Topology 101: graph

A graph is a collection of nodes in which each node can have an arbitrary number of successors and an arbitrary number of predecessors.

Two types of graphs

* Directed can only move in direction of arror
* Undirected can move in either direction

Graph terminology

NODE = vertices

LINK = edges (arcs in a directed graph)

A weighted graph has labels or metrics assigned to the edges

Special types of graphs

* 2-clique
* 3-clique
* 4-clique
* 5-clique
* Planar needles cross
* Nonplanar

A cycle is a path from a vertex back to itself.

Usually, we say that the edges in the path must be distinct.

A graph with a cycle is called cyclic, otherwise it’s acyclic.

A DAG is a directed acyclic graph.

Dags are often used to model dependencies.

Let:

V = # of vertices

E = # of edges

How big could E be in relation to V?

E could be O(v^2)

Graph traversals

The problem with graph traversals is that we want to avoid cycles and print each vertex only once.

The solution is simple: mark each vertex the first time you visit it, and if it is already marked, then don’t visit it again.

Simple traversal strategy: use recursion!

DFS is implemented recursively and therefore uses a stack

If we use a queue instead, we would have a breadth first search (BFS)