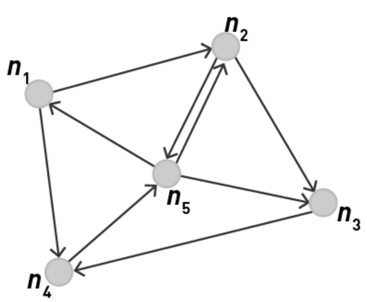
Graphs (cont.)

- Different types of graphs:
  - Directed vs. undirected edges
  - Presence or absence of cycles

Graph Representations

- Graph G = (V,E)
  - V represents the set of vertices (nodes).
  - E represents the set of edges (links).
  - Both vertices and edges may contain additional information.
- E.g.
  - V={1,2,3,4,5}
  - E={(1,2)(1,5)(2,5)(2,4)(4,5)(2,3)(2,4)}
- Standard ways to represent a graph
  - Vertices and edges: G = (V, E): Not food for computational purposes



	n_1	n_2	n_3	n_4	n_5
n_1	0	1	0		0
n_2	0	0	1	0	1
n_3	0	0	0	1	0
n_4	0	0	0	0	1
n_5	1	1	1	0	0

adjacency matrix

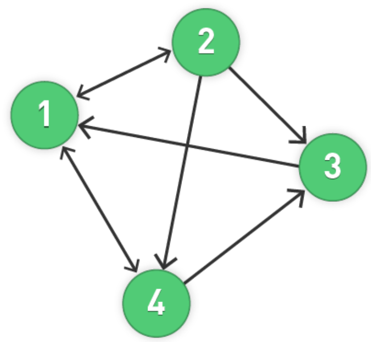
n_1	$[n_2, n_4]$
n_2	$[n_2, n_4]$
n_3	$[n_2, n_4]$
n_4	$[n_2, n_4]$
n_5	$[n_1, n_2, n_3]$

adjacency lists

Adjacency Matrices

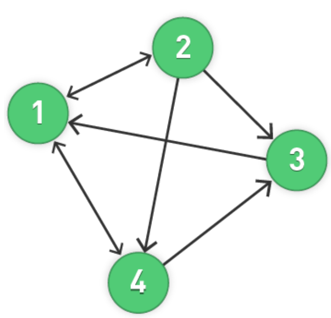
- Represent a graph as an n × n square matrix M.
  - n = |V|
  - Mij = 1 means a link from node i to j
- Advantages:
  - Naturally encapsulates iteration over nodes.
  - Rows and columns correspond to inlinks and outlinks.
- Disadvantages:
  - Lots of zeros for sparse matrices
  - Lots of wasted space
- Standard ways to represent a graph
  - Vertices and edges: G = (V, E): not good for computational purposes

	1	**2**	**3**	**4**
1	0	1	0	1
2	1	0	1	1
3	1	0	0	0
4	1	0	1	0



Adjacency Lists

- An array Adj of |V| lists.
- For each u in V, the adjacency list Adj[u] contains all the vertices v such that (u,v) in E.
- Take adjacency matrices...and throw away all the zeros.



	1	**2**	**3**	**4**
1	0	1	0	1
2	1	0	1	1
3	1	0	0	0
4	1	0	1	0

1: [2, 4]
2: [1, 3, 4]
3: [1]
2: [1, 3]

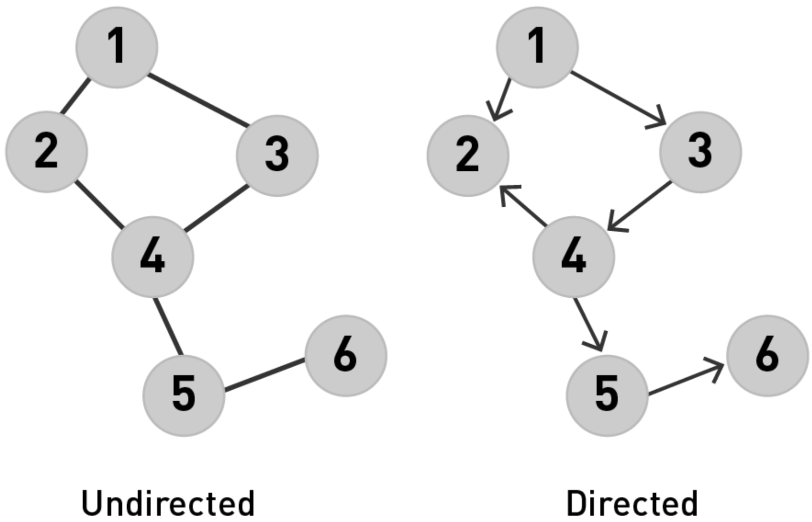
Adjacency List Is Preferred

- Adjacency list is usually preferred, because it provides a compact way to represent **sparse** graphs: those for which |E| is much less than |V|.
- Adjacency matrix may be preferred when the graph is **dense**, or when we need to be able to tell quickly if there is an edge connecting two given vertices.
- Adjacency list is preferred in the following:
    - In a social network of n individuals, there are n(n – 1) possible friendships (where n may be on the order of billions). However, even the most gregarious will have relatively few friends compared to the size of the network (thousands, perhaps, but still far smaller than hundreds of millions).
    - The same is true for the hyperlink structure of the web: Each individual web page links to a minuscule portion of all the pages on the web.

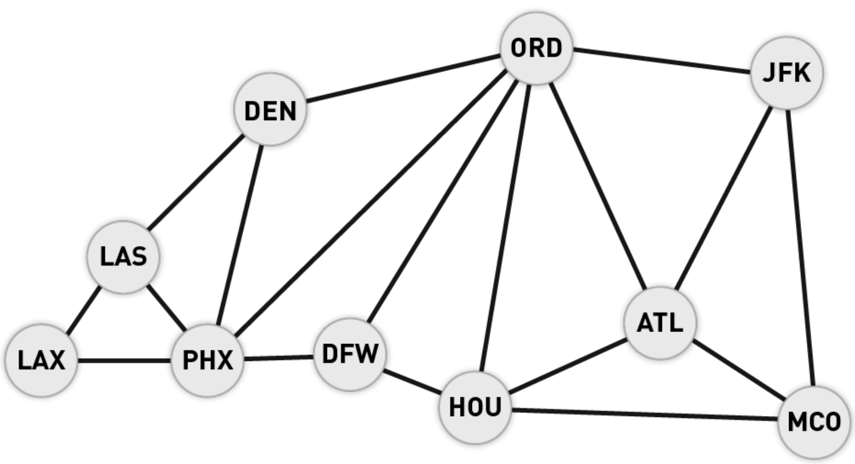
Paths and Graph Traversals

- In graph theory, a path in a graph is a finite or infinite sequence of edges that connect a sequence of vertices, which, by most definitions, are all distinct from one another.
- Directed path:
    - In a directed graph, a directed path (sometimes called dipath) is again a sequence of edges (or arcs) that connect a sequence of vertices but with the added restriction that the edges all be directed in the same direction.

Directed Graph



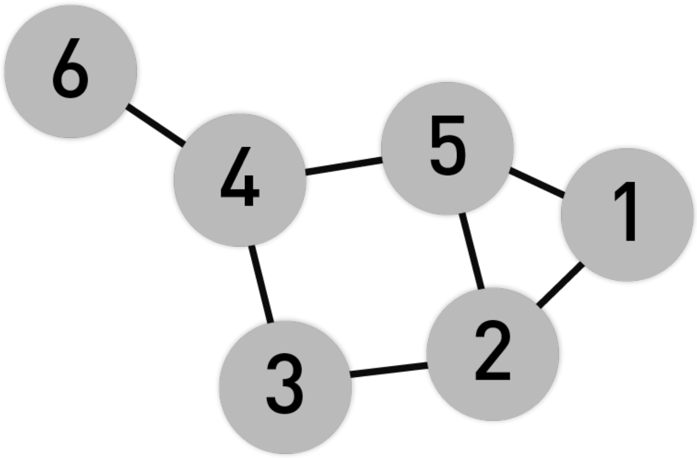
Graph Traversal: A Path as an Alternating Sequence of Vertices and Edges



Source	Destination	Distance	Shortest Paths
JFK	LAX	3	JFK-ORD-PHX-LAX
LAS	MCO	4	LAS-PHX-DFW-HOU-MCO and four others
HOU	JFK	2	HOU-ATL-JFK and two others

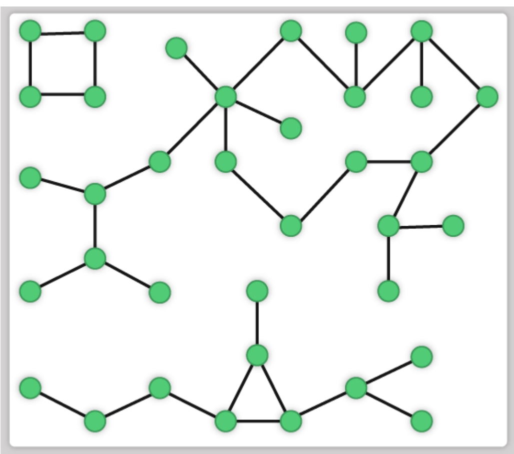
A path/walk of length k in a graph is an alternating sequence of vertices and edges, v_0,e_0,v_1,e_1,v_2,..., v_{k-1},e_{k-1},v_k, which begins and ends with vertices.

Graph Properties



An example graph, with the properties of being **planar** and being **connected**, and with order 6, size 7, **diameter** 3, **girth** 3, **vertex connectivity** 1, and **degree sequence** <3, 3, 3, 2, 2, 1>.

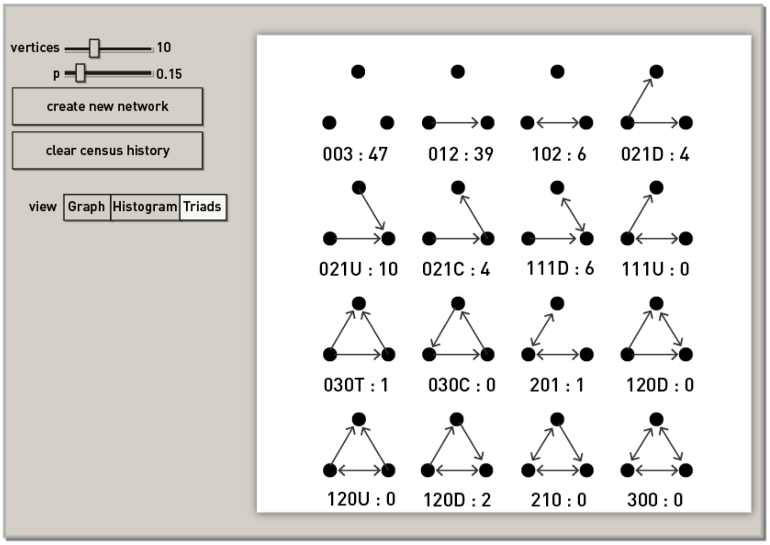
Connected Component



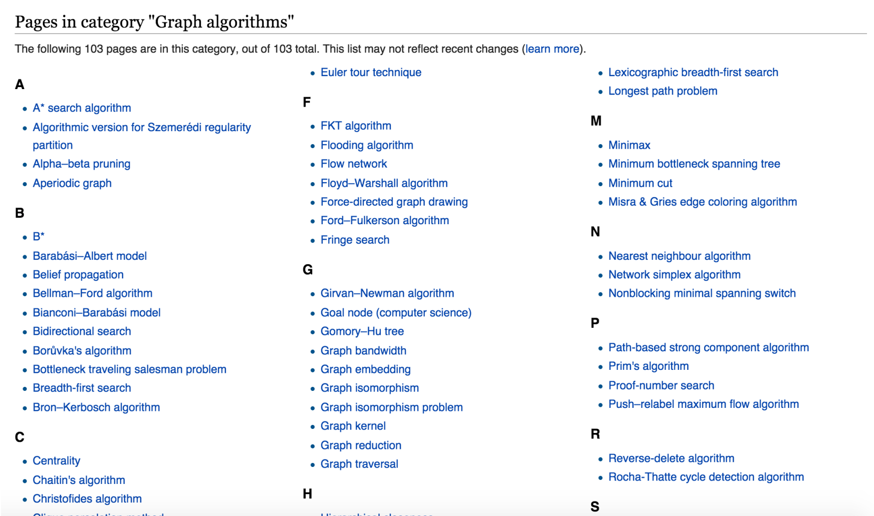
A graph with three connected components.

A connected component (or just component) of an undirected graph is a subgraph in which any two vertices are connected to each other by paths, and which is connected to no additional vertices in the supergraph.

Study Graphs Using Triads



Graph Algorithms



Pages in category "Graph algorithms"

The following 103 pages are in this category, out of 103 total. This list may not reflect recent changes (learn more).

A

- A* search algorithm
- Algorithmic version for Szemerédi regularity partition
- Alpha–beta pruning
- Aperiodic graph

B

- B*
- Barabási–Albert model
- Belief propagation
- Bellman–Ford algorithm
- Bianconi–Barabási model
- Bidirectional search
- Borůvka's algorithm
- Bottleneck traveling salesman problem
- Breadth-first search
- Bron–Kerbosch algorithm

C

- Centrality
- Chaitin's algorithm
- Christofides algorithm
- Clique percolation method

- Euler tour technique

F

- FKT algorithm
- Flooding algorithm
- Flow network
- Floyd–Warshall algorithm
- Force-directed graph drawing
- Ford–Fulkerson algorithm
- Fringe search

G

- Girvan–Newman algorithm
- Goal node (computer science)
- Gomory–Hu tree
- Graph bandwidth
- Graph embedding
- Graph isomorphism
- Graph isomorphism problem
- Graph kernel
- Graph reduction
- Graph traversal

H

- Hierarchical closeness

- Lexicographic breadth-first search
- Longest path problem

M

- Minimax
- Minimum bottleneck spanning tree
- Minimum cut
- Misra & Gries edge coloring algorithm

N

- Nearest neighbour algorithm
- Network simplex algorithm
- Nonblocking minimal spanning switch

P

- Path-based strong component algorithm
- Prim's algorithm
- Proof-number search
- Push–relabel maximum flow algorithm

R

- Reverse-delete algorithm
- Rocha-Thatte cycle detection algorithm

S

http://en.wikipedia.org/wiki/List_of_algorithms#Graph_algorithms