COMP - 285

Analysis of Algorithms

Welcome to COMP 285

Lecture 12: Graph Introduction & BFS

Lecturer: Chris Lucas (cflucas@ncat.edu)

HW4 Released!

Due 10/13 @ 11:59PM ET

T-1 week until Midterm!

Midterm Review on Tuesday!

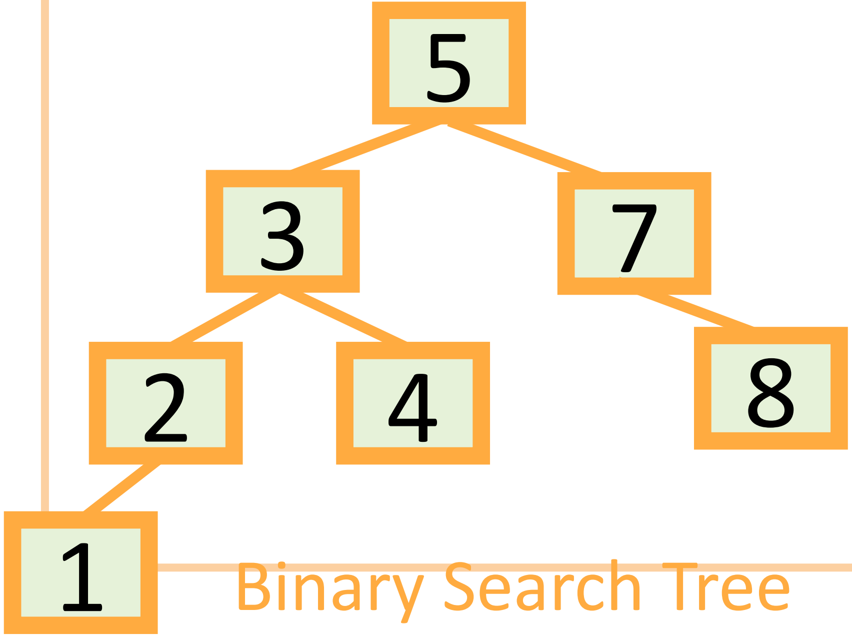
Suggest review topics on Piazza!

Practice Midterm Released by EoD!

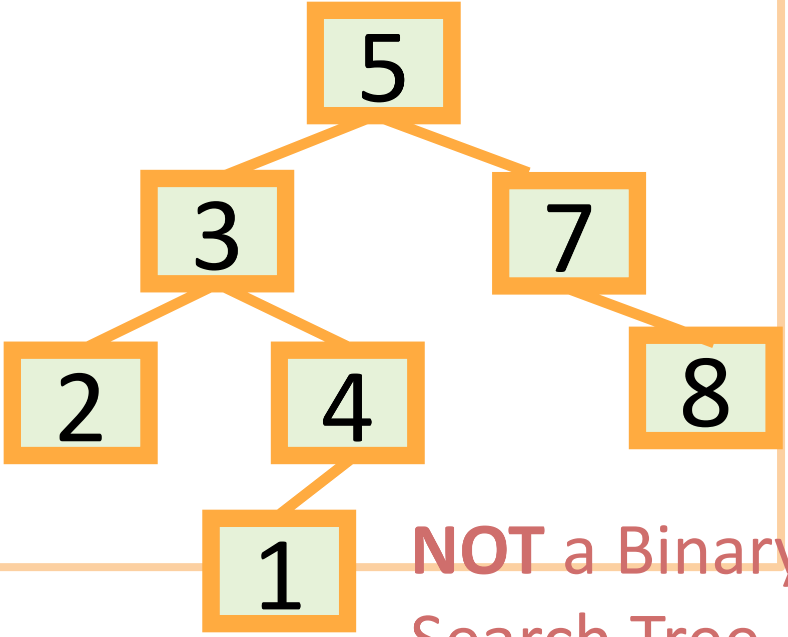
Recall where we ended last lecture…

Are these Binary Search Trees? Yes or No?

- A BST is a binary tree so that:
  - Every LEFT descendant of a node has value less than that node.
  - Every RIGHT descendant of a node has value larger than that node.

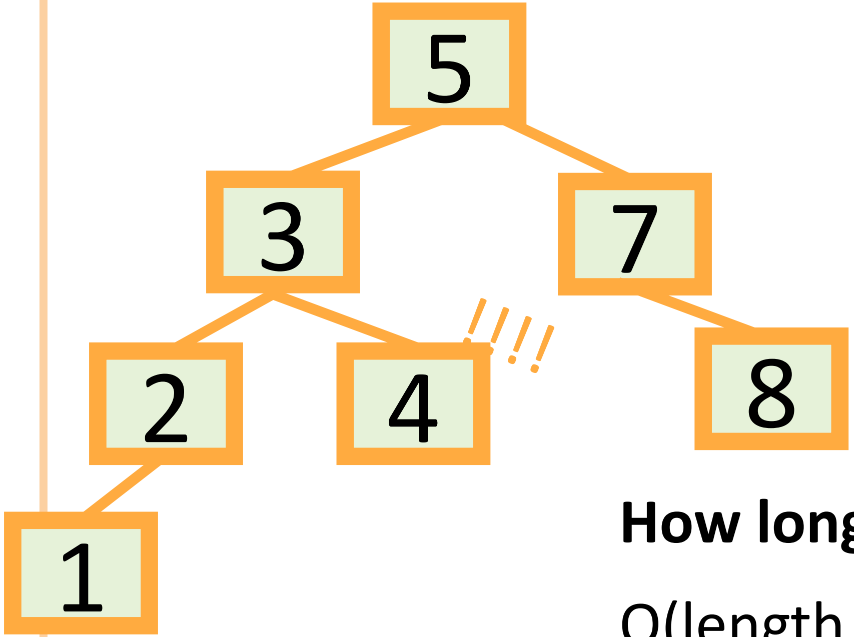


Binary Search Tree



NOT a Binary Search Tree

SEARCH in a Binary Search Tree



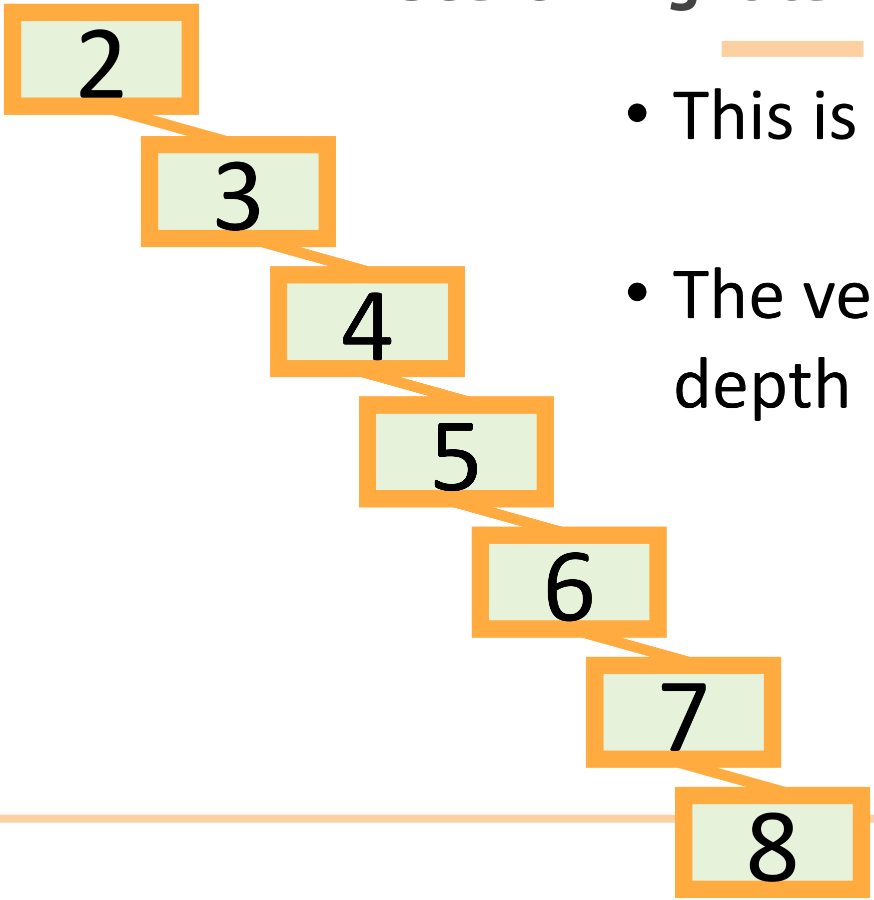
EXAMPLE: Search for 4.

- if value > x.value:
  - Recurse on x.rightChild
- if value < x.value:
  - Recurse on x.leftChild
- if x.value == value:
  - **Return x**

How long does this take?

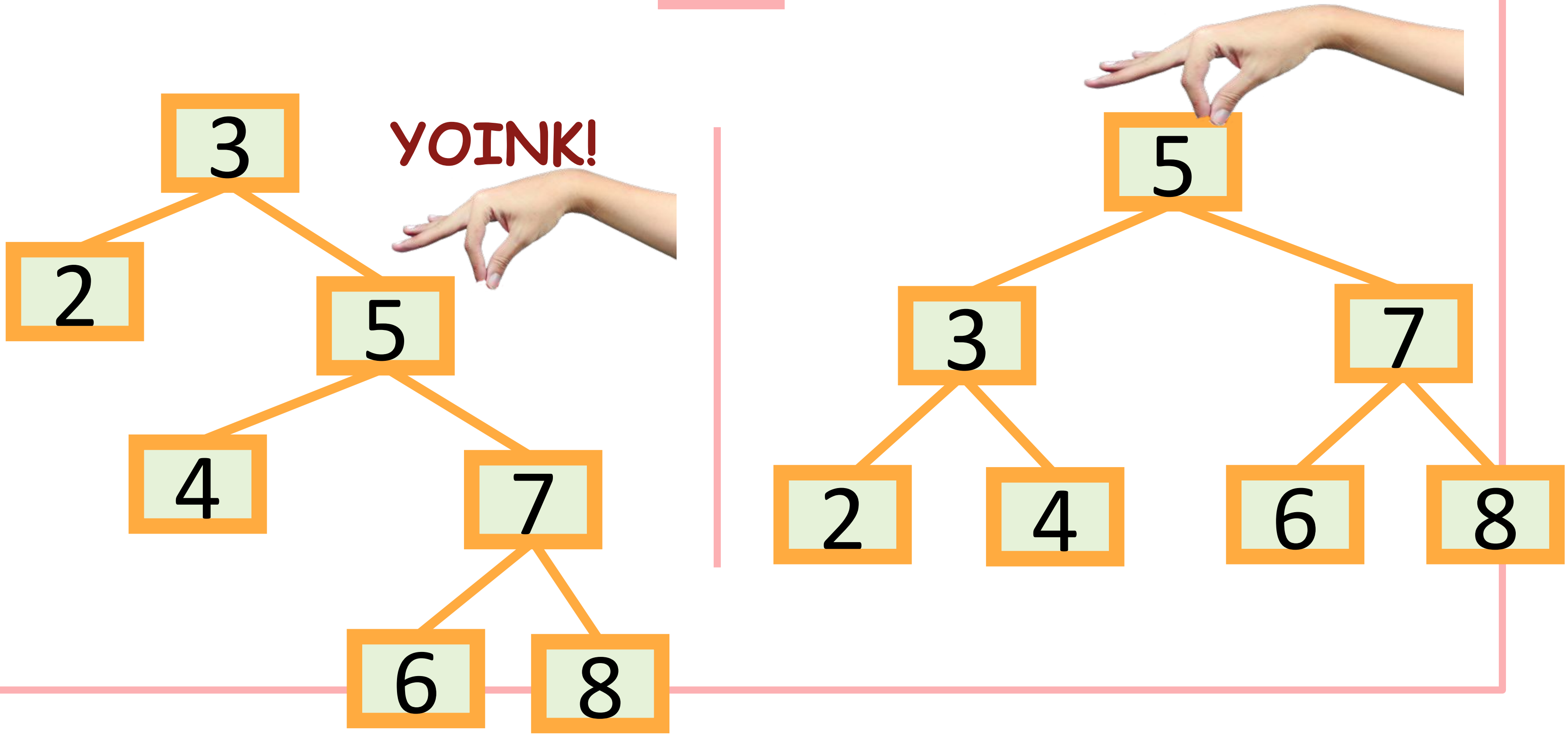
O(length of longest path) = O(height)

Search might take time O(n).



- This is a valid binary search tree.
- The version with n nodes has depth n, **not** O(log(n)).

Self-balancing trees and rotations!



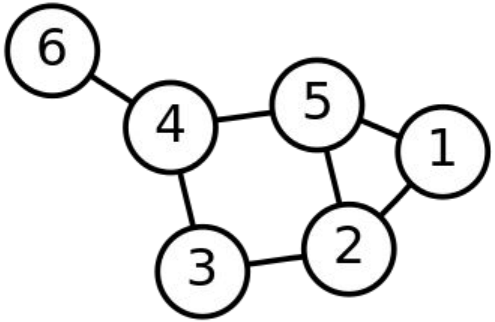
Big Questions!

- What are graphs?
- What are examples of graphs?
- How do we represent graphs?
- How to find a viable path in a graph?

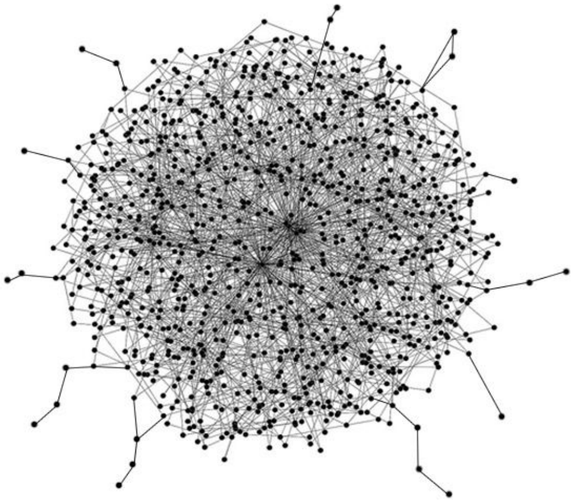
Big Questions!

- What are graphs?
- What are examples of graphs?
- How do we represent graphs?
- How to find a viable path in a graph?

Motivations for Graphs



- In software engineering, graphs are everywhere:
  - Social networks (e.g. Facebook)
  - Search Engines (e.g. Google Search)
  - Routing / Map Directions (e.g. Google Maps)
  - … and much more (see next slides)

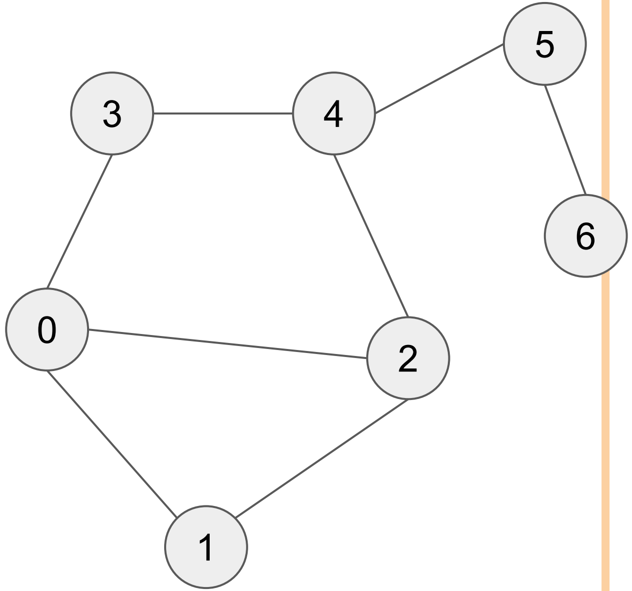


- Graphs and writing graph algorithms (or doing graph-like traversals) are very common whiteboard coding interview questions.

Graphs

G = (V, E) is an undirected graph where
V = {0, 1, 2, 3, 4, 5, 6}
E = {(0,1),(0,2),(0,3),(1,2),(2,4),(3,4),(4,5),(5,6)}

- Graphs are composed of **vertices** (AKA nodes) and **edges**.
- Each vertex is connected to other vertices with edges. Each vertex has a **degree**.
- The edges can be **directed** or **undirected**.
- Graphs may contain cycles or otherwise be acyclic.
- When two vertices are connected, we can say they are **neighbors** and that they are **adjacent** to one another.
- The notation G = (V, E) stands for “Graph G has a set of vertices V = {v_1, v_2, …} and set of edges E = {e_1, e_2, ... }”

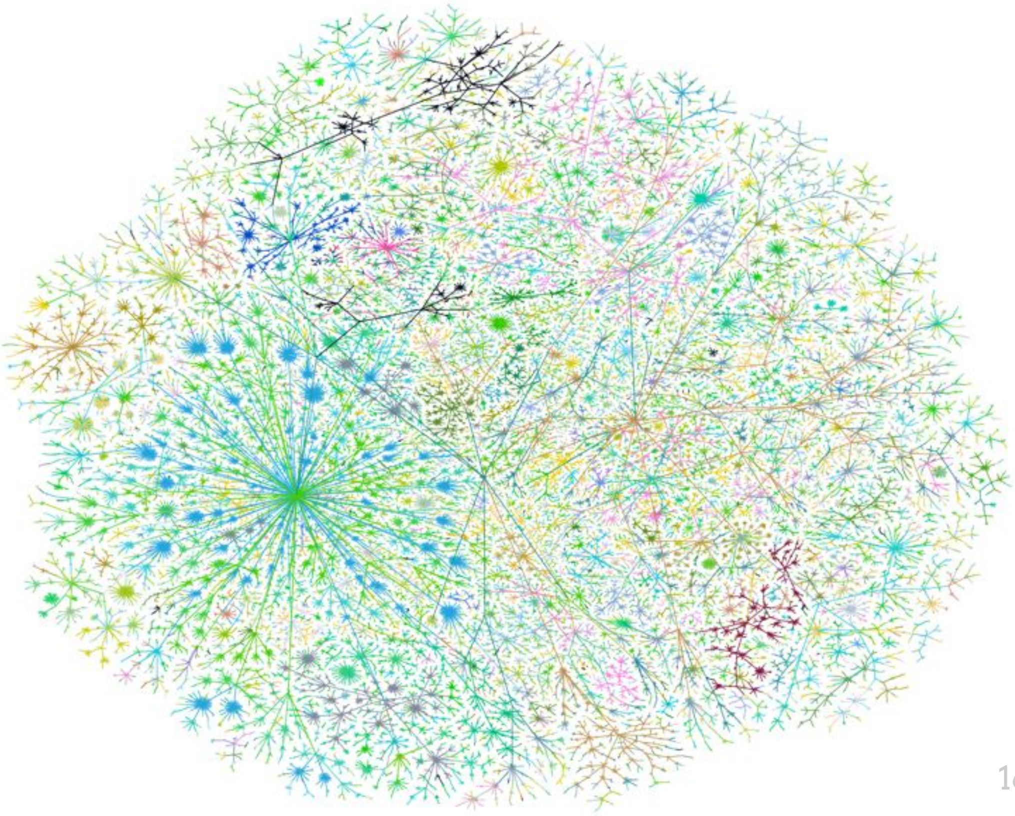


Big Questions!

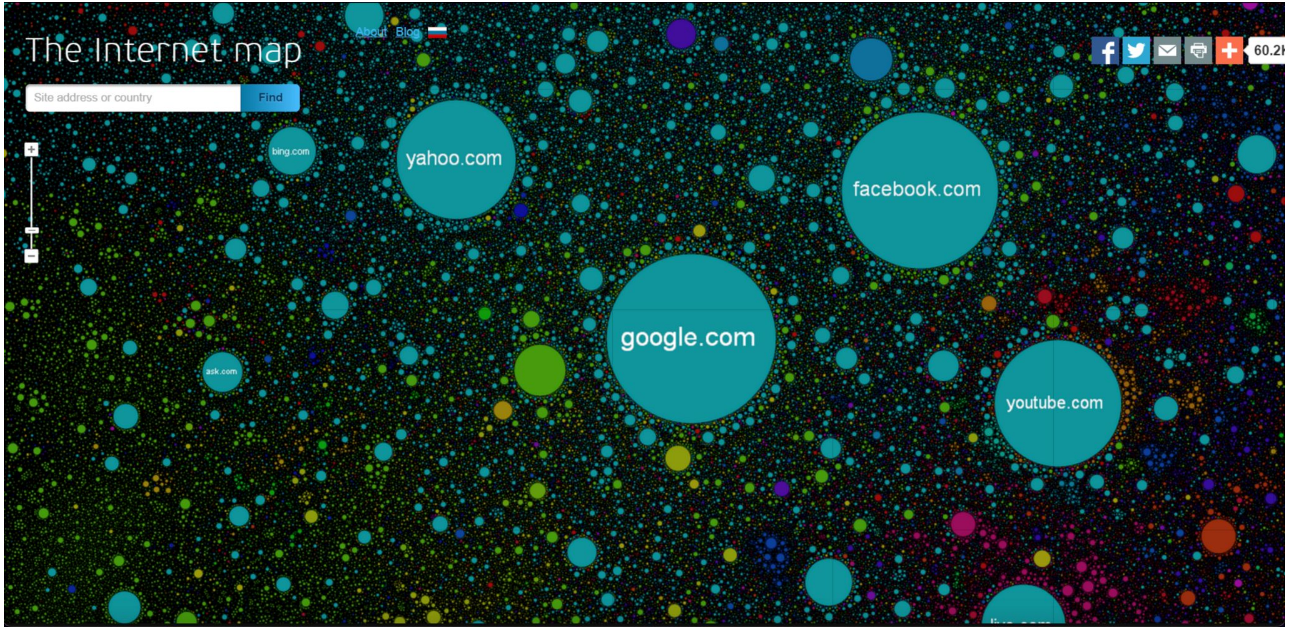
- What are graphs?
- What are examples of graphs?
- How do we represent graphs?
- How to find a viable path in a graph?

Graphs

Graph of the internet (circa 1999…it's a lot bigger now…)

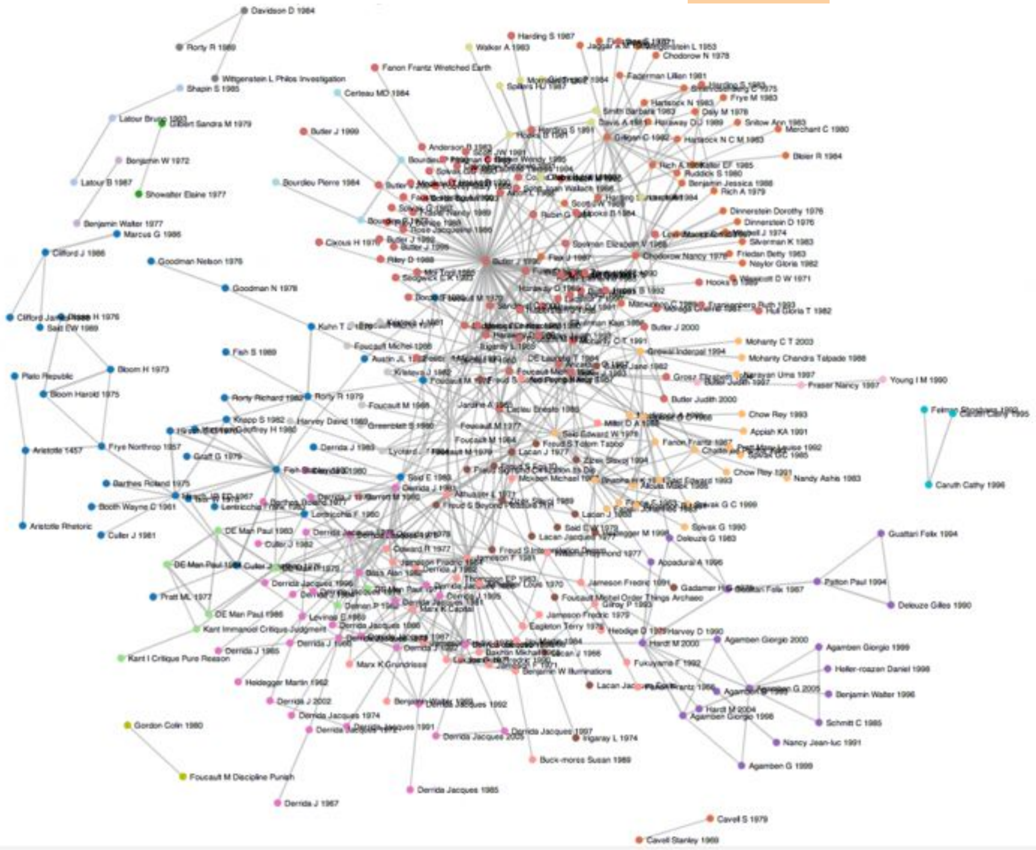


Graphs



So much so, I found a nifty [visualization tool]…

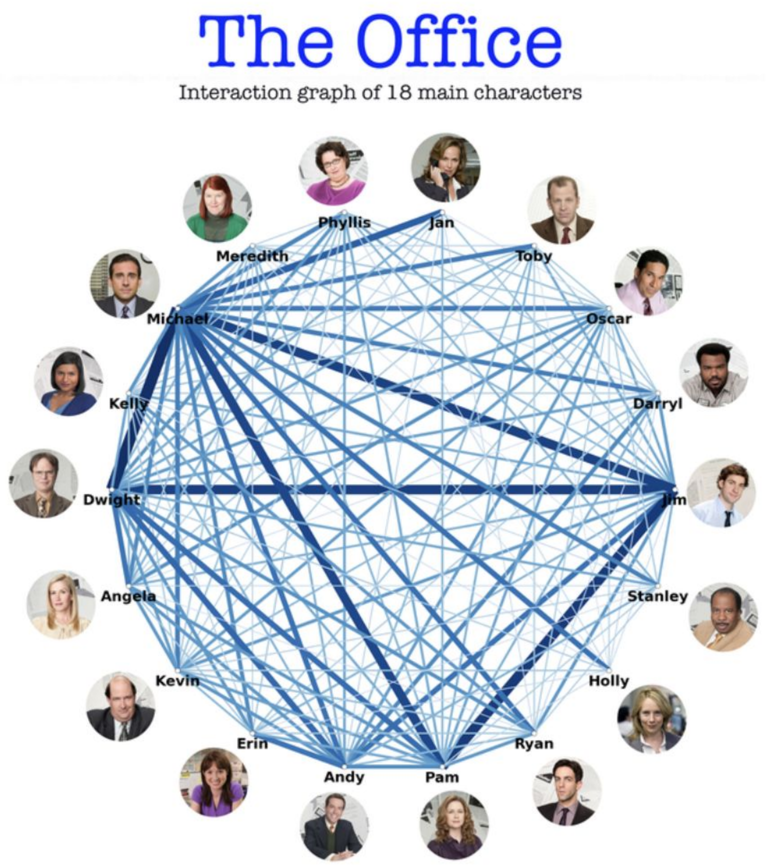
Graphs



Citation graph of literary theory academic papers

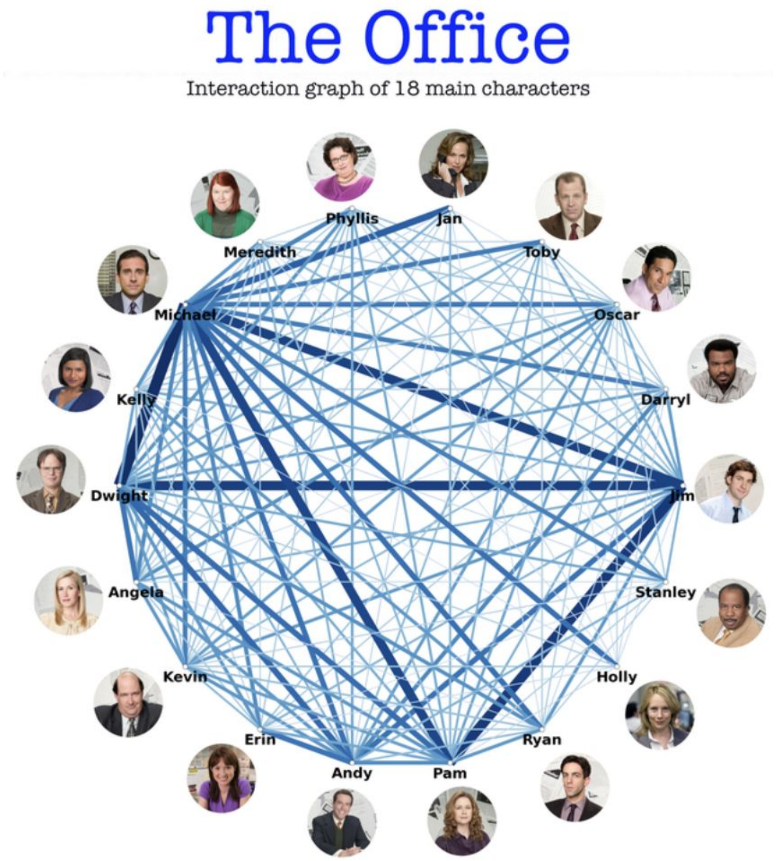
Graphs

The Office characters interaction network

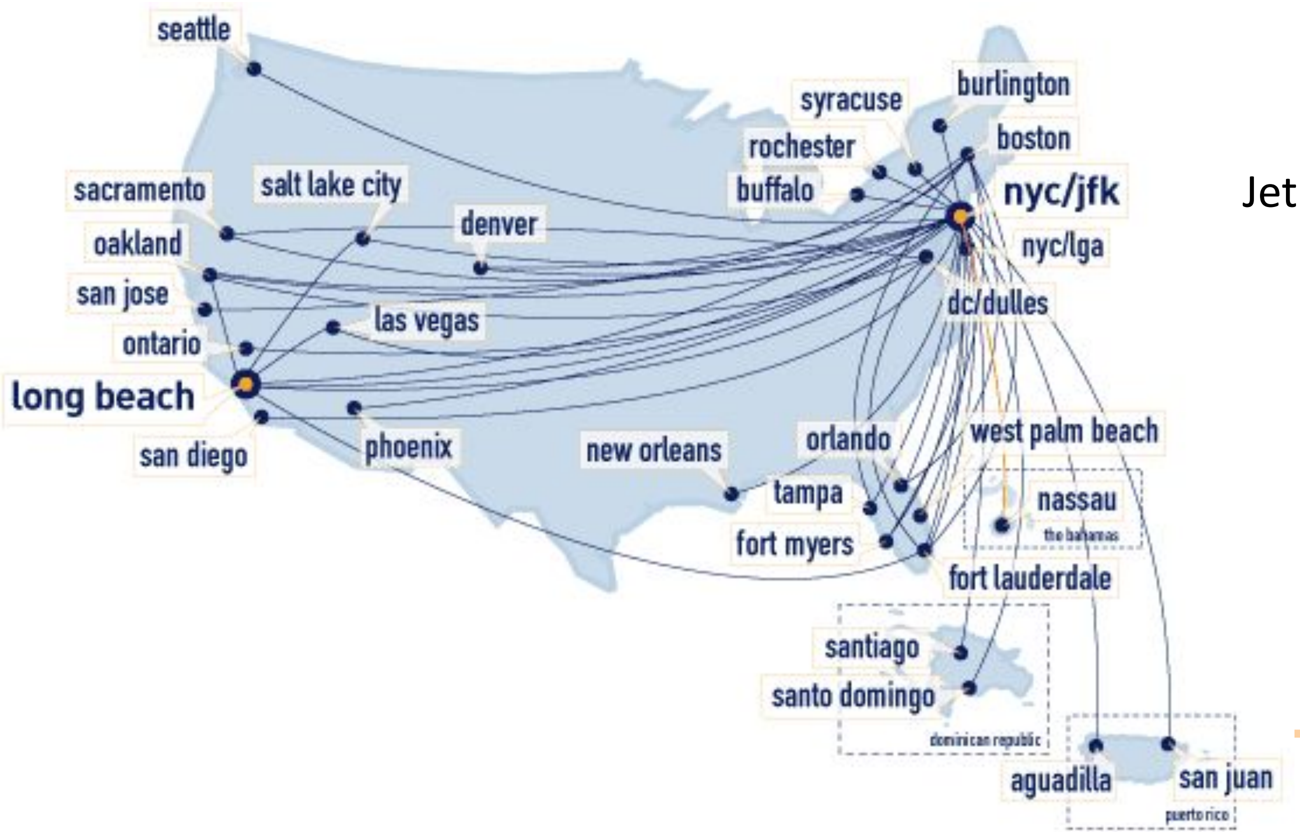


Graphs

The Office characters interaction network
why the different thicknesses?



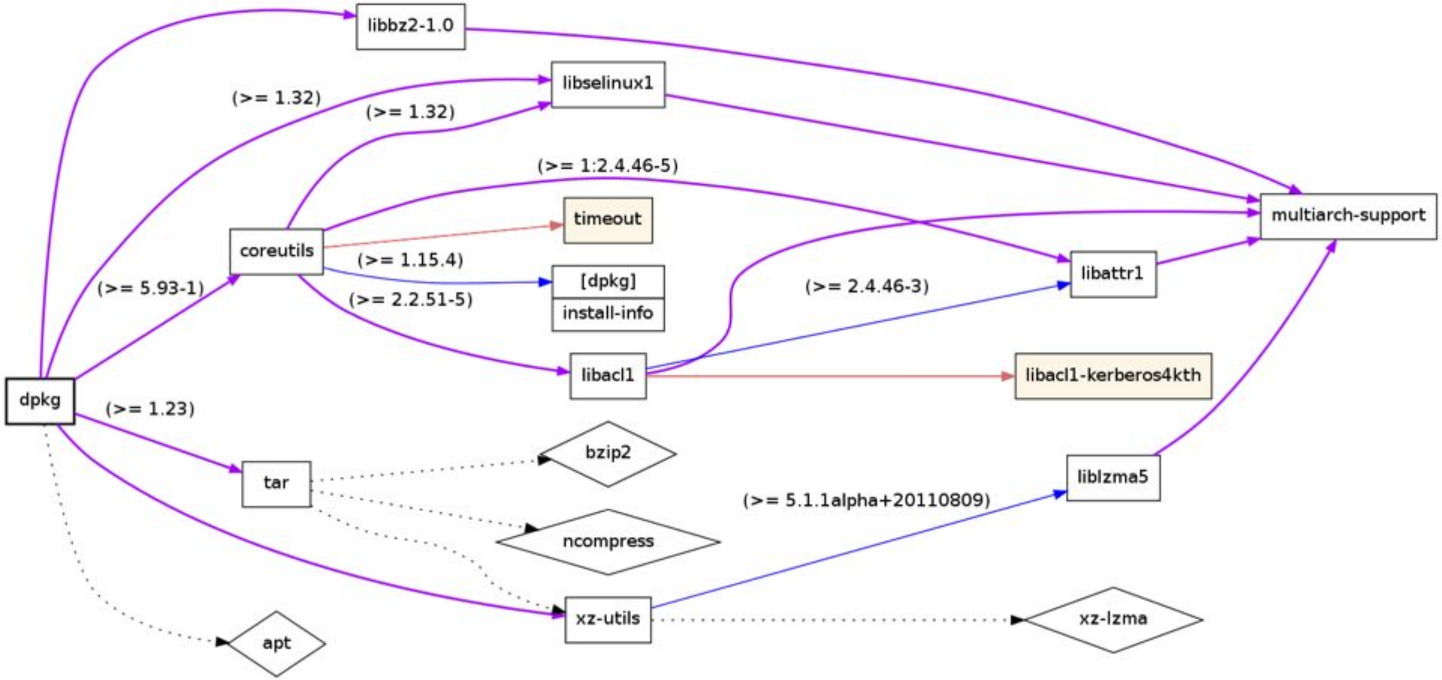
Graphs



JetBlue flight route

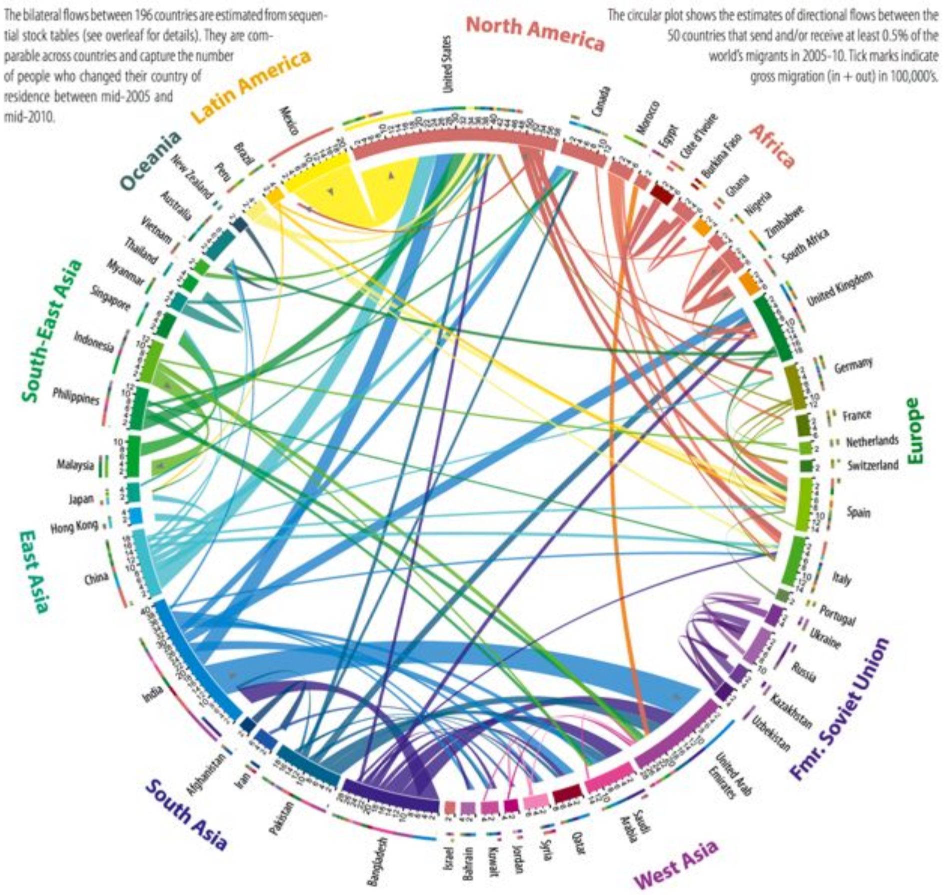
Graphs

Debian dependency (sub)graph



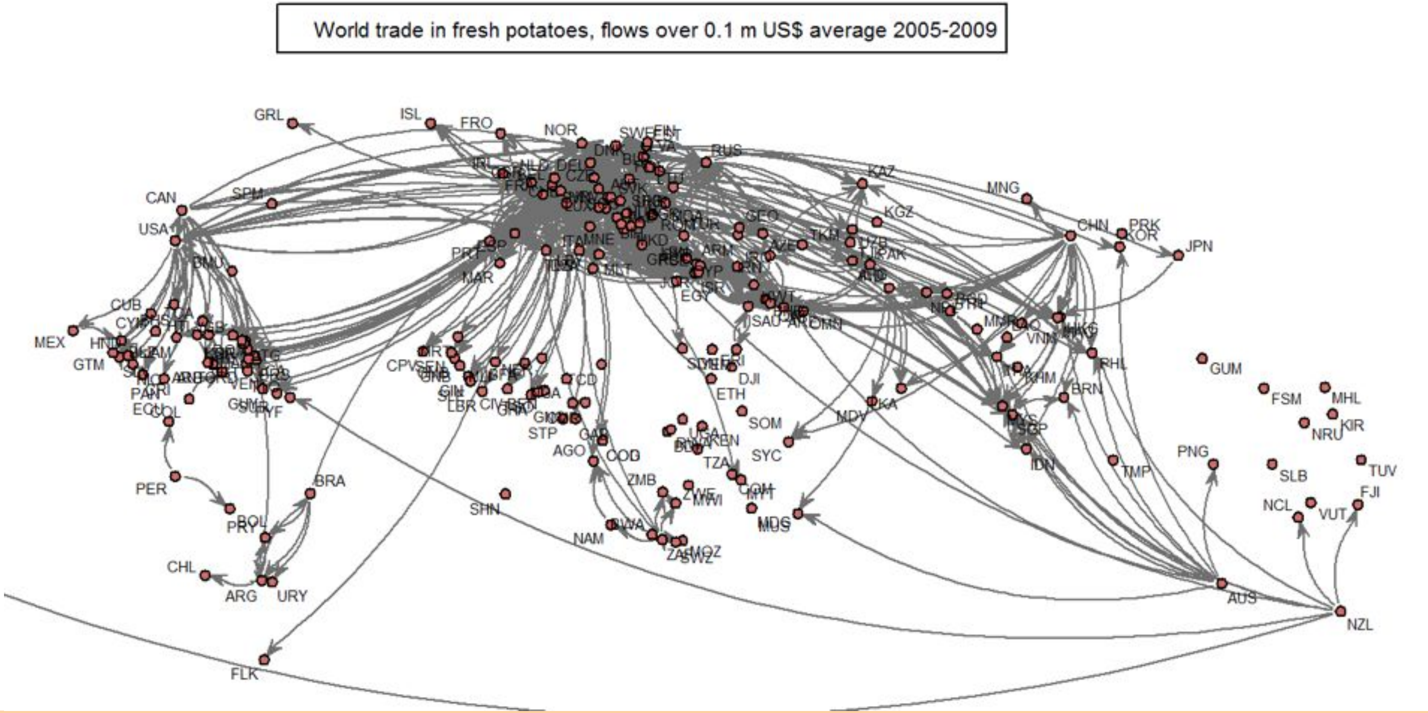
Graphs

Immigration flows

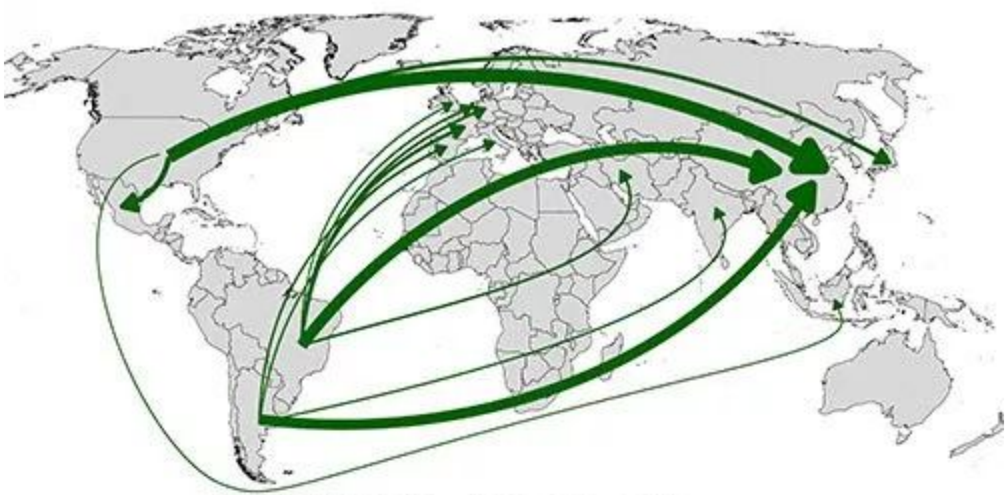


Graphs

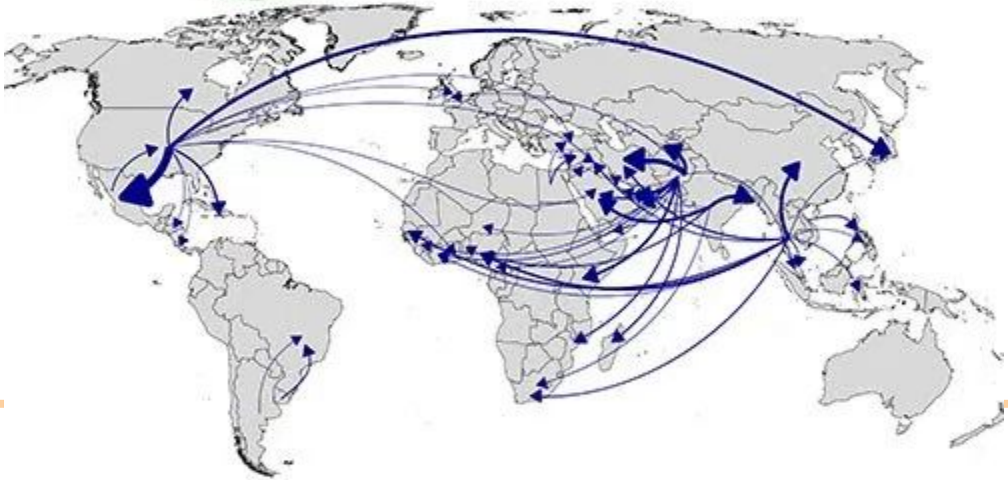
Potato trade



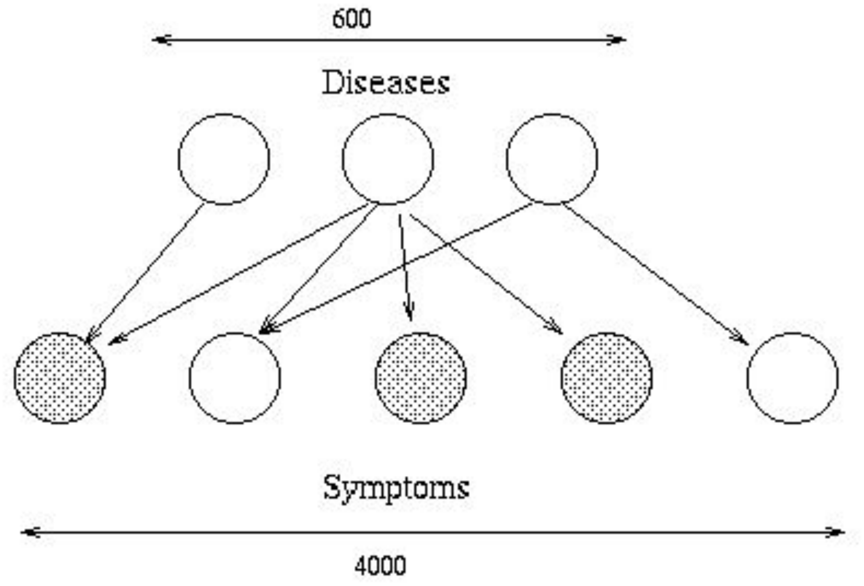
Soybeans



Water



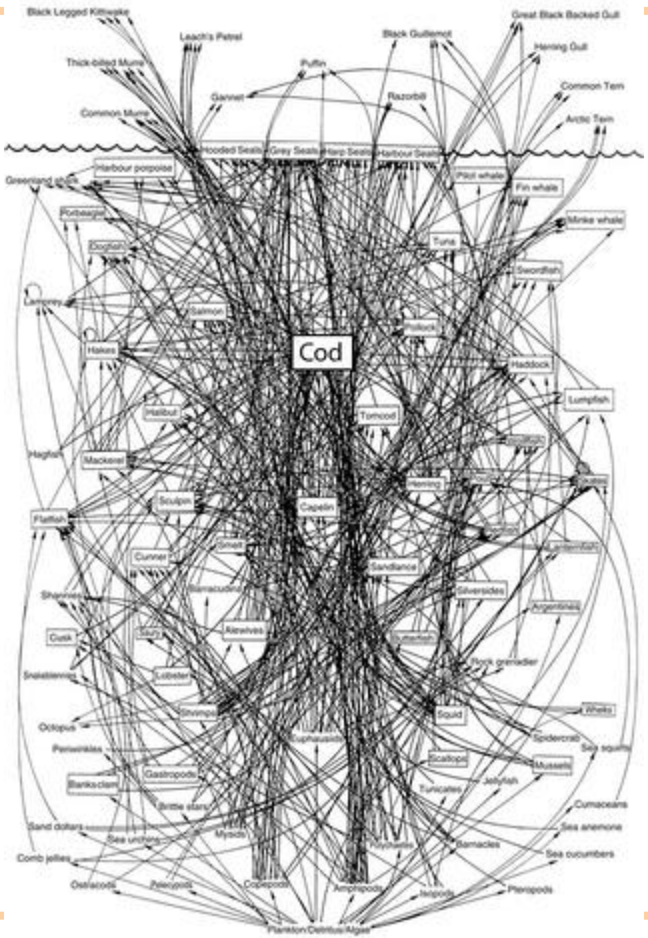
Graphs



Medical applications, disease/symptoms

Graphs

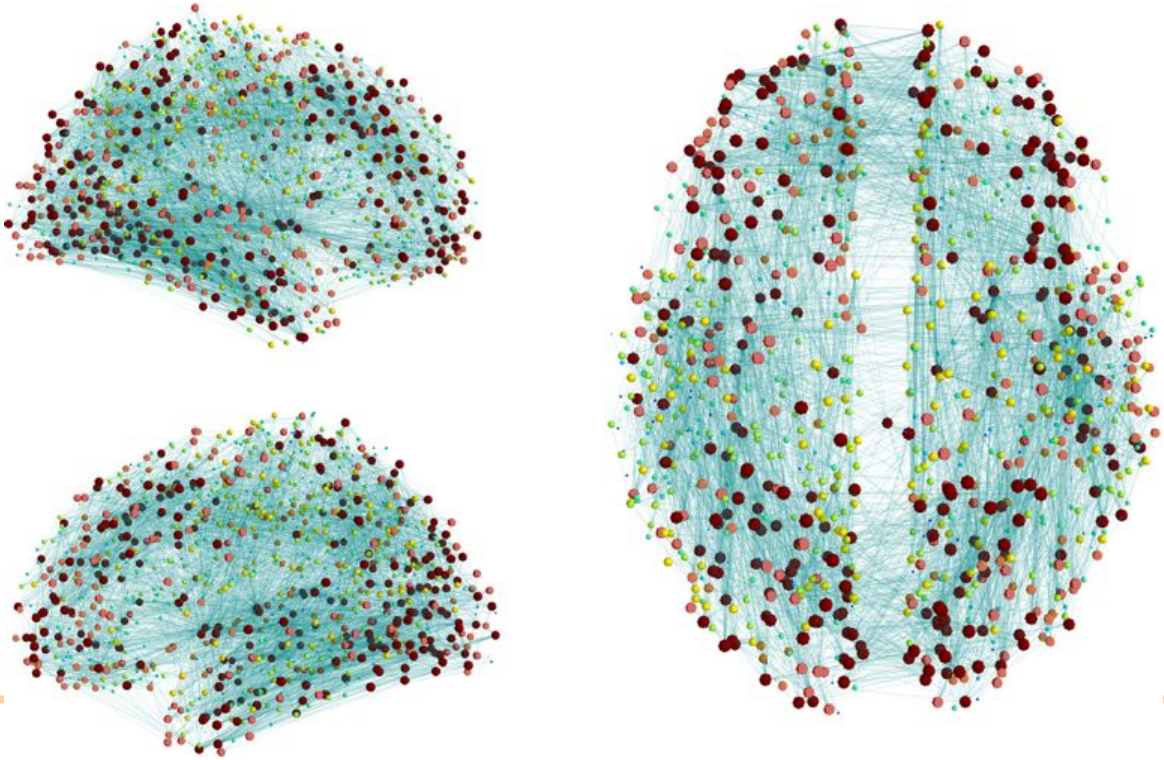
What eats what in the Atlantic ocean?



A simplified food web for the Northwest Atlantic. © IMMA

Graphs

Neural connections in the brain



Graphs

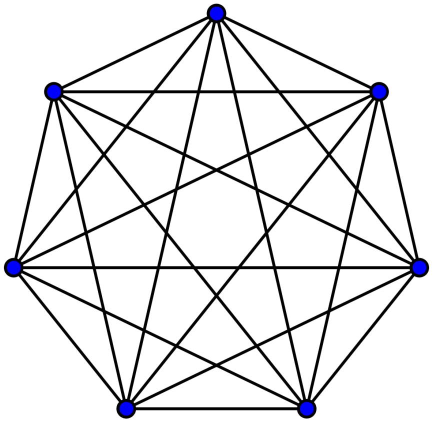
- There are a lot of graphs.
- We want to answer questions about them.
    - Efficient routing?
    - Community detection/clustering?
    - Computing Bacon numbers
    - Signing up for classes without violating pre-req constraints
    - How to distribute fish in tanks so that none of them will fight.
- This is what we'll do for the next several lectures.

Notation: V & E

- For a graph G = (V, E), V represents the set of all vertices and E represents the set of all edges.
- |V| is the cardinality or size of V & |E| is the cardinality or size of E
- When analyzing/discussing time/space complexity in terms of |V| and |E|, we may exclude the extra characters and just write O(V) when we really mean O(|V|)

By the way… |E| = O(???)

1. O(V)
2. $O(V^2)$
3. $O(V^3)$
4. O(V!)

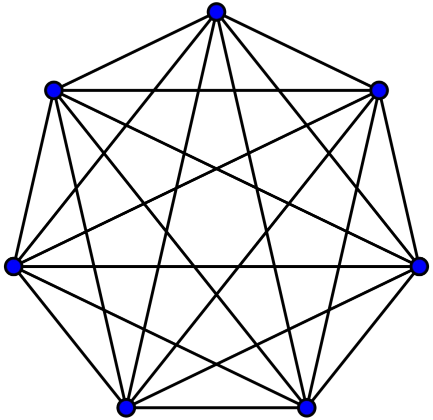


Notation: V & E

- For a graph G = (V, E), V represents the set of all vertices and E represents the set of all edges.
- |V| is the cardinality or size of V & |E| is the cardinality or size of E
- When doing runtimes / discussion sizes in terms of |V| and |E|, we may exclude the extra characters and just write O(V) when we really mean O(|V|)

By the way… |E| = O(???)

1. $O(V)$
2. $O(V^2)$
3. $O(V^3)$
4. $O(V!)$



$(V - 1) + (V - 2) + \ldots + 1$
$= V(V-1)/2 = O(V^2)$

Kahoot!

www.kahoot.it, Code: 956 2288

Enter your @aggies.ncat email

Polls

How many edges are there?

 8

How many neighbors does 0 have?

 3

Are vertex 2 and 6 adjacent?

 No

What’s the degree of node 1?

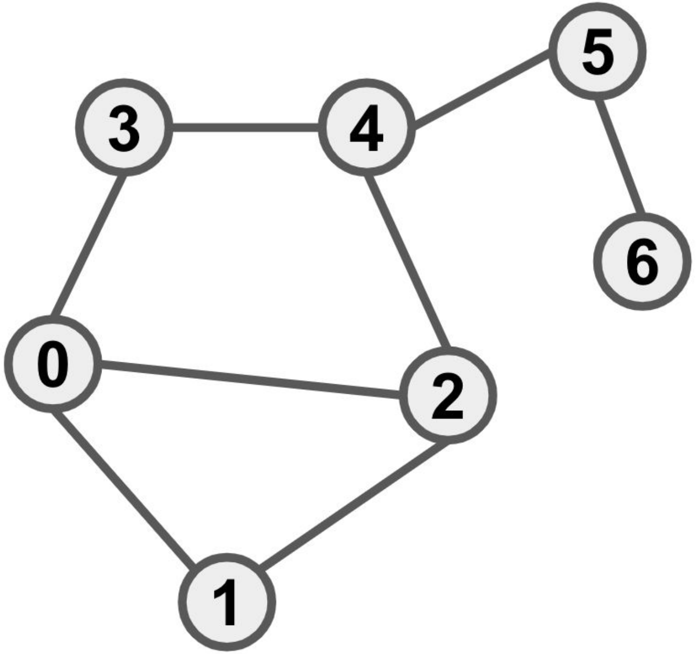
 2

Which vertex has the most neighbors?

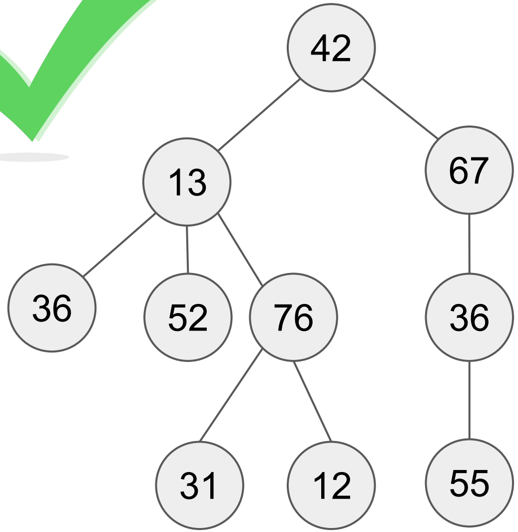
 Tie between 0, 2, and 4

How many distinct paths are there from 1 to 5?

 4

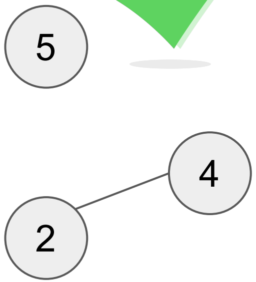


Polls - Are these graphs?



Trees are graphs!

No edges!



Disconnected!

Big Questions!

- What are graphs?
- What are examples of graphs?
- How do we represent graphs?
- How to find a viable path in a graph?

Graph Representations

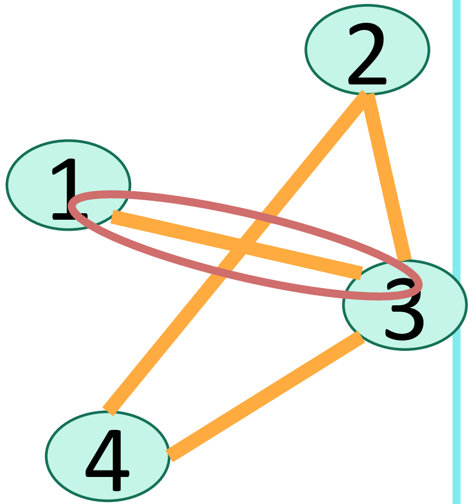
The two most common representations of Graphs:

- **Adjacency Matrix**: a 2D array (table) where the indices (rows and columns) represent vertices, and each entry at (i, j) will have a 1 if an edge is present (and have a 0 if no edge is present).
- **Adjacency List**: A list of lists, where each list at index i represents all neighbors for a vertex i.

How do we represent graphs?

- Option 1: adjacency matrix

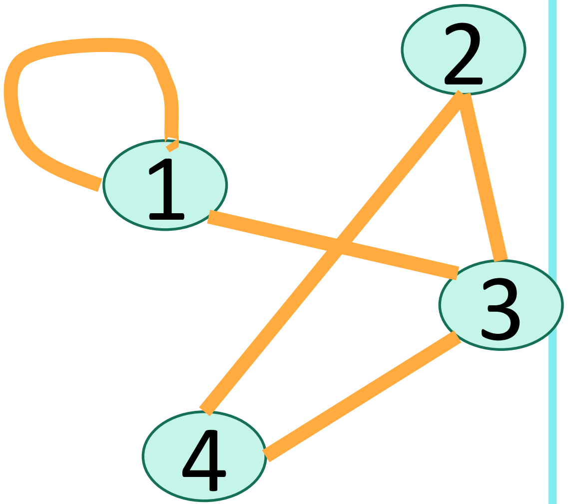
$$
\begin{array}{c c}
 & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\
\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}
\end{array}
$$



How do we represent graphs?

- Option 1: adjacency matrix

$$
\begin{array}{c c}
 & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\
\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}
\end{array}
$$



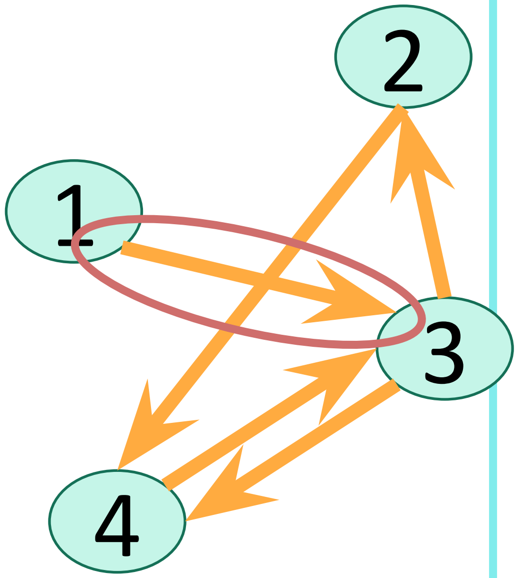
How do we represent graphs?

- Option 1: adjacency matrix

Destination

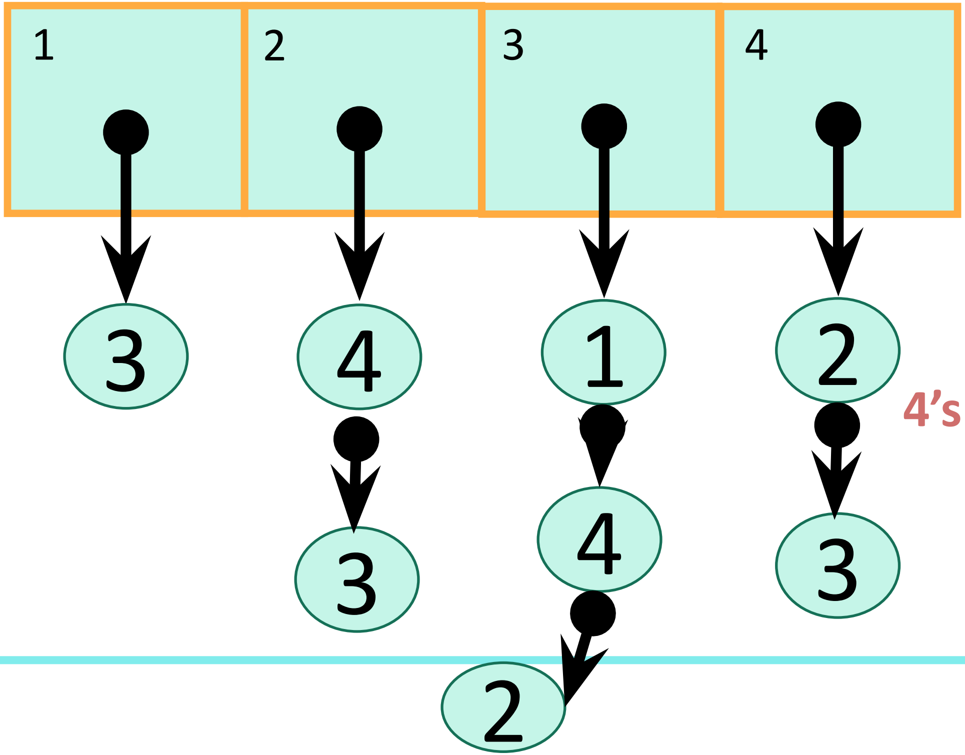
Source

$$
\begin{array}{c c}
 & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\
\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}
\end{array}
$$



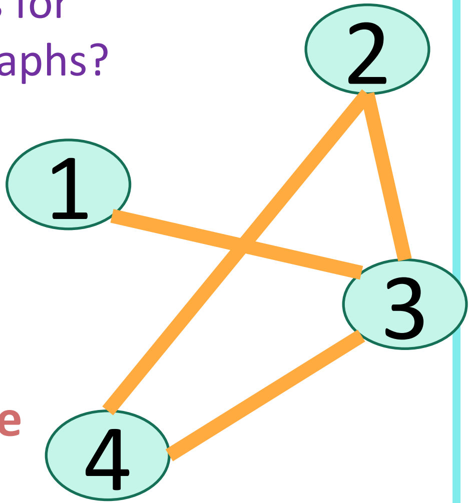
How do we represent graphs?

- Option 2: Adjacency lists



How would you modify this for directed graphs?

4's neighbors are 2 and 3

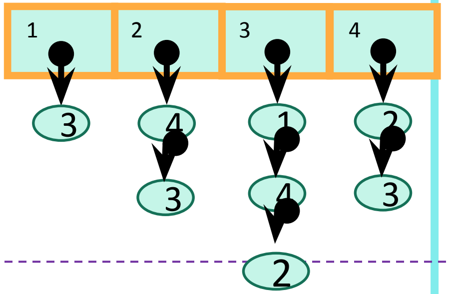


In either case

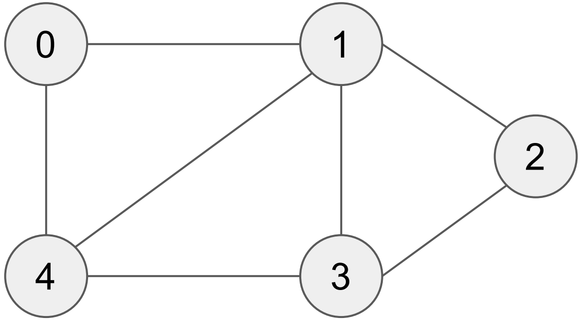
- Vertices can store other information
  - **Attributes** (name, IP address, …)
  - **Metadata** (helper info for algorithms that we will perform on the graph)
- Want to be able to do the following operations:
  - **Edge Membership** Is edge e in E?
  - **Neighbor Query** What are the neighbors of vertex v?

Trade-offs

Generally better for **sparse** graphs (where $m \ll n^2$)

Say there are V vertices and E edges.	$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$	
Edge membership Is e = {v,w} in E?	O(1)	O(deg(V)) or O(deg(W))
Neighbor query Find all of v's neighbors.	O(V)	O(1)
Space requirements	$O(V^2)$	O(V + E)

Graph Representation Examples



Graph

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

Adjacency Matrix

0: [1, 4]
1: [0, 4, 2, 3]
2: [1, 3]
3: [1, 4, 2]
4: [3, 0, 1]

Adjacency List

Graph Problem Example 1

In a town, there are n people labeled from 0 to n - 1.

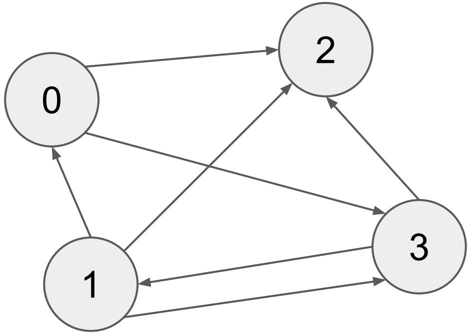
There is a “town judge”: (1) The town judge trusts nobody. (2) Everybody (aside from the town judge) trusts the town judge. (3) There is exactly one person that satisfies 1 and 2.

You are given an array trust of pairs where trust[i] = {a,b} representing that the person labeled a trusts the person labeled b.

Return the label of the town judge if the town judge exists and can be identified, or return -1 otherwise.

What would be the nodes and edges?

Nodes are people, and there is a directed edge from a to b if a trusts b.



	0	1	2	3
0	0	0	1	1
1	1	0	1	1
2	0	0	0	0
3	0	1	1	0

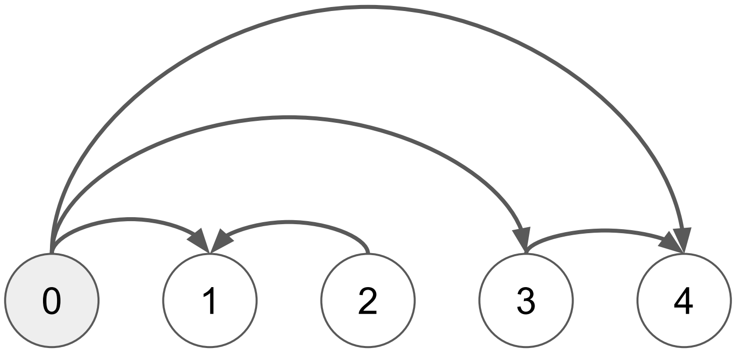
Graph Problem Example 2

There are N rooms and you start in room 0. Each room may have some keys to access other rooms.

All rooms start locked (except room 0). You can walk freely between the rooms.

Given an array rooms where rooms[i] is the set of keys that you can obtain if you visited room i. Return whether or not you can visit all the rooms.

What would be the nodes and edges? Nodes are rooms, and there is a directed edge from a to b if a has a key to b.



COMP - 285

Analysis of Algorithms

Welcome to COMP 285

Lecture 12: Graph Introduction & BFS

Lecturer: Chris Lucas (cflucas@ncat.edu)