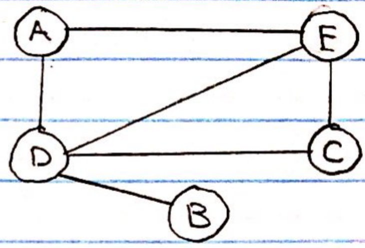
Graphs 7/10/2018

A Graph is a set of Vertices (nodes) and Edges used for depicting Entity and Relationships

Graph Notation: G = (V, E)

V = Set of Vertices: V = {A, B, C, D, E, F}

E = Set of Edges: E = {(A,D), (A,E), (B,D),
(C,D), (C,E), (D,E)}



Path: A a sequence of Connected Vertices: {A, E, C}

Cycle: A path that starts and ends with the same vertex

Sub Graph: a subsect of a graph

Types of Graphs:

Sparse: more Vertices than Edges

Dense: more edges than Vertices

Connected: Every Vertex Connected by a path

Disconnected: A Graph that has a clear break

Weighted: Edges have numerical Values

Undirected: Edges are bidirectional

Directed: Edges are unidirectional

Cyclic: Graph has a cycle

Acyclic: Graph has no cycle

Vertex Terminology:

Degree: number of edges a Vertex has

Indegree: number of incoming edges

Outdegree: number of outgoing edges

Representing Graphs:

Edge list: "pretty bad" - Joonho, a list of Every Edge in a graph, it has O(|E|) lookup time for any specific edge: (A,D), (A,E), (B,D), (C,D), (C,E), (D,E)

Adjacency list: A list of Vertices, where each Vertix also has a list of its neighbors. Very Space Efficiant, getting the neighbors is efficiant, but testing if two Vertices are connected is O(|V|)

A: D, E

B: D

C: D, E

D: A, B, C, E

E: A, C, D

Representing Graphs Continued. 7/10/2018

Adjacency Matrix: the graph is represented as A $V \times V$ matrix where V is the number of vertices, it has a very fast $O(1)$ lookup for vertex connection, but uses a lot of space, and thus should be used primarily with Dense graphs.

\	A	B	C	D	E
A				1	1
B				1	
C				1	1
D	1	1	1		1
E	1		1	1	

row - coming from Column - going to.

the number "1" would be replaced with that edges weight value if it exists.

Graph traversal

Breadth first Search: Use a set of visited vertices, and a queue of vertices to visit.

Depth first Search: Use a set of visited vertices, and a stack of vertices to visit