Graphs and Simulation

28 February 2024
Chemical bonds
Dystopian notions of friendship
Each of these structures consists of

a collection of objects and

links between those objects.

We’d like to find a general framework for describing these objects and their properties.
A graph is a mathematical structure for representing relationships between entities.
A graph consists of a set of *nodes* (or *vertices*) connected by *edges* (or *arcs*).
A graph consists of a set of nodes (or vertices) connected by edges (or arcs).
A graph consists of a set of *nodes* (or *vertices*) connected by *edges* (or *arcs*).
Some graphs are *directed*. 
Some graphs are undirected.
Every undirected graph can also be represented as a directed graph, albeit with twice the edges.
How can we represent a graph?
data Graph:
  | vertex(name :: String, neighbors :: List<Graph>)
end
data Graph:
    | vertex(name :: String, neighbors :: List<Graph>)
end

This breaks horribly if there are cycles in the graph.
Better plan:

```plaintext
data Vertex:
  | vertex(name :: String, neighbors :: List<String>)
end
```

Now a graph is a List<Vertex>.

This is called an “adjacency list” representation.
Example: London Underground
This isn't a complete model. We could keep adding directional edges – and, in fact, all of the stations that are connected would eventually have edges going both ways.
Exercise

Let’s use a reactor to simulate a traveller riding the subway, following the connections in the graph.
Code:

https://tinyurl.com/bdewxa6f
Challenge exercise

Update the simulation to allow any number of travellers at the same time.
Code:

https://tinyurl.com/bpyz53b5
Acknowledgments

This lecture incorporates material from:

Keith Schwarz, Stanford University
Laney Strange, Northeastern University