SL.NO:CAA 216972

SRI SIVASUBRAMANIYA NADAR COLLEGE OF ENGINEERING

(An Autonomous Institution, Affiliated to Anna University, Chennai)
Rajiv Gandhi Salai (OMR), Kalavakkam – 603 110

THEORY EXAMINATIONS

Register Number	205001085		
Name of the Student	Sabarivasan.V		
Degree and Branch	BE CSE	Semester	VIII
Subject Code and Name	UCS1722 Social Network Analysis		
Assessment Test No.	III	Date	

Details of Marks Obtained									
Part A		Part B				Part C			
Question No.	Marks	Question No.	(a) Marks	(b) Marks	Total Marks	Question No.	(a) Marks	(b) Marks	Total Marks
1	2	7				10	8		
2	1								
3	0.5	8				11			
4	2					12			
5	2	9				13	9		
6	2								
Total (A)		Total (B)				Total (C)			
Grand Total (A+B+C)					Marks (in words)				
Signature of Faculty									

SSN

PART-A

SSN

(1) Triad analysis:

It is defined as network is analysed with a group of triads to know about the patterns and connections. In this analysis, 16 varieties of triads are been verified for analysis.

(2) Forbidden triads:

Forbidden triads are involved when unstable data is present on graphs or in a network. Graphs which have forbidden triads denotes that graph involves lot of useless or unstable data.

(3) Visualization:

Visualization is useful in social network analysis where nodes and edges in graphs can be visibly seen. Hence easy analysis can take place and network study can be done on them which helps in easy understanding.

(4) Matrix-based visualization is best suited for co-author networks or collaborative networks where node density is feable and easily can be analysed. It is also applied on larger networks.

(5) Betweenness centrality

It is used to check nodes what act as bridges among clusters or network.

Degree centrality

It is used to check nodes which have lot of connections.

Closeness centrality

It is used to check nodes which are closer to a group of nodes.

(6) ~~Advantages~~

SSN 4

(6) Advantages of Node-Link Diagrams

SSN 5

=> Easily understandable for good visualisation.

=> Scaling on those networks for diagn analysis.

=> Find out clusters, patterns and details about network structures can be studied.

PART - E

SSN

(10) Visualisation in readable form when scaling to large networks

⇒ A subset of network can be taken from a larger network based on the necessity parameters.

⇒ The subset chosen can be well analysed based on deep search among those nodes.

Visual space

⇒ Visual space can be added more when network size gets increased.

⇒ But adding visual space can be done to a certain limit.

Apply filtration

⇒ Filtration of nodes can be applied from a larger network.

⇒ One of the approaches that can be done is filtering a sample network from the larger network based on different studies applied.

⇒ Nodes which don't show a common interest or common pattern are removed out from the larger network.

⇒ But filtration of nodes considering various different parameters or a large network leads to increase in challenging factors.

⇒ Filtration of nodes can be done by reducing the power factor of nodes on that network.

⇒ Another approach that can be handled is hierarchical decomposition of nodes

⇒ After decomposition, the present network structure will be studied from the parents one.

⇒ An increasing scaling to larger networks, another approach which we can follow is cutting out the edges from the network which is larger and removal can be done by applying different centralities.

⇒ Applying different studies and analyses on a large network, will give the idea of constable or useless aggregate data, otherwise nodes which don't share a common pattern or design.

⇒ With the help of those studies and analyses, scaling can be done to larger networks leading to readable form of visualization.

(13)

(a)

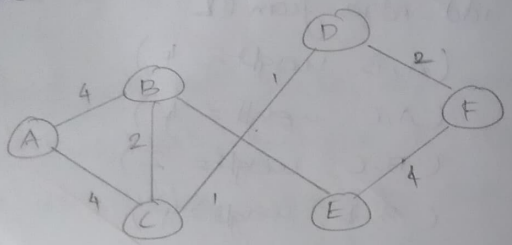
```
import networkx as nx
G = nx.Graph()
G.add_edges_from([
    (A, B, weight = 4)
    (A, C, weight = 4)
    (B, C, weight = 2)
    (B, E, weight = 1)
    (C, D, weight = 1)
    (D, F, weight = 2)
    (E, F, weight = 4)
])
layout = nx.seed(3145721)
position = nx.layout(spring)
G.spring_layout(position)

node_columns = ['green' for - in G.nodes()]
node_edges = ['blue' for - in G.edges()]
```

nx.draw_networkx(G, node_colours, node_edges, with_labels = True)

print(G).

Output:



(b)

```
def dfs(G, start_node):
    visited = set()
    stack = [start_node]
    while (stack):
        current_node = stack.pop()
        if current_node not in visited:
            visited.add(current_node)
            print(current_node)
```



```
        for neighbour in G.neighbours(current_node):
            if neighbour not in visited:
                stack.append(neighbour)
```

dfs(G, 'A') # dfs applied on node 'A'.

Output:

A
B
E
F
C
D

PART-B

SSN

(7) Node-Edge Diagrams

(i) Random-Edge diagrams

⇒ In this type of diagram, nodes are drawn at random space.

⇒ Like nodes, edges are also created randomly on different nodes

⇒ The advantage of this diagram is that no overlead or weightage is considered during visualization.

⇒ Make random-edged diagrams will lead to unfollowed pattern structure.

⇒ Hence, random-edge diagrams are suitable for large networks where patterns are not considered.

(ii) Forced-Edge diagrams

SSN

⇒ In this type of diagram, edges are made among the nodes in the network based on patterns or weightage of nodes..

⇒ These patterns and design structures can be seen, since edges made here in these diagrams are meaningful

⇒ In force-edge diagrams, subset of network can be taken done for deeper analysis.

(iii) Tree-based diagrams

⇒ Tree-based diagram can be drawn for large networks, where each branch can define different patterns.

⇒ Each nodes are considered and formed into clusters based on their closeness to other nodes.

⇒ Those clusters can be connected to other clusters via branches.

⇒ Tree-based structures can be further divided into

(i) Radial tree-diagram

(ii) Hyperbolic tree-diagram.

Radial tree diagram: The node clusters are radially merged with the boundaries of a large network and are pointed towards centre of that network with branches.

Hyperbolic tree diagram

Hyperbolic structure is been created based on the larger network data and patterns among them.

(8)

User Centric Visualisation

Similarities:

They have user defined data or readable form of data which is available in other visualisations too.

Dissimilarities:

⇒ They are allowed to change dynamically based on the user centric data.

⇒ But in other visualisations, the data modification changes are little lower.

Content Centric Visualisation

Similarities:

Content can be stored as nodes and can be connected with other nodes which is seen in other visualisations too.

Dissimilarities

SSN

⇒ Content based data won't get changed easily or frequently.

⇒ Here the network structure is preserved which is not seen in other two visualizations.

Hybrid visualizations:

Similarities:

They hold user based centric data or dynamically changing data in the network visualizations.

Dissimilarities:

⇒ They hold the contracted based nodes which includes leading to large network size.

⇒ Forbidden triads may happen due to larger network size.

(9)

SSN 17

Matrix based diagrams.	Node-Link Based Diagrams.
Matrix is used.	Node-Link diagrams are used.
Can be applied on less denser graphs.	Can be applied on both denser and non-denser graphs.
Node and edge details are based on numbers.	Easy visualization of nodes and edges.
Weightage or values of nodes are added as value in matrix.	Weightage or value of nodes can be visibly seen.
Analyses done using matrices are mathematically easy.	Analyses done using diagrams are visually seen.
Clusters cannot be analysed.	Clusters are easily seen and studied.

Seach algorithms on graphs use matrices efficiently.	search algorithms on graphs applied on diagrams visually.
Matrices can be used for training algorithms for pattern analysis.	Patterns and designs are easily studied or analysed.
Matrices can be used upto a certain limit for graphical representation.	Diagrams can be drawn on larger space wherever network size increases.