Graph Theory Exercises (Pre-Reading for Data Science MSc)

1. Define a graph $G = (V, E)$. Give two examples relevant to data science.

G = Graph
V = Vertices
E = Edges

- Transport network
- Social network

2. Distinguish between weighted and unweighted graphs with examples.

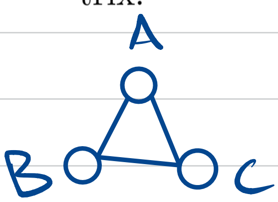
Unweighted
- Edges either present or absent
- All edges are equidistant

Weighted
- Each edge has a number (weight)

3. Explain the difference between simple graphs, multigraphs, and self-loops.

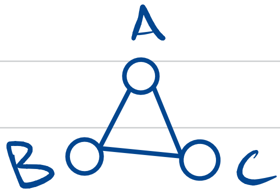
Simple → only 1 edge between 2 vertices
Multi → > 1 edge between 2 vertices
Self-loops → An edge from a vertex back to the original vertex

4. Represent a triangle graph (3 nodes fully connected) as an adjacency matrix.



$$\begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

5. Write the adjacency list for the graph with edges $\{(A, B), (B, C), (C, A)\}$.

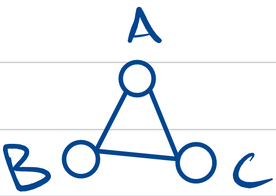


A : {B, C}
B : {A, C}
C : {A, B}

6. Construct the incidence matrix for the graph with edges $e_1 = (A, B), e_2 = (B, C)$.

$$\begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{bmatrix}$$

7. Define the degree of a node. Compute the degrees in the graph with edges $\{(A, B), (A, C), (B, C)\}$.



Degree = No. of edges
A = 2, B = 2, C = 2

8. In a directed graph with edges $A \to B$, $B \to C$, $C \to A$, compute in-degree and out-degree for each node.

In:
A = 1
B = 1
C = 1

Out:
A = 1
B = 1
C = 1

9. What is a degree distribution? Give an example of a network and describe its degree distribution.

Deg. Dist.: degrees across whole graph - count of how many nodes have each frequency of degrees

10. Define a complete graph K_n. How many edges does K_5 have?

$K_n = \frac{n(n-1)}{2}$

$n(n-1)$ → each vertex connects to $n-1$ vertices

$\frac{1}{2}$ → each edge connects 2 vertices

$K_5 = \frac{5(5-1)}{2} = 10$

11. Define a bipartite graph. Give a data science example.

Biparte Graph = Vertices split into two disjoint sets

E.g.: recommender system user-item graph

12. What is a tree? State the number of edges in a tree with n vertices.

Tree → connected graph with no cycles

$n - 1$ edges

13. In the graph $A - B - C - D$, list a path of length 2 and a cycle (if any).

A – B – C (length 2) → no cycle

14. Explain BFS. Why does it guarantee the shortest path in an unweighted graph?

→ Breadth-First Search

BFS explores by level → finds shortest unweighted path by visiting closest neighbours first.

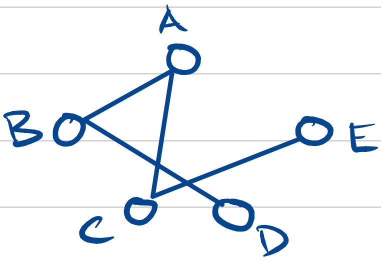
15. Explain DFS. Give one application in graph analysis.

DFS → Depth-First Search

DFS explores deep before backtracking

Applications = Cycle detection, connected components

16. Run BFS on the graph with edges $\{(A,B),(A,C),(B,D),(C,E)\}$ starting at A. List order of visit.



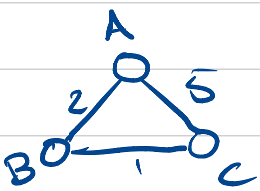
A → B → C → D → E

17. Describe Dijkstra's algorithm in steps. Why must edge weights be non-negative?

1. Initialise dist.
2. Pick next vertex → shortest distance
3. Relax neighbours
4. Repeat

Requires non-negative weights to guarantee correctness

18. Apply Dijkstra's algorithm to the graph: edges $A-B=2, A-C=5, B-C=1$. Find shortest path distances from A.



A to B: A → B (2)

A to C: A → B → C (3)

19. Define graph connectivity. Give an example of a connected and disconnected graph.

Connected: every vertex has a path to
every other vertex.

20. Define clique, community, and diameter of a graph. Give one real-world application for each.

Clique -> fully conneted subset
Community -> densely connected group
(friend clusters)
Diameter -> max shortest path length.