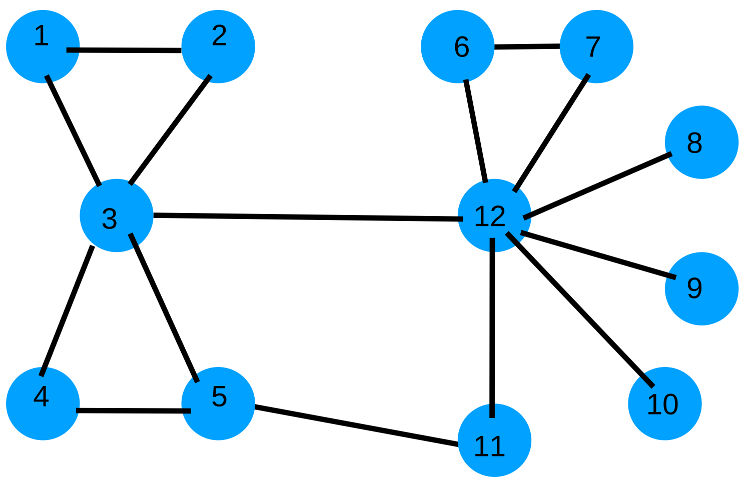
Name(s): ______________________________

Homework 3: CSCI 347: Data Mining

Show your work. Include any code snippets you used to generate an answer, using comments in the code to clearly indicate which problem corresponds to which code.

Consider the following graph:



1. [3 points] Without using networkx or other graph analysis packages (though you may use the to check your answer), find the closeness centrality of vertices 3 and 12.

Closeness $c(v_i)$ **of a vertex** v_i **is defined as** $c(v_i) = \frac{1}{\sum_j d(v_i, v_j)}$**, where** $d(v_i, v_j)$ **are the shortest path distances between vertices** v_i **and** v_j**.**

Therefore the closeness of vertex 3 is:

$$c(v_3) = \frac{1}{\sum_j d(v_3, v_j)}$$

$$= \frac{1}{d(v_3,v_1)+d(v_3,v_2)+d(v_3,v_4)+d(v_3,v_5)+d(v_3,v_6)+d(v_3,v_7)+d(v_3,v_8)+d(v_3,v_9)+d(v_3,v_{10})+d(v_3,v_{11})+d(v_3,v_{12})}$$

$$= \frac{1}{1+1+1+1+2+2+2+2+2+2+1}$$

$$= \frac{1}{17} = 0.059$$

The closeness of vertex 12 is:

$$c(v_{12}) = \frac{1}{\sum_j d(v_{12}, v_j)}$$

$$= \frac{1}{d(v_{12}, v_1) + d(v_{12}, v_2) + d(v_{12}, v_3) + d(v_{12}, v_4) + d(v_{12}, v_5) + d(v_{12}, v_6) + d(v_{12}, v_7) + d(v_{12}, v_8) + d(v_{12}, v_9) + d(v_{12}, v_{10}) + d(v_{12}, v_{11})}$$

$$= \frac{1}{2+2+1+2+2+1+1+1+1+1+1}$$

$$= \frac{1}{15} = 0.067$$

2. [3 points] Without using networkx or other graph analysis packages (though you may use them to check your answer), find the eccentricity of vertices 3, 12, and 11.

The eccentricity of a vertex v_i is defined as:

$\max_j d(v_i, v_j)$ **where $d(v_i, v_j)$ are the shortest path distances between vertices v_i and v_j.**

The maximum shortest path distance from vertex 3 to any other vertex is 2. Therefore, the eccentricity of vertex 3 is 2.

The maximum shortest path distance from vertex 12 to any other vertex is 2. Therefore, the eccentricity of vertex 12 is also 2.

The maximum shortest path distance from vertex 11 to any other vertex is 3. Therefore, the eccentricity of vertex 11 is 3.

3. [3 points] Without using networkx or other graph analysis packages (though you may use them to check your answer), find the clustering coefficient of vertex 3.

The subgraph induced by the neighbors of vertex 3 includes vertices 1,2,4, and 5, and has 2 edges. The number of possible edges in a simple graph with 4 vertices is 6. Therefore, the clustering coefficient of vertex 3 is 2/6, or 1/3.

4. [3 points] Without using networkx or other graph analysis packages (though you may use them to check your answer), find the clustering coefficient of the graph.

For all the nodes with degree 1, the clustering coefficient is 0 by definition. This includes vertices 8,9, and 10.

We've already determined that the clustering coefficient of vertex 3 is 1/3.

The clustering coefficient of vertices 1, 2, 4, 6, and 7 is the same, because each of these vertices has degree 2, and the two neighbors have an edge between them. Therefore the clustering coefficient for vertices 1,2,4, 6, and 7 is 1.

The subgraph induced by the neighbors of vertex 12 includes vertices 3 and 6-11. However, these vertices share only one edge, out of a possible 21 edges, and so vertex 12 has a clustering coefficient of 1/12.
The subgraph induced by the neighbors of vertex 5 includes vertices 3, 4, and 11. These vertices share one edge, out of a possible 3 edges, and so vertex 5 has a clustering coefficient of 1/3.

The subgraph induced by the neighbors of vertex 11 includes vertices 12 and 5. However, these vertices are not connected by an edge, so vertex 11 has a clustering coefficient of 0.

The average clustering coefficient is therefore (2/3 + 5+1/12)/21 = 5.75/21 = 0.273809

5. [3 points] Find the betweenness centrality of vertices 3 and 12. You may use networkx or other graph analysis packages, but include the code used to generate your answer in your submission.

Here is an edgelist for the graph that can be read by networkx, which I stored in a file called "hw3graph.txt":

```
1,2
1,3
2,3
3,4
3,5
3,12
4,5
5,11
6,7
6,12
7,12
```

8,12
9,12
10,12
11,12

Then, we can use the following code to read in the edgelist and calculate the (normalized) betweenness centrality for each node:

```
G_hw3 = nx.read_edgelist('hw3graph.txt', nodetype=int, delimiter=',')
nx.betweenness_centrality(G_hw3)
```

The output shows that the normalized betweenness centrality of node 3 is 0.491 and of node 12 is 0.736.

The following code can be used to find the unnormalized betweenness centrality:

```
nx.betweenness_centrality(G_hw3, normalized=False)
```

The unnormalized betweenness centrality of node 3 is 27 and of node 12 is 40.5.

6. [3 points] Using networkx, find the prestige/eigenvector centrality of vertices 3 and 12. Include the code used to generate the answer.

We can use the following code to read in the edgelist as defined in Problem 1, then to find the eigenvector centralities of all the vertices:

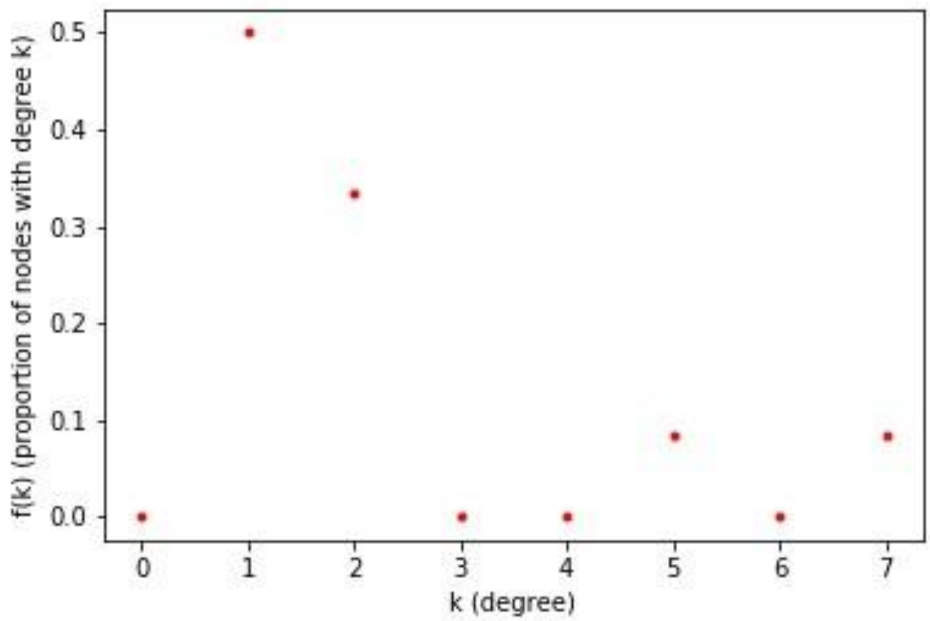
```
G_hw3 = nx.read_edgelist('hw3graph.txt', nodetype=int, delimiter=`,')
nx.eigenvector_centrality(G_hw3)
```

The eigenvector centrality of vertex 3 is 0.465 an the eigenvector centrality of vertex 12 is 0.531.

7. [3 points] Use Python to create a plot of the degree distribution of this graph. Include the code and the plot.

Using the edgelist text file we created in the first problem, we can load the graph, plot its degree distribution, and save the image using the following code:

```
G_hw3 = nx.read_edgelist('hw3graph.txt', nodetype=int, delimiter=‘,')
degree_distribution = nx.degree_histogram(G_hw3) f_k =
np.asarray(degree_distribution)/sum(degree_distribution)
plt.plot(np.arange(len(degree_distribution)), f_k, 'r.') plt.xlabel('k (degree)')
plt.ylabel('f(k) (proportion of nodes with degree k)') plt.savefig('hw3graph_degree_distribution.jpeg')
```


f(k) (proportion of nodes with degree k)
0.0
0.1
0.2
0.3
0.4
0.5
0
1
2
3
4
5
6
7
k (degree)