Graph Theory *lite*

What am I looking at?

Daiwei Chen

<2019-03-06 Wed>

Contents

1 Graph Theory *lite*

When you would like to model problems with "complex" relationships.

Flights Catching flights and changing planes.

Internet Routing The internet is a giant graph. Figuring out the number of hops and which hops to take in order to get to your final destination.

Circuit Boards Making sure connections are efficient.

Basically, things that are connected to each other. A Graph Consists of a set of **vertices** and **nodes** that are *connected* by edges.
Tools: Python - Networkx, c/c++ - nauty. Assumptions:

- No Self Loops
- Edges are Unique

1.1 Paths

Path List of vertices l such that $(l[i], l[i+1]) \epsilon E$.
On each small step of a path, the 2 nodes much have an edge between them.

Simple Path Path without repeated vertices.

Cycle A (mostly) simple path but the 1^{st} and last vertices are the same. Only the beginning and end are repeated.

Path Length The number of edges in a path.

1.2 Graphs

Acyclic Graph A graph without **ANY** cycles.

Connected Graph A graph in which for some a there is an a path from every vertex to every vertex.

Tree Connected acyclic graph.

- $|v| - 1$ edges.

1.3 Subgraphs

Subgraph Only contains verticies and edges of the original graph. Edges must have 2 end-points.

Spanning Tree A subgraph of G that contains all nodes in G and is a tree.