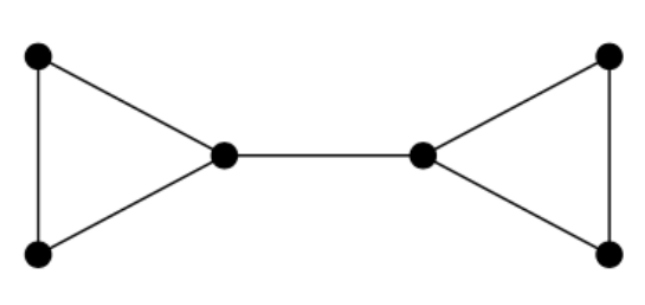
Largest eigen value = 1.7321

Corresponding eigen vector =





Original Network:



A

F

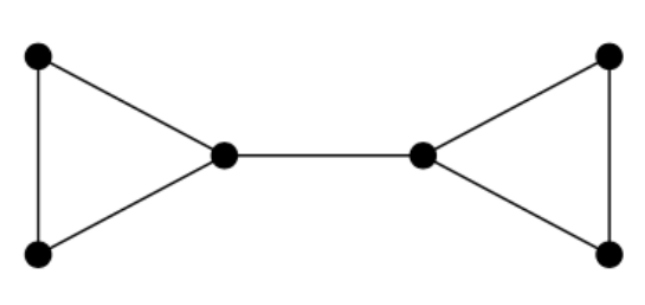
E

D

C

B

After split:



A

B

C

D

E

F

Remove e(C,D)

{D, E, F}

{A, B, C}

{A, B, C, D, E, F}



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

* 1. repository name: homework-4-terrylu\_kinaanpatel
  2. repository name: homework-4-terrylu\_kinaanpatel
  3. repository name: homework-4-terrylu\_kinaanpatel
  4. Problem 3 part e

For the karate dataset ,

If we use betweeness based clustering,

Number of cluster found is 2

Modularity score is 0.480111768573

Time to complete is 0.0754609107971 seconds

If we use modularity based clustering

Number of cluster found is 4

Modularity score is 0.528270874425

Time to complete is 0.0586040019989 seconds

If we use spectral clustering

Number of cluster found is 3

Modularity score is 0.410420775805

Time to complete is 0.0482380390167 seconds

For the dolphin dataset

If we use betweeness based clustering,

Number of cluster found is 5

Modularity score is 0.60994422689

Time to complete is 0.574102878571 seconds

If we use modularity based clustering

Number of cluster found is 3

Modularity score is 0.582097227167

Time to complete is 0.207121133804seconds

If we use spectral clustering

Number of cluster found is 3

Modularity score is 0.427672955975

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 0.532180746217

jazz part c

2 is the num of clusters

0.480923883869 is the modularity score

time to complete was 2.40068697929 seconds

jazz part d

3 is the num of clusters

0.480554265416 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 ore barely the second fastest algorithm, and it has the best or second best modularity score each time