Networks Notes @toastR

Simple Terms

Network: Graph that represents real-life applications

Adjacent Vertices: Two vertices are said to be adjacent if they are at opposite ends of an edge

Multiple Edges: Two or more edges which connect the same vertices are called multiple edges

Loop: A loop is an edge that joins a vertex to itself

Isolated Vertex: A vertex that is not connected to any other vertex

In-Degree: For a digraph, the edges that come into the vertex

Out-Degree: For a digraph, the edges that come out of the vertex

Vertex and edge sets: Set of vertices: V = {A,B,C,D,E}

Set of edges: E = {AB, AD, AE, CD, CE, DE}

Adjacency List: Vertex A is adjacent to vertices B,D and E

Planar graph: a graph that can be drawn in such a way that no edges cross each other.

$$v + e - f = 2$$

for any connected planar graph

this rule also satisfies every platonic solid

Simple Graph: no loops or multiple edges/parallel edges and the number of vertices with an odd degree is even

Null Graph: no edges, only isolated vertices

Trivial Graph: A null graph with one vertex

Connected graph: Each pair of vertices are connected

Disconnected graph: There is at least one pair of vertices that are not connected

Complete Graph: A simple undirected graph in which every vertex is connected to every other vertex by an edge

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

Bipartite Graph: Vertices can be split into two groups/each edge of the graph joins a vertex in the first group to the second group

Complete Bipartite Graph: Every vertex in the first group is connected to a vertex in the second group

Planar graph: Undirected, connected graph, that can be drawn without any edges crossing

Subgraph: A graph that contains no vertices or edges that are not in the original graph

Weighted Graph: A graph in which each edge is labelled with a number used to represent some quantity associated with the edge

Tree: A connected graph that has no cycles or multiple edges.

Other Graph Terms

Network: A graph that satisfies real-life applications

Walk: An alternating sequence of vertices and connecting edges

Open Walk: Starts and finishes at different vertices

Closed Walk: Starts and finishes at the same vertex

Length of the walk: the number of edges of the walk

Trail: Vertices can repeat, edges cannot

Circuit: Vertices can repeat, edges cannot (start and end at the same vertex)

Path: Vertices and edges cannot repeat

Cycle: Vertices and edges cannot repeat (start and end at the same vertex)

Walk: Vertices and edges can repeat

Bridge: an edge if removed, will leave the graph disconnected

Eulerian and Hamiltonian

Eulerian Trail: A trail (vertices can repeat, edges cannot) that travels every edge only once

Eulerian Circuit: A closed Eulerian trail - trail (vertices can repeat, edges cannot) that travels every edge only once and starts and ends at the same vertex

For a connected graph to be Eulerian and for an Eulerian Circuit to exist:

- every vertex must be of an even degree
- you must start and finish at the same vertex
- **every edge must be traversed only once**

Semi-Eulerian Trail: An open eulerian trail - (vertices can repeat, edges cannot) that travels every edge only once and starts and ends at different vertices

Semi-Eulerian Graph: A graph that has a semi-Eulerian trail - if you can start at a vertex, traverse through every edge once and end up at a different vertex

For a connected graph to be semi-Eulerian:

- only two vertices can be of odd degree
- you must start at a vertex with an odd degree and end at the other vertex with an odd degree
- **every edge must be traversed only once**

For a network to be traversable it must be connected and have either zero odd vertices or exactly two odd vertices

If a connected network has zero odd vertices a traversable route can start at any vertex and will finish at that vertex

If a connected network has exactly two odd vertices a traversable route must start at one of these and will then finish at the other

Traversable: If it can be drawn without taking the pen off the paper and without going over the same edge twice (no odd vertices)

Hamiltonian Path: A path (vertices and edges cannot repeat) that includes every vertex once and starts and finishes at different vertices

Hamiltonian Cycle: A closed Hamiltonian Path - includes every vertex in a graph once and it begins and ends at the same vertex

Semi-Hamiltonian graph: A connected graph that contains a Hamiltonian Path but not a Hamiltonian cycle

Properties of Hamiltonian Graphs:

- Hamiltonian graphs are connected
- any Hamiltonian cycle can be converted into a Hamiltonian path by removing one of its edges
- a Hamiltonian path can be converted to a Hamiltonian cycle only **if** the endpoints of the Hamiltonian path are adjacent
- all complete graphs K_n are Hamiltonian
- Complete bipartite graphs which have the same number of vertices in each group are hamiltonian

For a graph to be Hamiltonian or Semi-Hamiltonian it must be a connected graph and it can have no more than two vertices of degree 1

Eulerian trail: A trail that passes along every edge **once**

If an Eulerian trail ends at the starting vertex it is an **Eulerian cycle**

Eulerian graph: A connected traversable graph, starting and finishing at the same vertex with no odd vertices

Semi-Eulerian graph: A connected traversable graph, starting and finishing at different vertices with two odd vertices

Hamiltonian Path: A path that passes through each vertex **once**

If a Hamiltonian Path ends at the starting vertex it is a **Hamiltonian cycle**

Hamiltonian Graph: A connected graph which has a Hamiltonian cycle

Semi-Hamiltonian Graph: A connected graph which has a Hamiltonian path but not a Hamiltonian cycle

Matrix Terms

Adjacency Matrices: Two vertices are said to be adjacent if they are connected by an edge

Undirected Graph

Adjacency Matrix: An adjacency matrix for an undirected graph containing n vertices is a square matrix of order $n * n$ in which the entry in row *i* and column *j* is the number of edges joining the vertices *i* and *j*

Matrix properties of an undirected graph:

- A square matrix of other n*n where n is the number of graph vertices
- The entry in row *i*, column *j* is the **number of edges** joining vertex *i* to *j*
- The entries in the matrix are **symmetrical across the leading diagonal**, any entry in row *i* and column *j* is the same in column *i* and row *j*
- If the graph has no loops, then the leading diagonal shows only zeros
- A loop is counted as one edge
- The sum of the row entries gives the degree of the vertex corresponding to that row
- If the graph has no multiple edges then all entries are either 0 or 1

Directed Graph:

Adjacency Matrix: An adjacency matrix for an undirected graph containing n vertices is a square matrix of order $n * n$ in which the entry in row *i* and column *j* is the number of **directed** edges joining the vertices *i* and *j*

Matrix properties of an undirected graph:

- A square matrix of other n*n where n is the number of graph vertices
- The entry in row *i*, column *j* is the **number of directed edges** joining vertex *i* to *j*

- If the graph has no loops, then the leading diagonal shows only zeros
- the sum of entries in a column is equal to the in-degree of the vertex that corresponds to that column
- the sum of entries in a row is equal to te out-degree of the vertex that corresponds to that row

Powers of the Adjacency Matrix - for directed and undirected graphs

M = the number of walks of length **one** between selected vertices (no stop-over)

M^2 = the number of walks of length **two** between selected vertices (one stop-over)

M^3 = the number of walks of length **three** between selected vertices (two stop-over)

Example: Matrix $N = M + M^2$, gives the total number of walks of length 1 or 2 between selected vertices, or Matrix N gives the number of walks with at most one-stop over