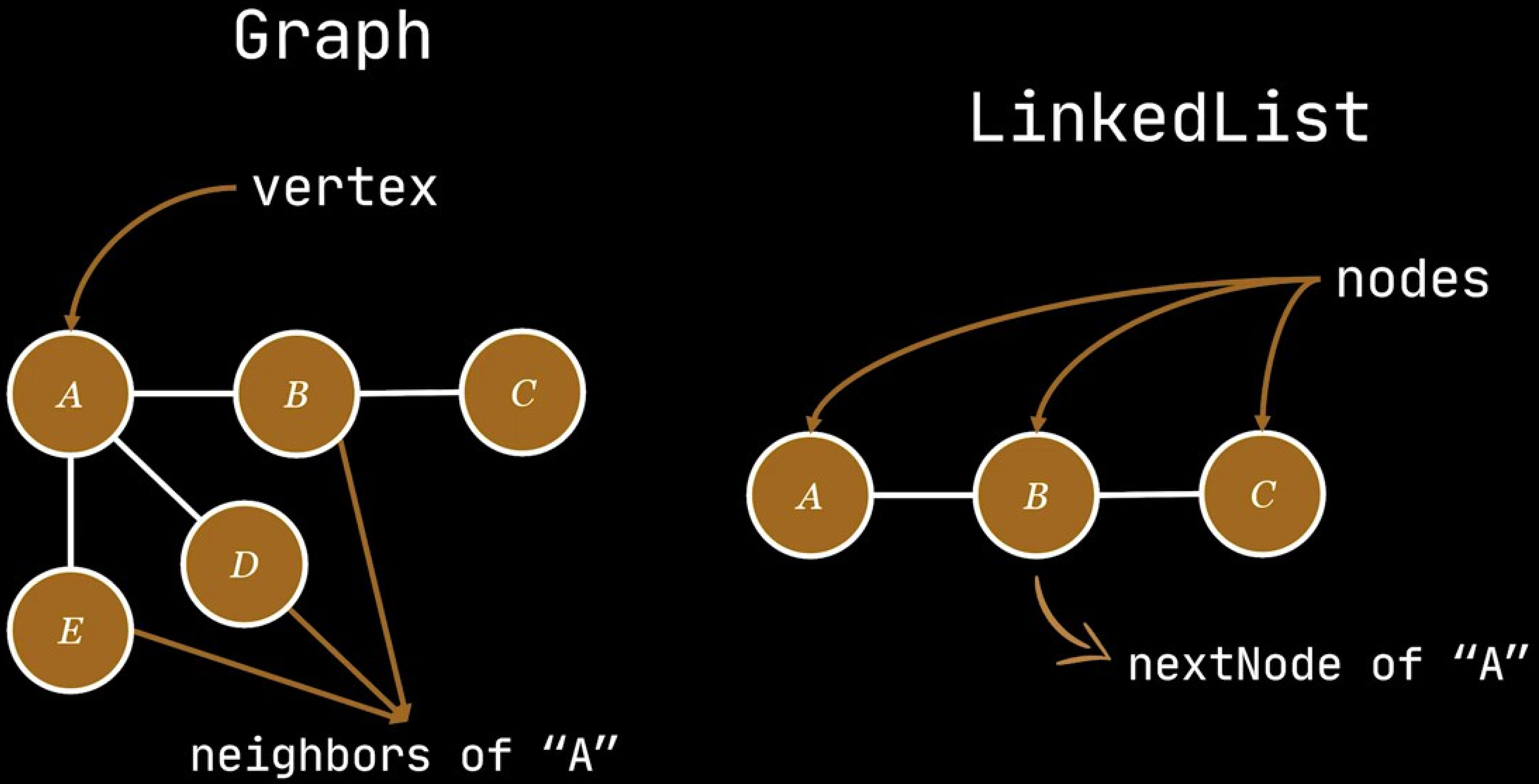
Lecture 08

Graphs


Graph
vertex
A
B
C
D
E
neighbors of "A"
LinkedList
nodes
A
B
C
nextNode of "A"

Graph terminology

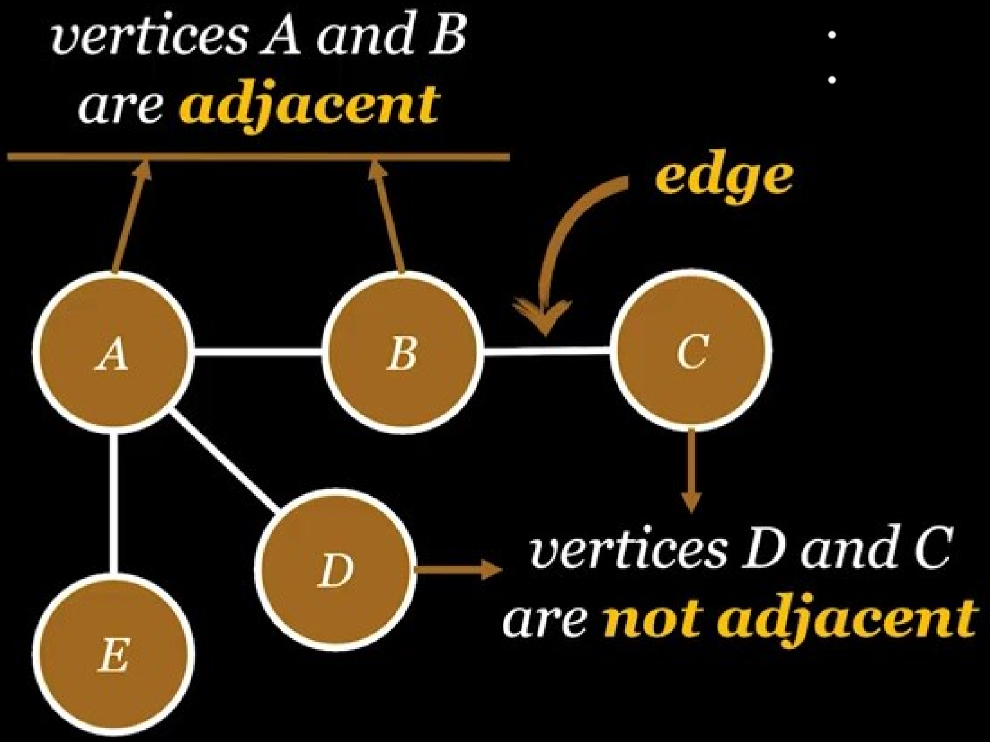
- A graph is a collection of nodes (or vertices, singular is vertex) and edges (or arcs)
- Each node contains an element
- Each edge connects two nodes together (or possibly the same node to itself) and may contain an edge attribute

edge connecting *A* and *B* = *(A, B)*

edge connecting *B* and *A* = *(B, A)*

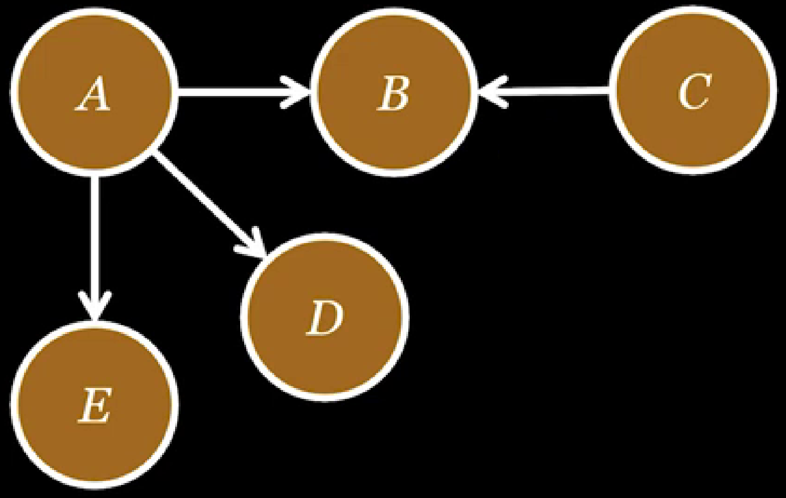
edge connecting *C* and *B* = *(C, B)*

⋮



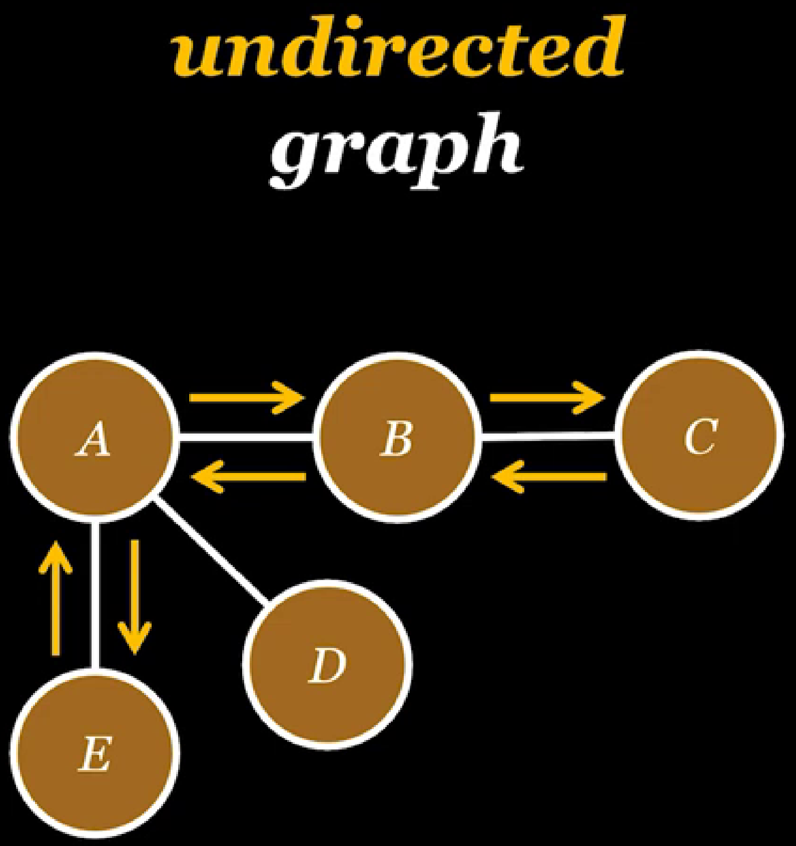
Graph terminology

- A directed graph is one in which the edges have a direction
  - If a directed edge goes from node S to node D, we call S the source and D the destination of the edge
  - The edge is an out-edge of S and an in-edge of D
  - S is a predecessor of D, and D is a successor of S
  - The in-degree of a node is the number of in-edges it has
  - The out-degree of a node is the number of out-edges it has



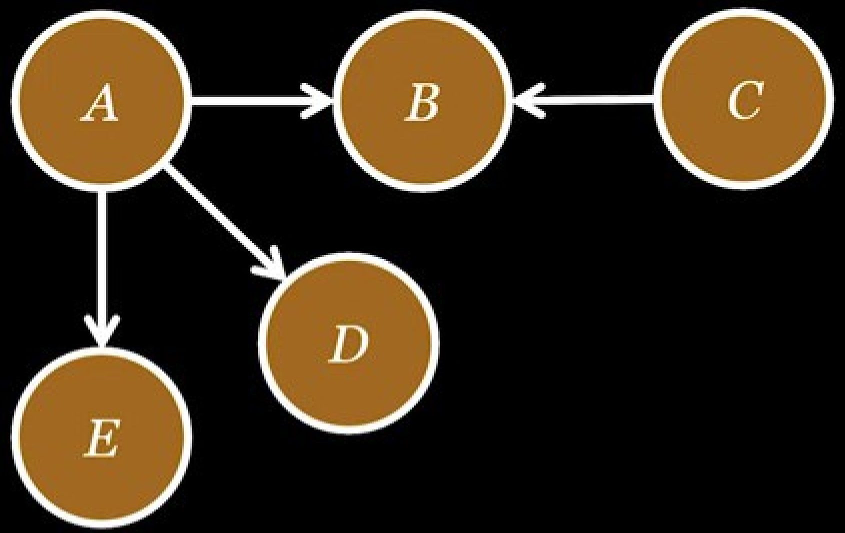
Graph terminology

- An undirected graph is one in which the edges do not have a direction



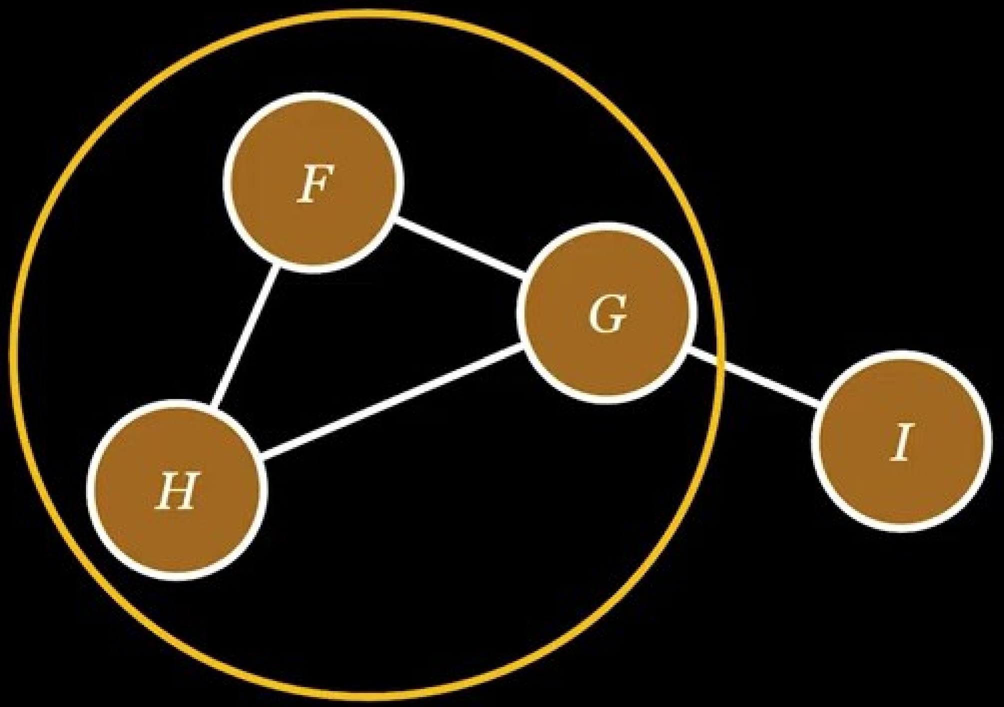
Graph terminology

- The size of a graph is the number of nodes in it
- The empty graph has size zero (no nodes)
- If two nodes are connected by an edge, they are neighbors (and the nodes are adjacent to each other)
- The degree of a node is the number of edges it has



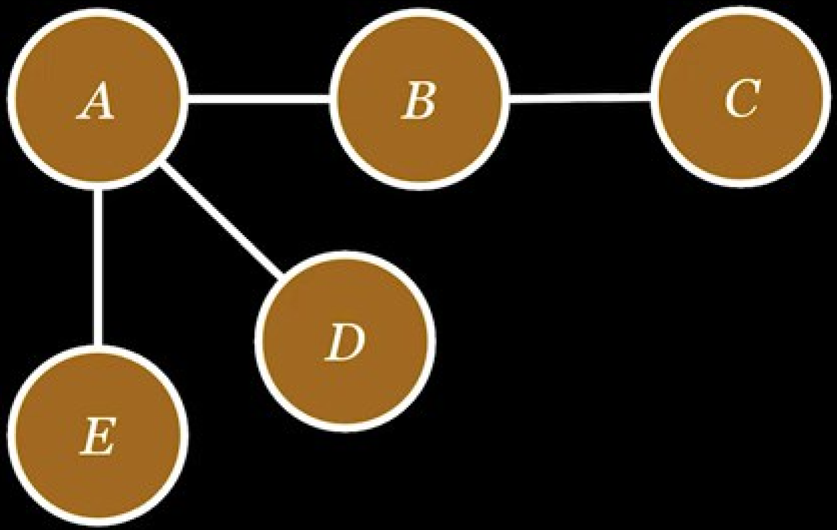
Graph terminology

- A path is a list of edges such that each node (but the last) is the predecessor of the next node in the list
- A cycle is a path whose first and last nodes are the same
  - A cyclic graph contains at least one cycle
  - An acyclic graph does not contain any cycles



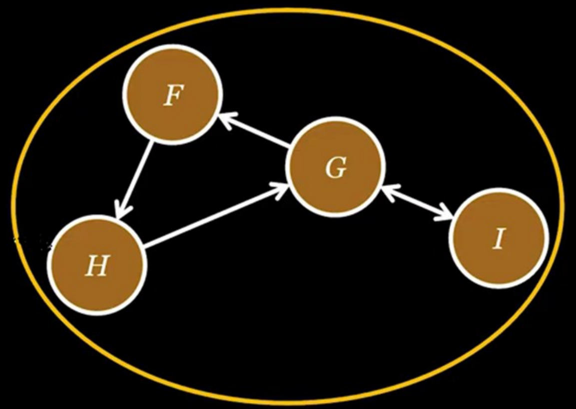
Graph terminology

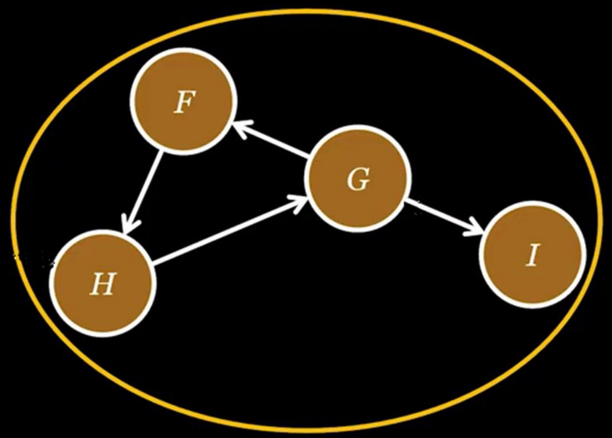
- An undirected graph is connected if there is a path from every node to every other node



Graph terminology

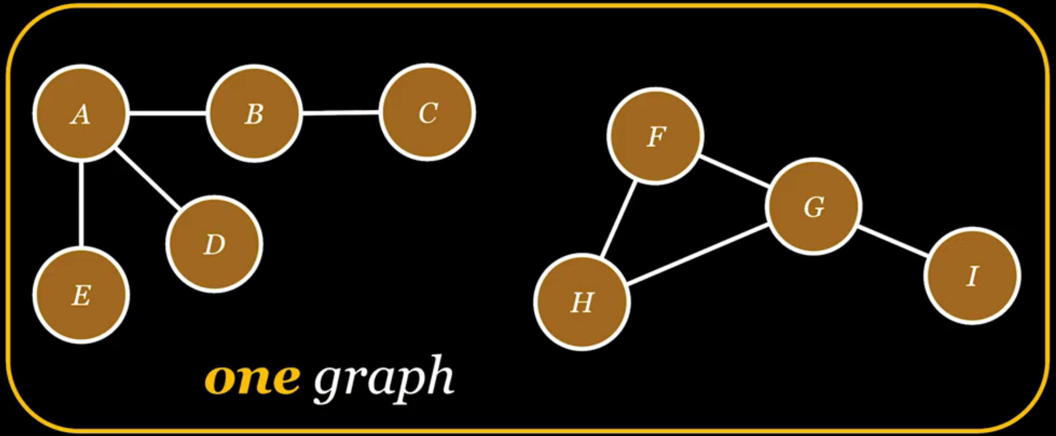
- A directed graph is
  - strongly connected if there is a path from every node to every other node
  - weakly connected if the underlying undirected graph is connected
  - Node X is reachable from node Y if there is a path from Y to X





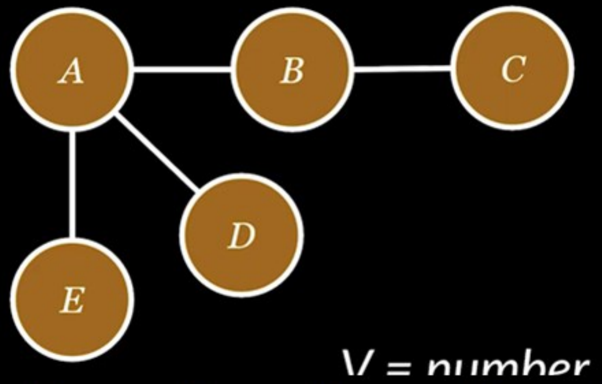
Graph terminology

- A subset of the nodes of the graph is a connected component (or just a component) if there is a path from every node in the subset to every other node in the subset



Adjacency-matrix representation

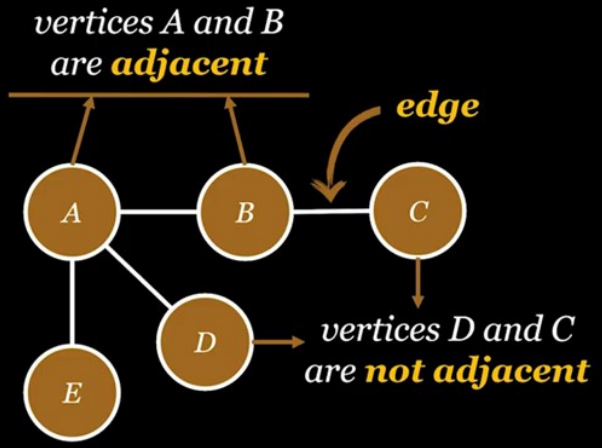
- One simple way of representing a graph is the adjacency matrix
- A 2-D array has a mark at [i][j] if there is an edge from node i to node j
- The adjacency matrix is symmetric about the main diagonal
- This representation is only suitable for small graphs



	A	B	C	D	E
A	false	true	false	true	true
B	true	false	true	false	false
C	false	true	false	false	false
D	true	false	false	false	false
E	true	false	false	false	false

Edge-set representation

- An edge-set representation uses a set of nodes and a set of edges
  - The sets might be represented by, say, linked lists
  - The set links are stored in the nodes and edges themselves
  - This is seldom a good representation



edge *connecting A and B* = (A, B)

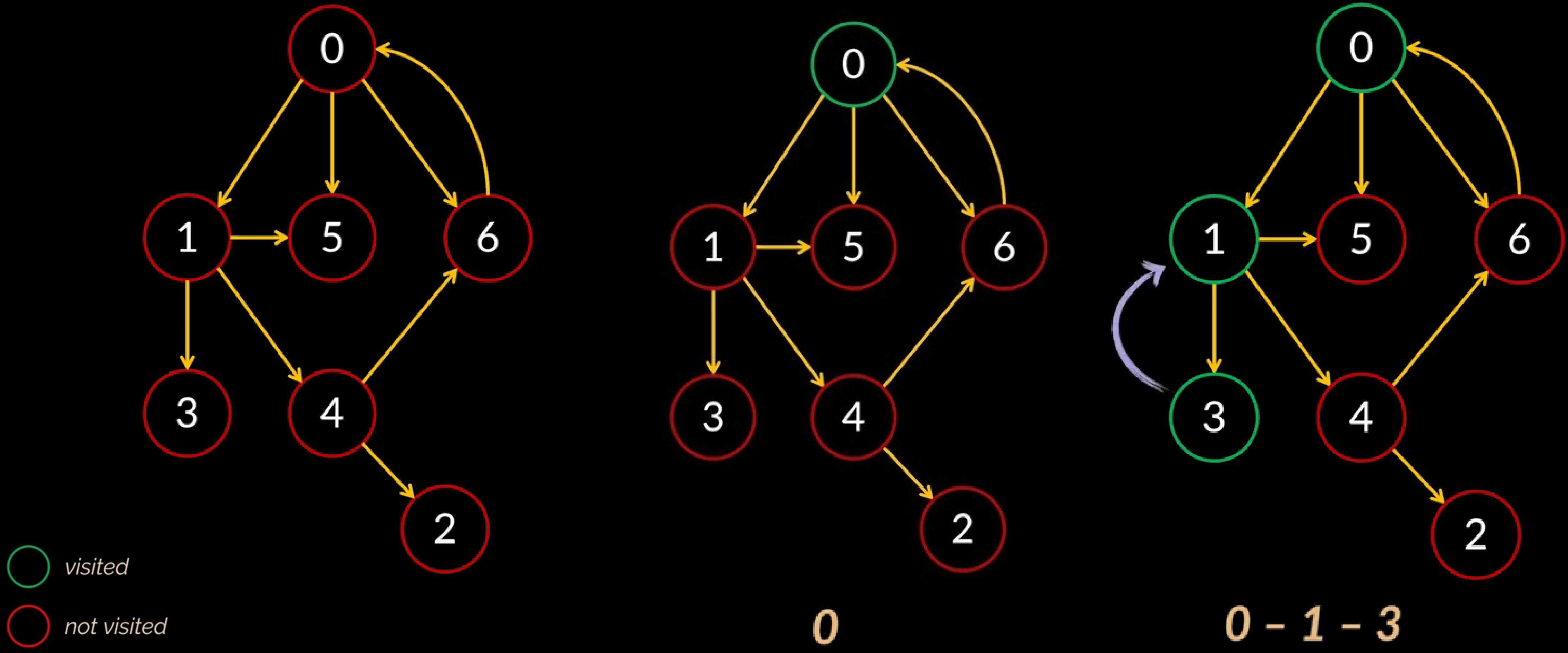
edge *connecting B and A* = (B, A)

edge *connecting C and B* = (C, B)

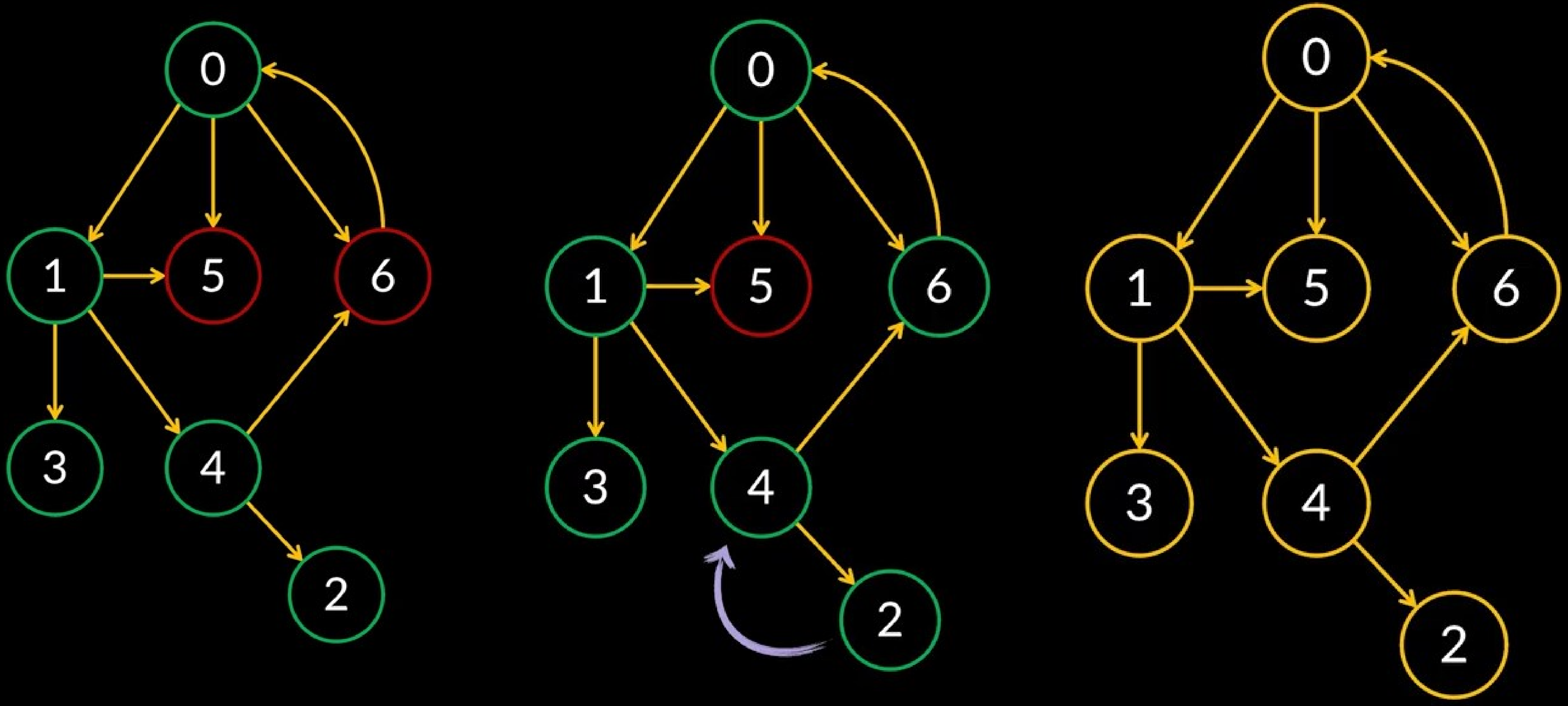
Searching a graph

- With certain modifications, any tree search technique can be applied to a graph
  - This includes depth-first, breadth-first, depth-first iterative deepening, and other types of searches
- The difference is that a graph may have cycles
- We don't want to search around and around in a cycle
- To avoid getting caught in a cycle, we must keep track of which nodes we have already explored

Depth-first search



Depth-first search



visited

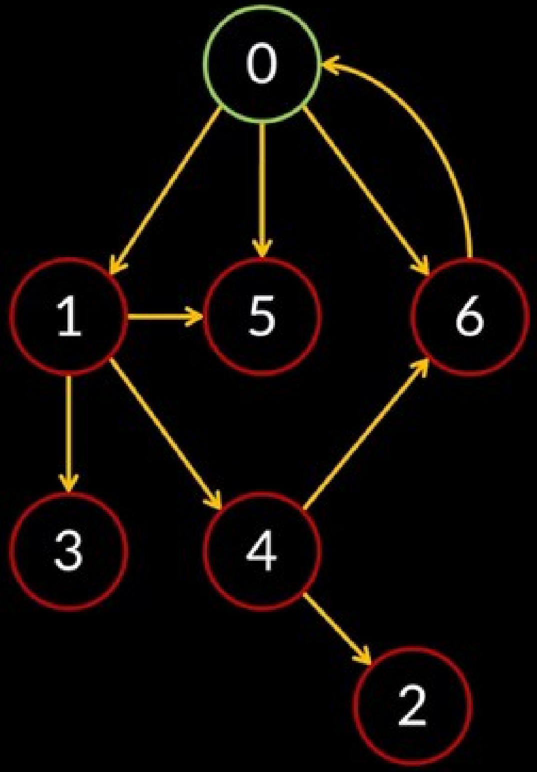
not visited

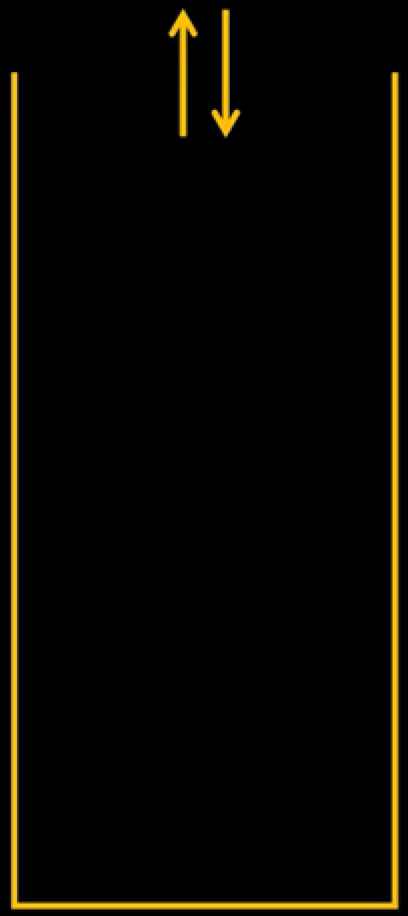
0 – 1 – 3 – 4 – 2

0 – 1 – 3 – 4 – 2 – 6

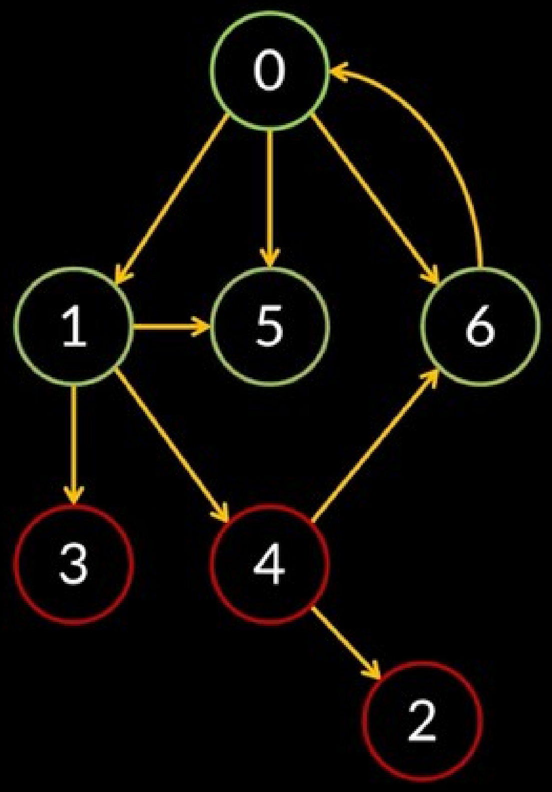
0 – 1 – 3 – 4 – 2 – 6 – 5

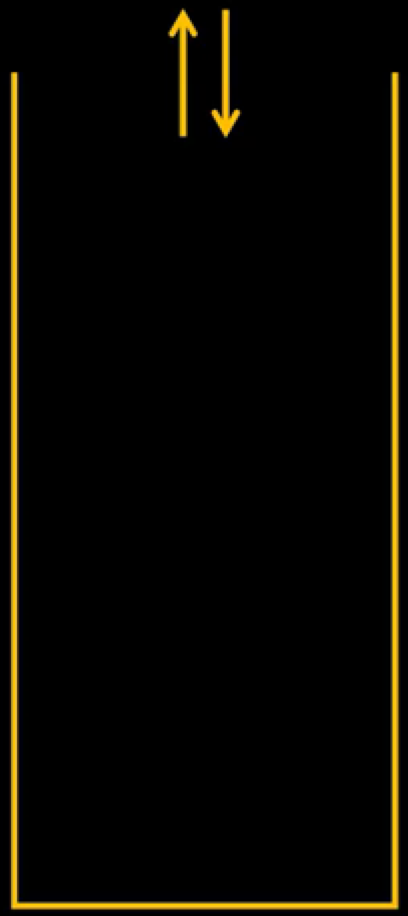
Depth-first search



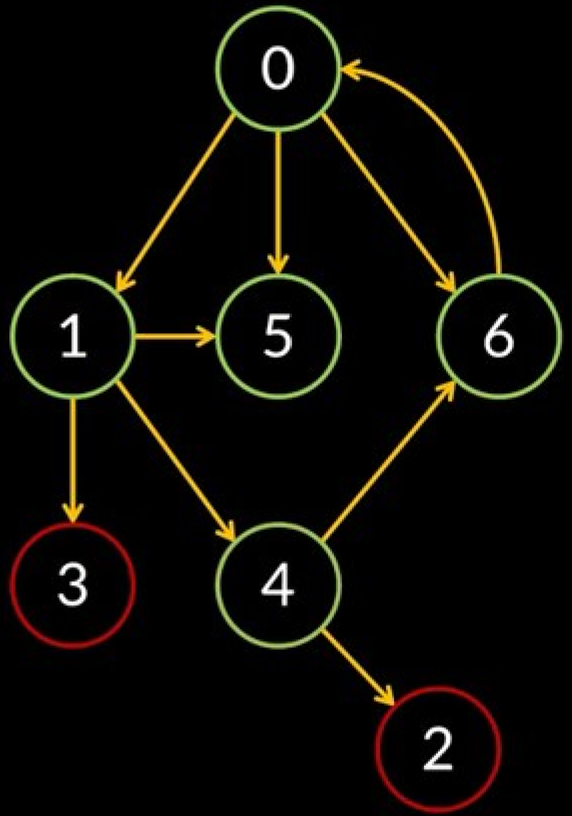


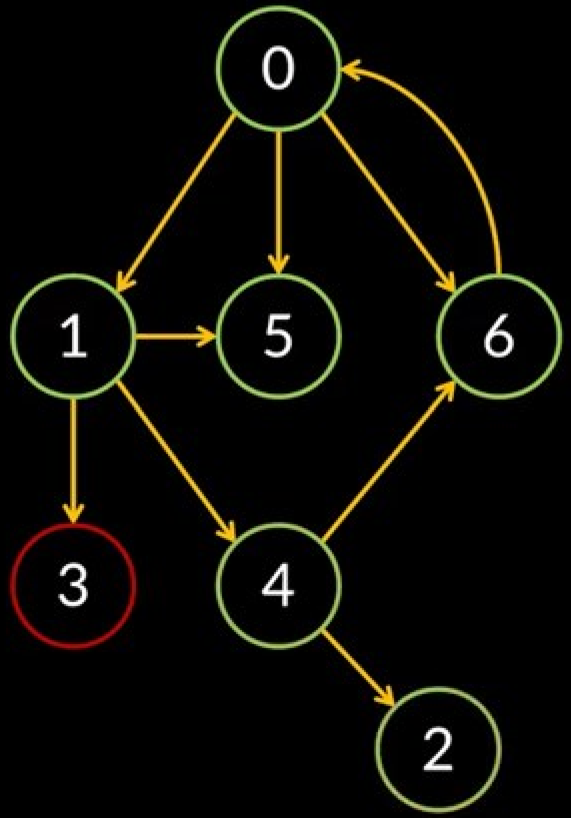
Depth-first search

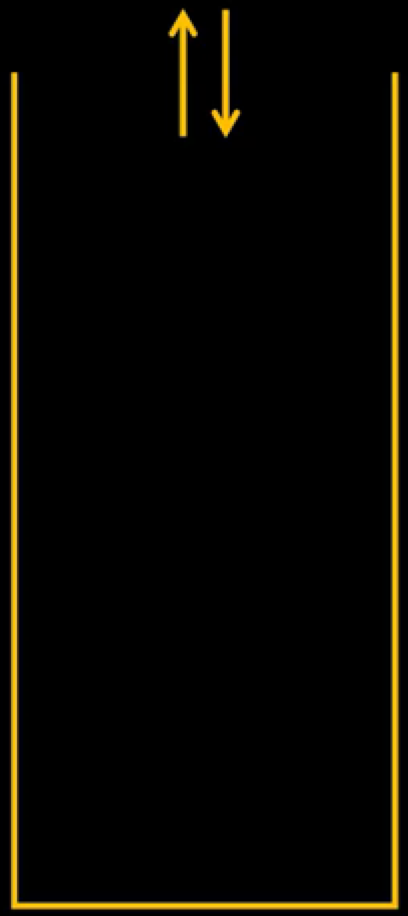




Depth-first search

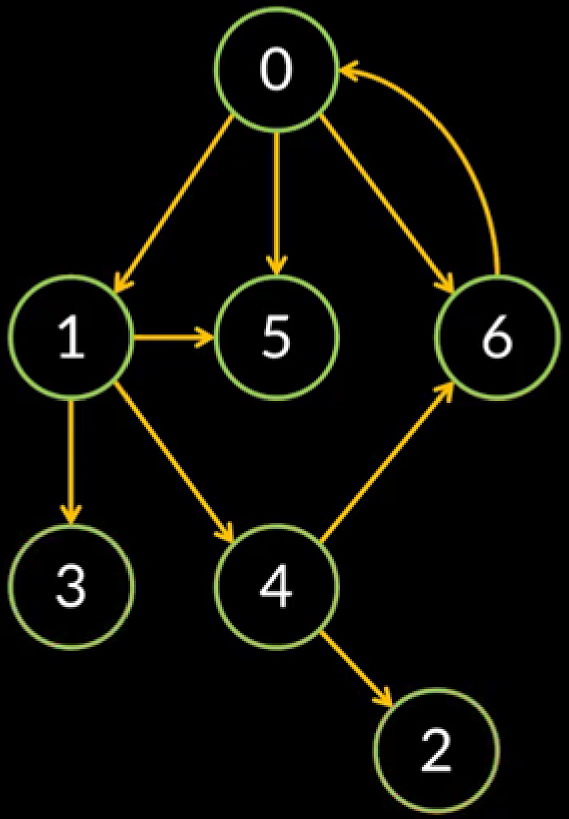






Depth-first search

0
6
5
1
4
2
3



0 – 1 – 3 – 4 – 2 – 6 – 5