## COMP9312 Assignment 2

## Written by Tianwei Mo

* 1. The two graphs are isomorphic. Here is the node mapping from figure 1 to figure 2: {(1, 8), (2, 2), (3, 3), (4, 5), (5, 6), (6, 7), (7, 1), (8, 4)}.

Betweenness centrality:

c\_A = 0. No shortest path that not starts from or ends at A pass through A.

c\_D = 5. Paths are {(A-B-D-F), (A-B-D-E), (A-B-D-G), (A-B-D-G-I), (A-B-D-G-H)}.

c\_G = 2. Paths are {(A-B-D-G-I), (A-B-D-G-H)}

Closeness centrality:

c\_A = 1 / (1 + 1 + 2 + 3 + 3 + 3 + 4 + 4) = 1/21. Paths are {(A-B), (A-C), (A-B-D), (A-B-D-E), (A-B-D-F), (A-B-D-G), (A-B-D-G-H), (A-B-D-G-I)}

c\_D = 1 / (2 + 1 + 1 + 1 + 1 + 1 + 2 + 2) = 1/11. Paths are {(D-B-A), (D-B), (D-C), (D-E), (D-F), (D-G), (D-G-H), (D-F-I)}

c\_G = 1 / (3 + 2 + 2 + 1 + 1 + 1 + 1 + 1) = 1/12. Paths are {(G-D-B-A), (G-D-B), (G-D-C), (G-D), (G-F), (G-E), (G-H), (G-I)}

* 1. Assign initial colors

有钟表的地图

描述已自动生成

Aggregate neighboring colors

图示

描述已自动生成

Hash aggregated colors

图片包含 文具, 信封, 小, 桌子

描述已自动生成

Aggregated colors

图片包含 钟表, 信封, 文具, 小

描述已自动生成

Hash aggregated colors

图片包含 小, 室内, 桌子, 大

描述已自动生成

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13]

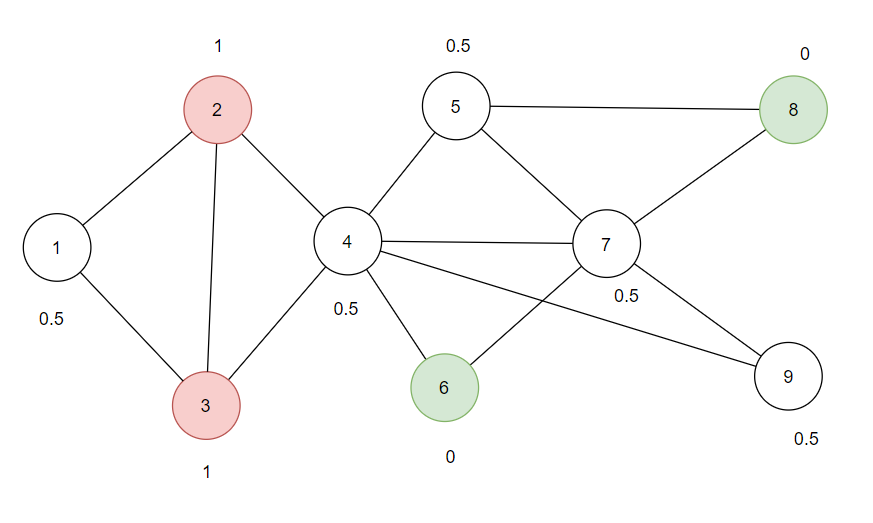
Phi(left) = [8, 3, 4, 0, 1, 3, 0, 2, 2, 0, 0, 1, 0]

Phi(right) = [8, 4, 2, 1, 1, 4, 1, 0, 0, 1, 1, 0, 1]

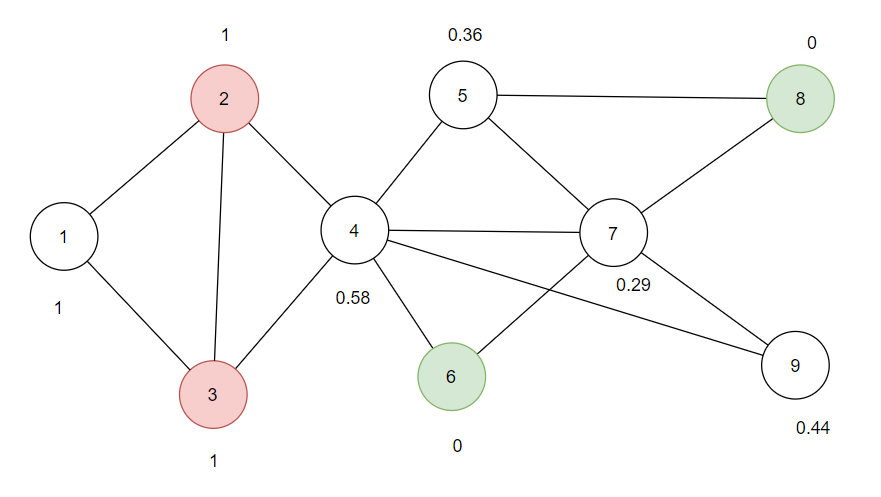
K(left, right) = phi(left)^T \* phi(right) = 97

2.1.

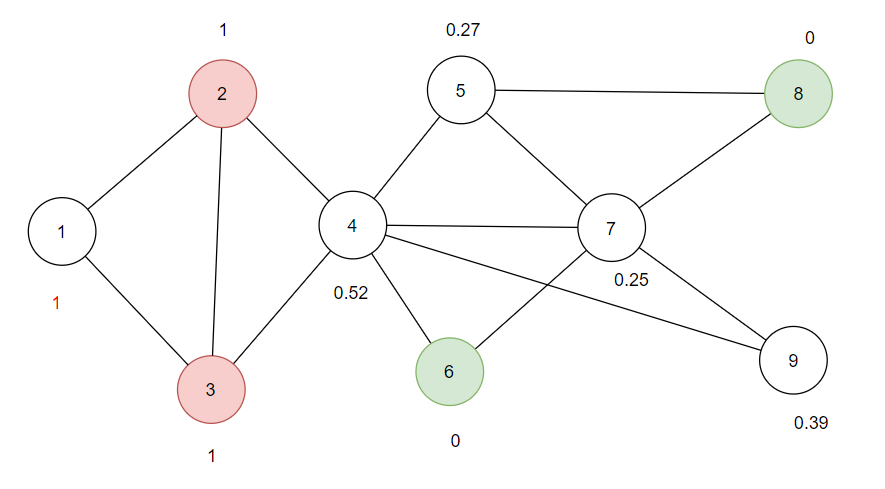
Initialization: set 2 and 3 with label 1, and 6 and 8 with label 0. Unlabeled nodes are set with label 0.5.



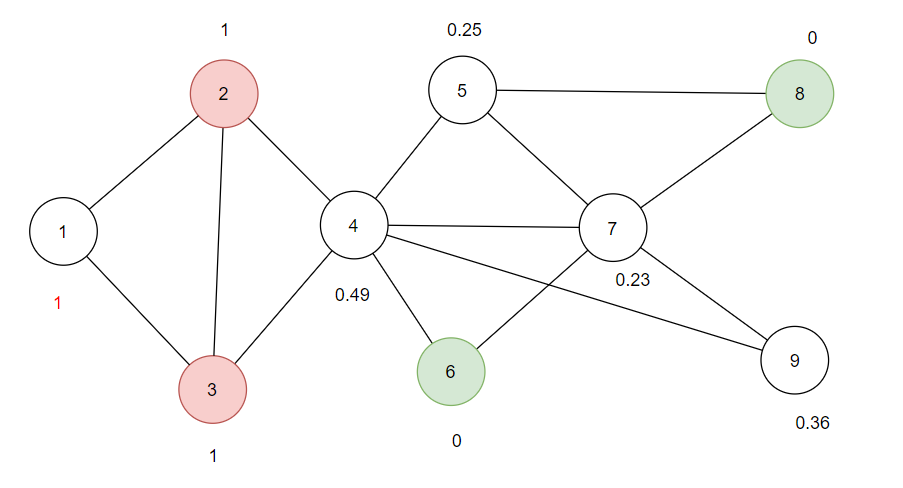
First iteration. Unlabeled nodes are always updated in node order. i.e., in the order: {1, 4, 5, 7, 9}



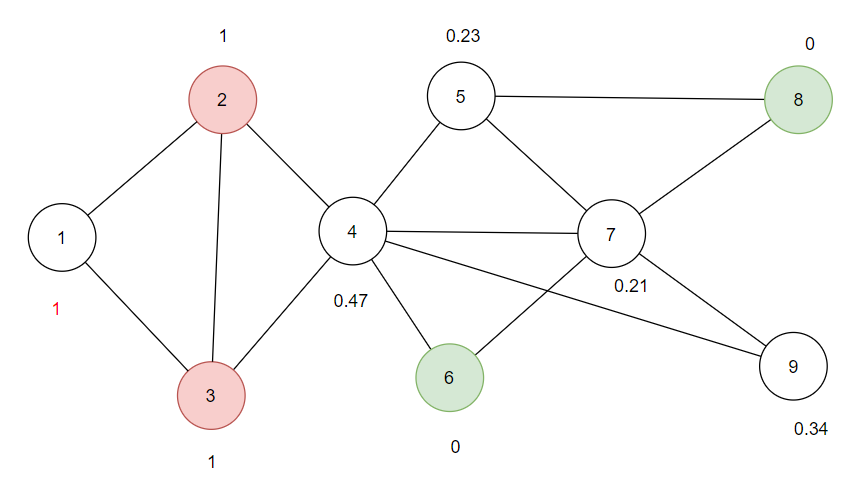
After second iteration, node 1 is converged.



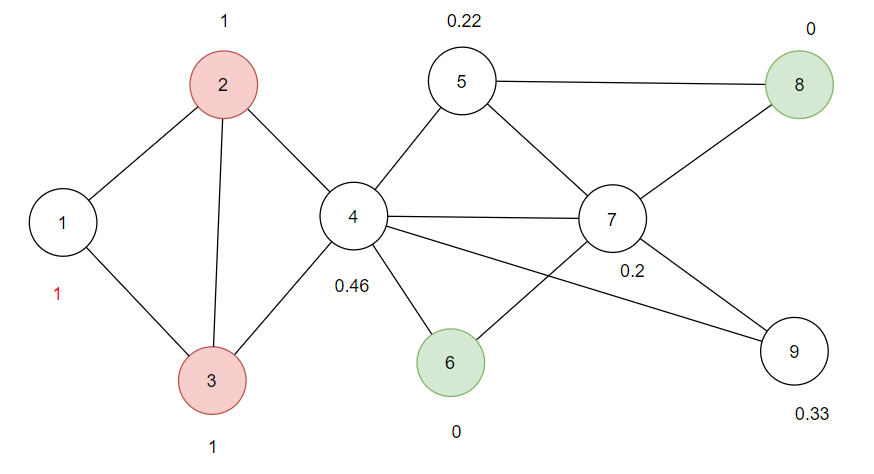
Iteration 3.



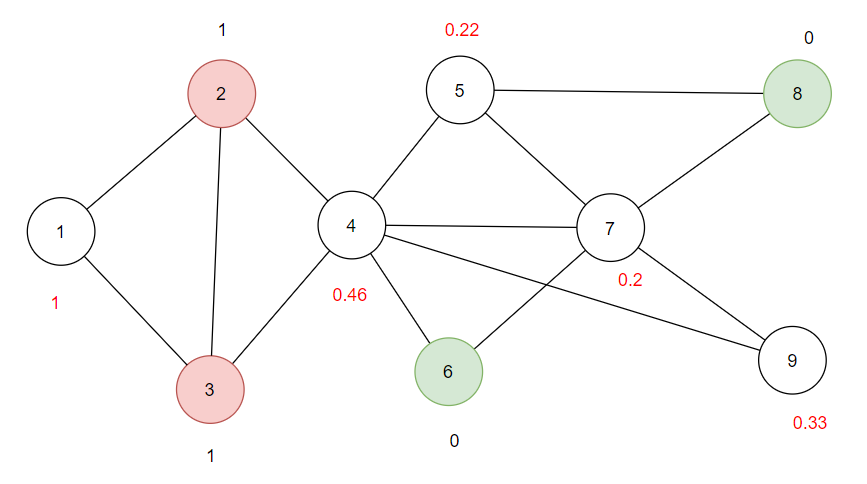
Iteration 4.



Iteration 5.



Iteration 6.



After seventh iteration, all the node labels converged.

2.2.

Probability of node 3 is 1/4 since edge (3, 4) were just traversed, the probability to go back is 1/p.

Probability of node 2 is 1 since distance between 3 and 2 is the same as from 3 to 4.

Probability of node 5, 6, 7, 9 are 1/3 since they are all further away from node 3 by 1, so the probability to explore is 1/q.

|  |  |  |
| --- | --- | --- |
| Target t | Probability | Dist(3, t) |
| 3 | 1/4 | 0 |
| 2 | 1 | 1 |
| 5 | 1/3 | 2 |
| 6 | 1/3 | 2 |
| 7 | 1/3 | 2 |
| 9 | 1/3 | 2 |

3.

I calculated it with Python

H1 = [[0.5000, 1.3000, 0.7000, 0.9000],

[1.3333, 1.2667, 1.5000, 0.5333],

[1.8667, 0.0667, 0.9667, 0.8667],

[1.5000, 0.3500, 1.3000, 0.6500],

[1.6333, 0.2333, 1.4333, 0.1000],

[1.0000, 1.3000, 1.4000, 0.7000],

[0.8200, 0.9400, 1.4600, 0.1800],

[0.9500, 0.1500, 0.9000, 0.2000],

[1.0000, 1.0000, 2.0000, -0.0000]]

4.1.

Sim(u4, u5) = cos(u4, u5) = 0.937

Sim(u2, u7) = cos(u2, u7) = 0.719

4.2.

If t = 0.6, it does not correctly predict edges as similarity between u2 and u7 is higher than the threshold, but there is no edges between them. If t = 0.8, it correctly predict edges as similarity between u2 and u7 does not reach to the threshold and there is no edge between them. Similarity between u4 and u5 is higher than the threshold, and there exist an edge between them.