Social Networks

Week 2 Questions

1.
For graph G, what will the following code snippet return?

```
values = nx.degree(G).values()
x =0
for value in values:
    if(x<value):
        x = value
return x
```

    A. Returns the number of nodes with the minimum degree.
    B. Returns the number of nodes with the maximum degree.
    C. Returns the minimum degree of the graph.
    D. Returns the maximum degree of the graph.
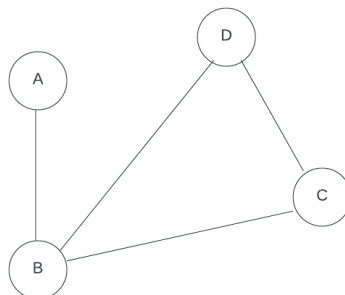
Reference - Lecture-21
Timestamp - 18:42

Answer - (D)

Solution -
values = list of degrees of each node in the graph.
x = maximum degree of the graph.

2.

The density of the given graph above is?

    A. 4/3

B. 2/3
C. 1/3
D. 3/4

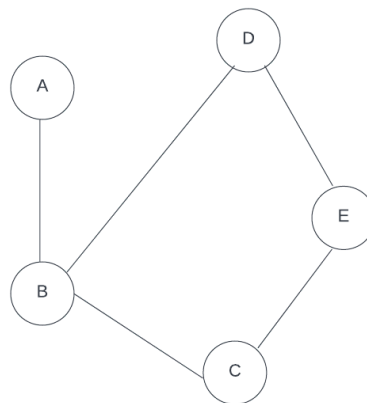Answer - (B)

Solution -
$$Density\ of\ a\ graph\ = \frac{2|E|}{|V|(|V|-1)}$$
Where,
E = Number of edges
V = Number of nodes

3.
For the given graph, If $A = \frac{Highest\ degree}{\Sigma degree}$, what will be the value of A?

A. $\frac{3}{5}$

B. $\frac{2}{5}$

C. $\frac{2}{15}$

D. $\frac{3}{10}$

Answer - (D)

Solution -

Highest degree = 3

$\Sigma degree = 2|E|$, where $E = number\ of\ edges$

4.

Which of the following is an example of a Directed graph?

    I.   Network of Instagram followers
   II.   Ancestral Tree
  III.   Email network
  IV.   Road network

    A. Only IV
    B. Only I, II
    C. Only II, III
    D. Only I, II, III

Reference - Lecture-19

Timestamp -2:20

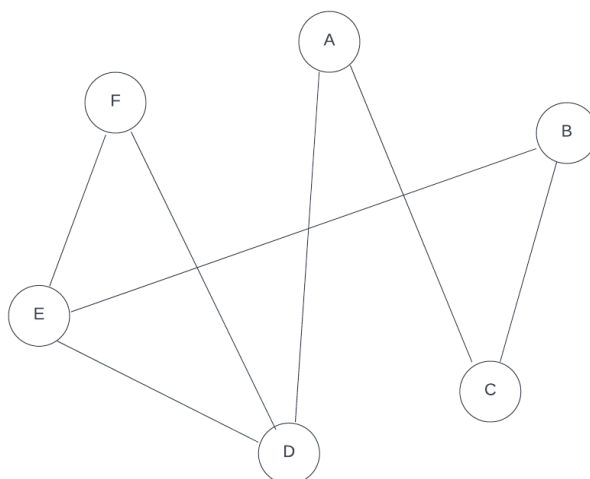Answer - (D)

Solution-

The Road network is an undirected graph whereas the other graphs are directed.

5.

If $X = \frac{clustering\ coefficient\ of\ node\ E}{clustering\ coefficient\ of\ node\ F}$ in the given graph, the value of X is ___.

    A. 1
    B. $\frac{1}{3}$

C. $\frac{1}{9}$

D. $\frac{2}{3}$

Reference - Lecture-21
Timestamp - 31:00

Answer - (B)

Solution -

$Clustering\ Coefficient\ of\ a\ node\ = \frac{Number\ of\ edges\ present\ among\ neighbours\ of\ the\ node}{Total\ number\ of\ edges\ possible\ among\ the\ neighbours}$

So,

Clustering coefficient of node E = $\frac{1}{3}$

Clustering coefficient of node F = $\frac{1}{1}$

Hence, $X = \frac{1}{3}$

6.
Which of the following is/are network dataset format?

 I.   GraphML
 II.   Pajek NET
 III. Comma Separated Value(Edge List format)

   A. Only II
   B. Only III
   C. Only I, III
   D. Only I, II, III

Reference - Lecture-19
Timestamp - 3:40

Answer - (D)

7.
In graph G, where nodes represent words in a dictionary and there is an edge between two nodes if the two words are synonymous. Then, choose the correct option according to the given two statements.

Statement I: The graph G is connected.
Statement II: If the word A is connected to B & B is connected to C, then A is synonymous to C.

A. Both statements are incorrect.
B. Statement I is incorrect & Statement II is correct.
C. Statement I is correct & Statement II is incorrect.
D. Both statements are correct.

Reference - lecture-16
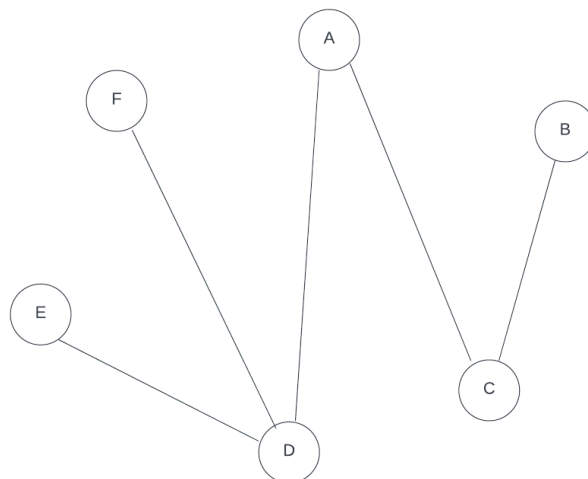Timestamp - 1:40

Answer - (C)

Solution-
Every word gets connected to some word which in turn leads to a connected graph.
If A is a synonym of B and B is a synonym of C, then A might not be a synonym of C.

8.
The diameter of the given graph G is ___.

A. 5
B. 4
C. 3
D. 2

Reference - Lecture-21
Timestamp - 34:30

Answer - (B)

Solution - The maximum length between any two nodes in a graph is known as the Diameter.
From node B to node E distance is 4.

9.

If there exist n nodes with no edges initially then, what is the probability of node V being isolated after including $nlog(n)$ edges uniformly at random?

    A. $\frac{1}{e}$

    B. $\frac{1}{nlog(n)}$

    C. $\frac{1}{n}$

    D. $\frac{1}{n^2}$

Reference - Lecture-24
Timestamp - 18:00

Answer - (D)

Solution -
Probability of node V not including after $nlog(n)$ edges =

$$\left(\left(1 - \frac{1}{\frac{n}{2}}\right)^{\frac{n}{2}}\right)^{2log(n)} = \left(\frac{1}{e}\right)^{2log(n)} = \left(\frac{1}{e^{log(n)}}\right)^2 = \left(\frac{1}{n}\right)^2$$

10.

Choose the data set format which starts with the keyword "graph"?

    A. GML
    B. Graph Exchange XML
    C. Pajek Net format
    D. GEXF

Reference - Lecture-19
Timestamp - 7:50

Answer - (A)

Solution -
The structure of a GML file is -
graph
[
  node
  [
   id A

```
      ]
      node
    [
      id B
    ]
      edge
    [
      source B
      target A
    ]
  ]
```