Definition (math): A graph $G$ is a set of "vertices" $V$ (also called "nodes") and a set of edges $E$.

An edge $e$ is an ordered pair of vertices $(v_1, v_2)$.

Example

Vertices: $a, b, c, d, e, f, g, h$

Examples of Graphs

- vertices
- edges
- airports
- flights
- web pages
- links (URLs)
- Java objects
- references
class SLinkedList< V > {
    SNode< V > head;
    
    }

class Tree< V > {
    TNode< V > root
    
    }

class Graph {
    
    }

- no special "head" or "root" node
- no standard way
- no Java class Graph

Data Structures for graphs

vertices:
- a class V

edges:
- list of vertices V
- boolean [ ][ ]

"Adjacency List"

Example: the vertices are characters

a - c
b - f
c - f
d - a, c
e - b, f
f - b, e
g - h
h -

Suppose each list is required to be sorted.
How to represent the graph?

class Graph<V> {
    LinkedList<GraphNode<V>> gNodes
    // Array List<GraphNode<V>> gNodes
}

Data Structures for graphs?

vertices:
- a class V

edges:
- list of vertices V
- boolean[][]
Example 1

adj[i][j] =
\[
\begin{cases}
1 & \text{if } (v_i, v_j) \in E \\
0 & \text{if } (v_i, v_j) \notin E
\end{cases}
\]

Example 2

adj[i][j] =
\[
\begin{cases}
1 & \text{if } (v_i, v_j) \in E \\
0 & \text{if } (v_i, v_j) \notin E
\end{cases}
\]

Adjacency List vs. Matrix

- space efficiency?
- time efficiency? (get edges of v)
- flexibility? (add or remove v)

Terminology
- in degree of vertex
- out degree of vertex
- path
- cycle

Examples
- \((c, 2), (b, 2), (g, 0)\)
- \((g, 1)\)
- \((a, c, f, e, b, f)\)
- \((e, b, f, c), (c, f, e)\).

Graph Algorithms in {Comp 251, Comp 360}
- shortest paths {see slides below
- travelling salesman
- topological sorting
- network flows
- ....

Weighted graphs

Examples

\[
\begin{align*}
ge & \to e & 5 \\
e & \to d & 2 \\
d & \to c & 3 \\
c & \to f & 7 \\
f & \to e & 6 \\
e & \to b & 1 \\
\end{align*}
\]
What is the shortest path from V to W?

What is the shortest weighted path from V to W?

Travelling Salesman

Find the shortest cycle that visits all vertices once (except for first = last vertex, which is visited twice)

Directed Graph

Undirected Graph

Directed Acyclic Graphs
edges no cycles are directed

e.g. there are three paths from a to d

Math prerequisites for many Comp Tech, Sys courses

SOFTWARE

SYSTEMS (computers, networks, distributed systems, web)
APPLICATIONS (graphics, vision, VR, machine learning, games)
THEORY (crypto, optimization, coding error correction, complexity)

201 Java

206 Software Systems

230 Comp Design

303 Prog Tech

251 Data Structure Alg.

350 Numerical Methods

MATH prerequisites for many Comp Tech, Sys courses

240 Discrete

223 Lin Alg

222 Calc III

340 Discrete 2

323 Probability

310 Operating Systems

421 Databases

424 A.I.

360 Algorithm