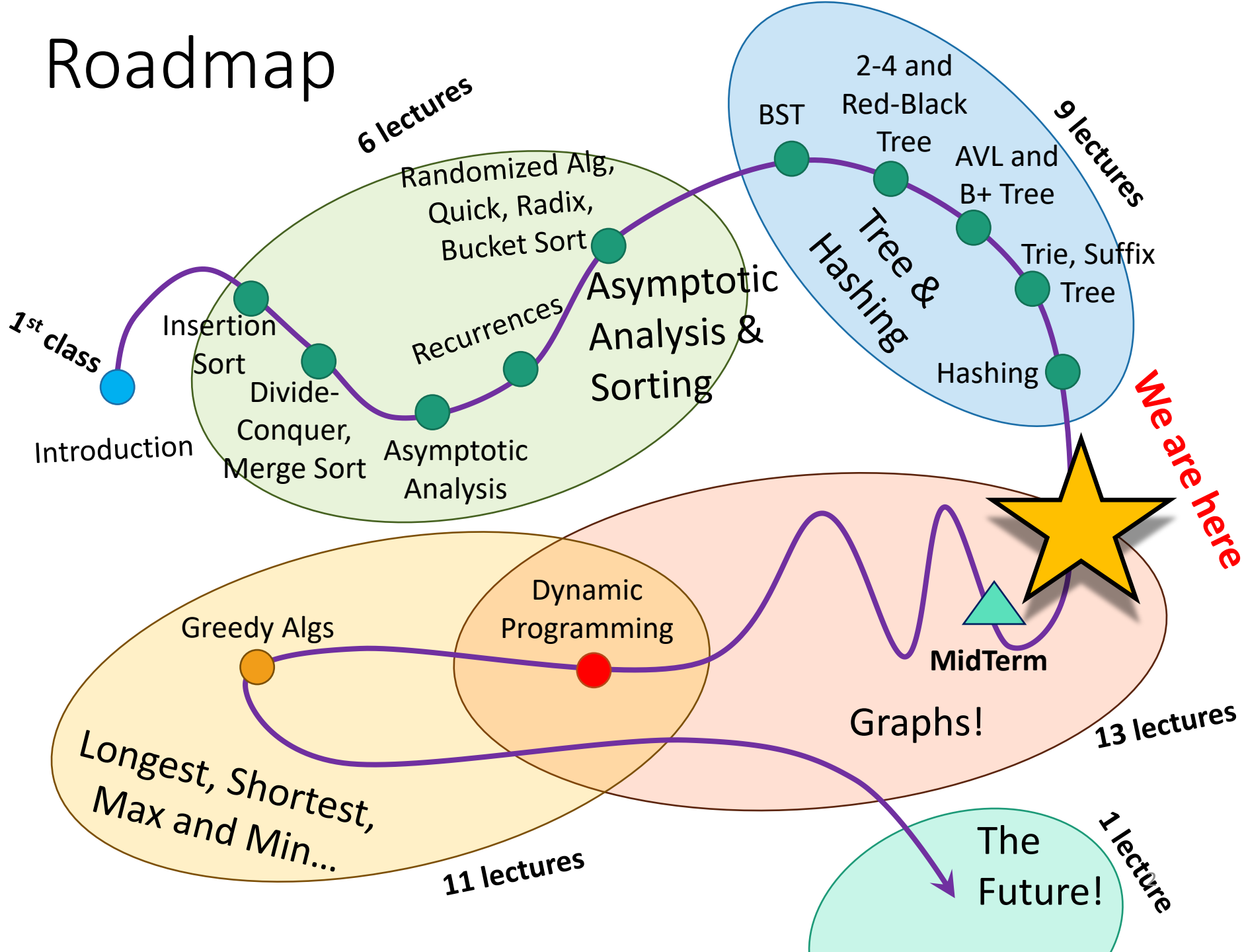


Advanced Data Structures and Algorithms

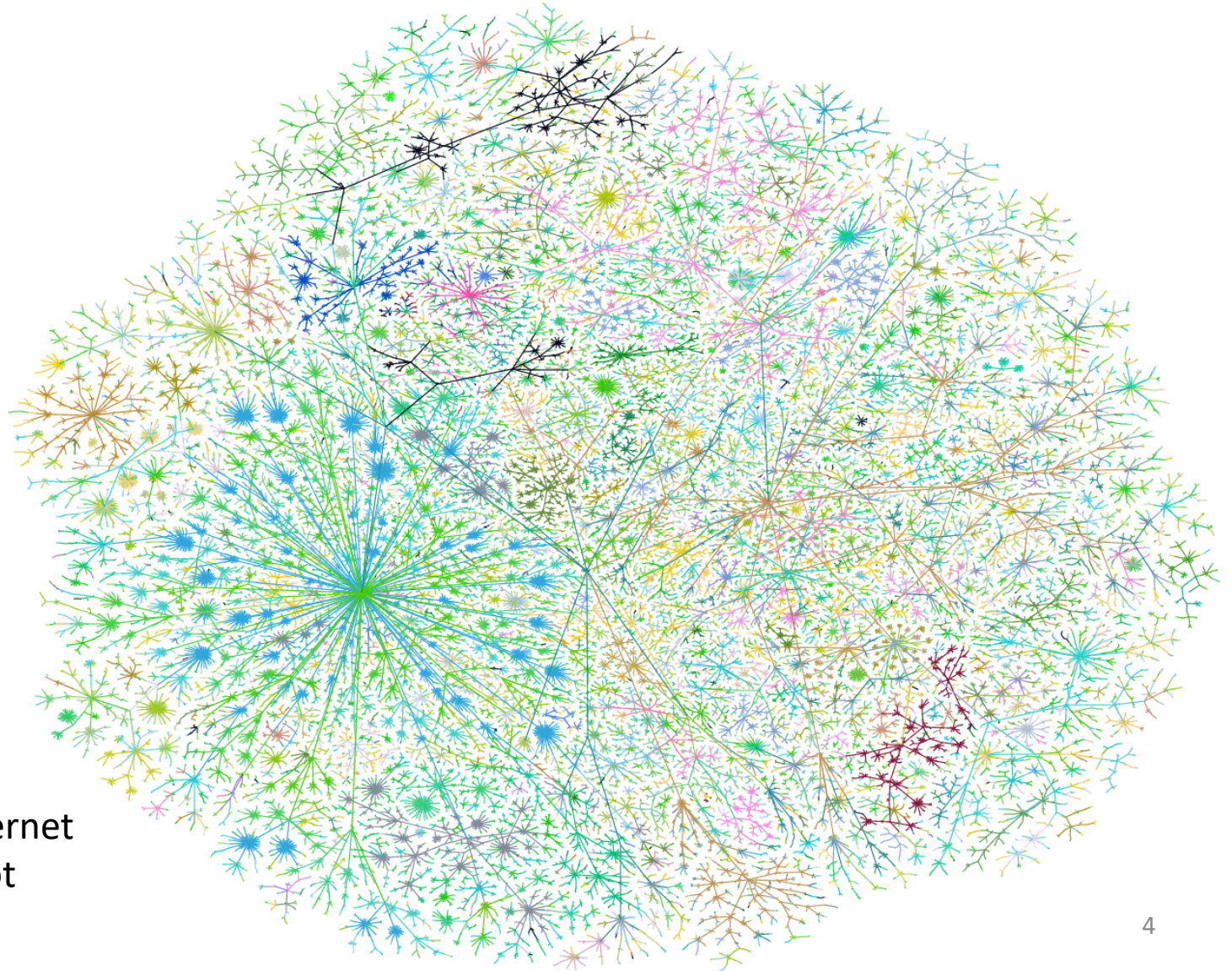
Graphs

Roadmap



Graphs

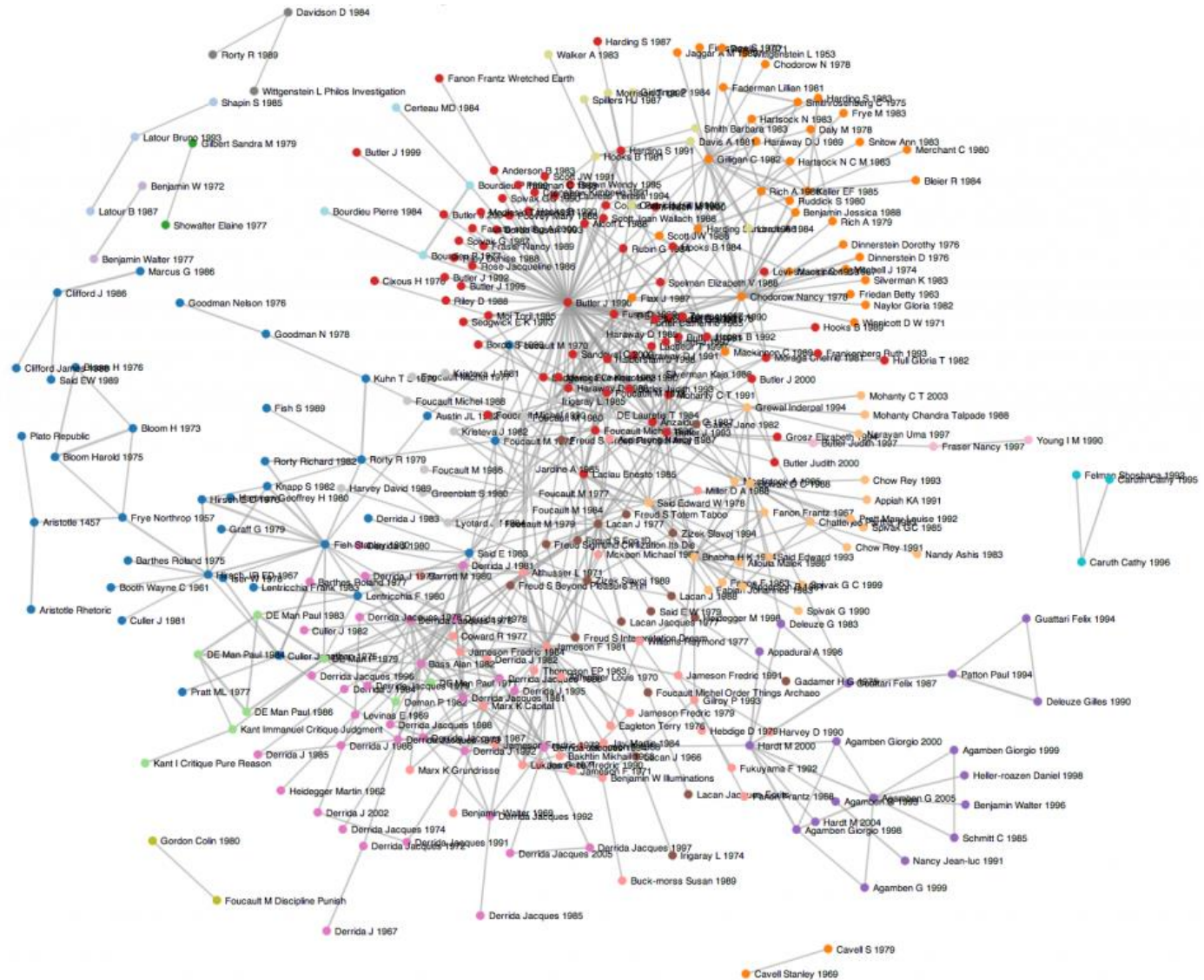
Graphs



Graph of the internet
(in 1999...it's a lot
bigger now...)

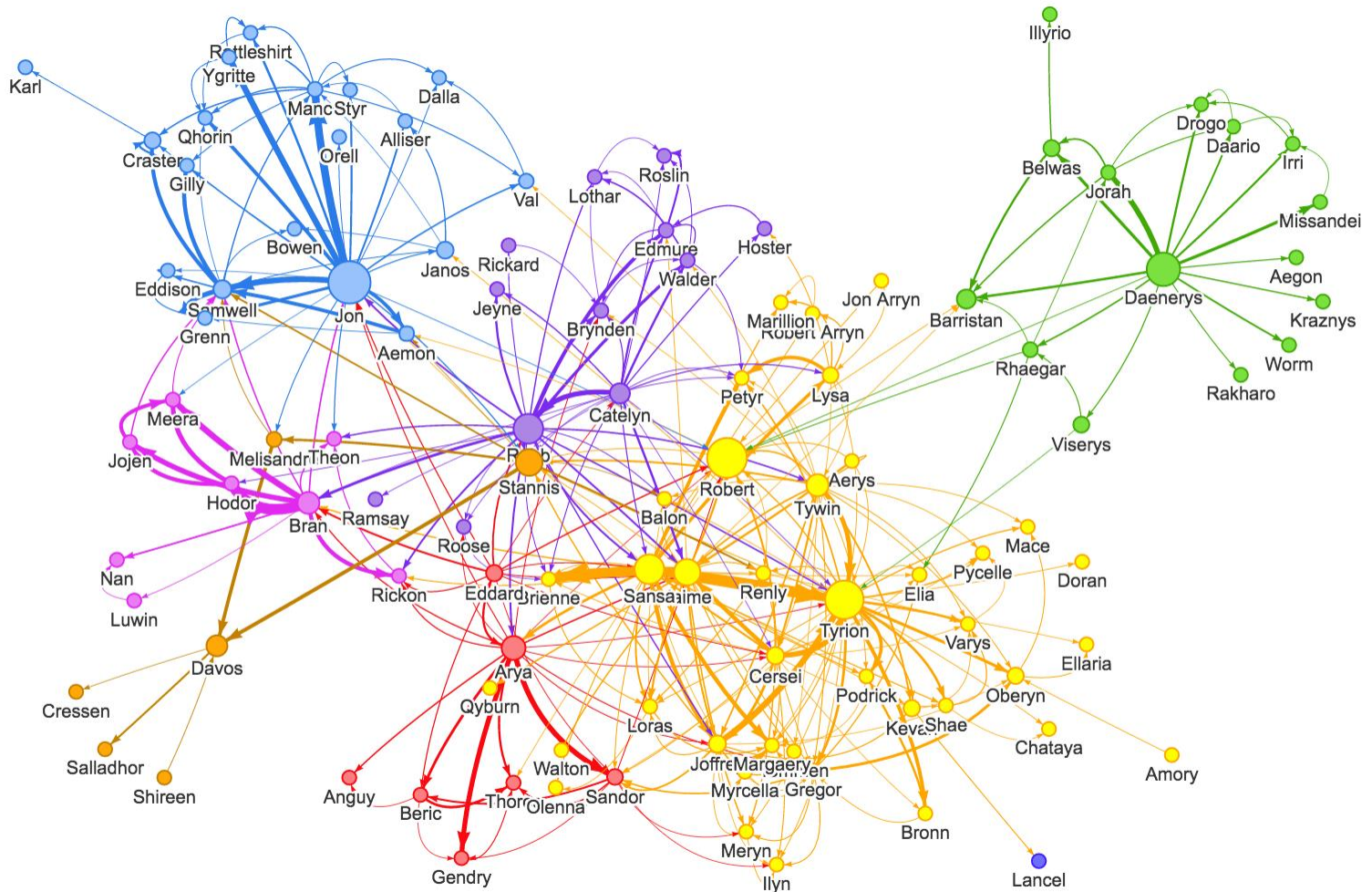
Graphs

Citation graph of
literary theory
academic papers



Graphs

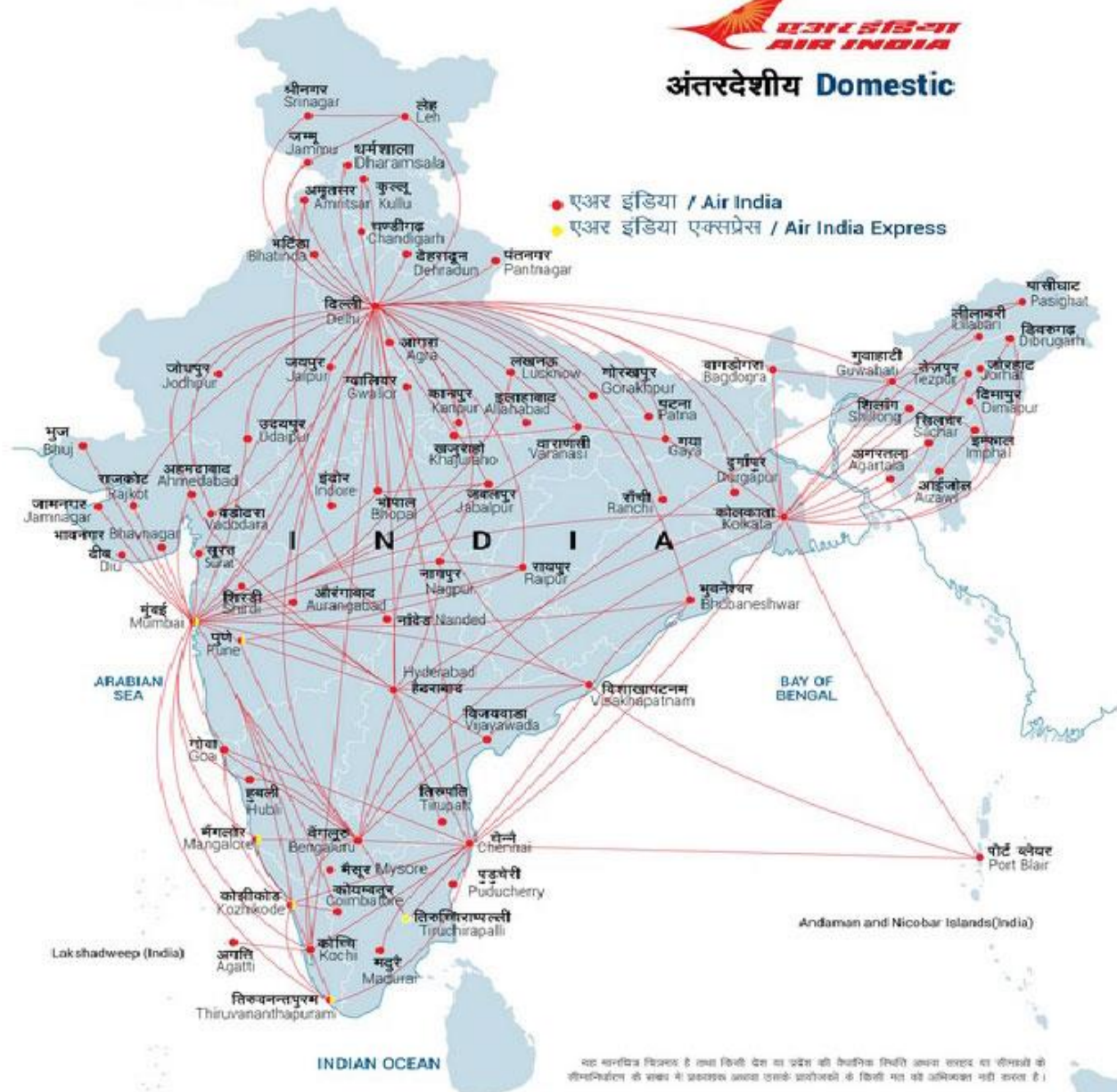
Game of Thrones Character Interaction Network



Graphs

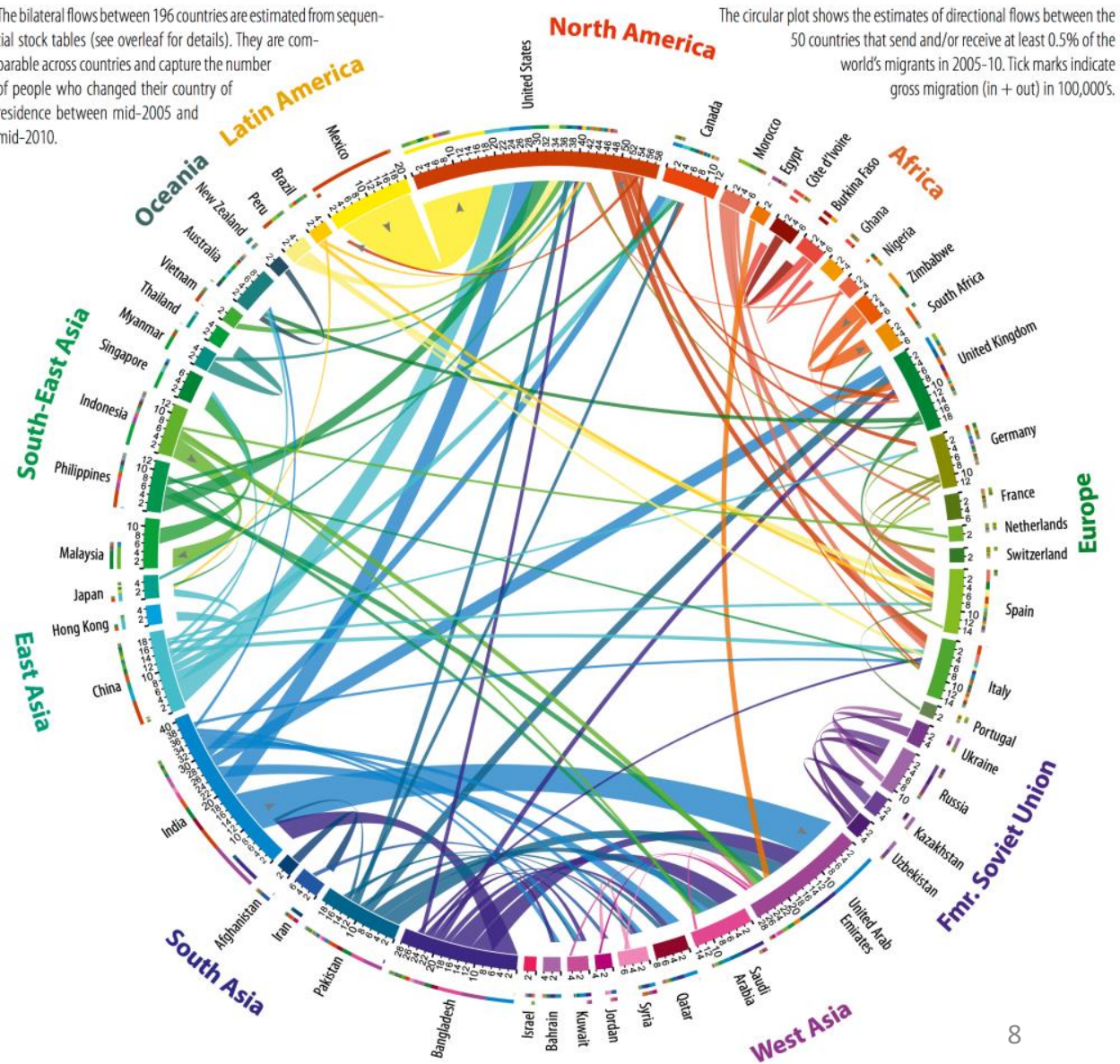
AIR INDIA
route map

एअर इंडिया
AIR INDIA
अंतरदेशीय Domestic



Graphs

The bilateral flows between 196 countries are estimated from sequential stock tables (see overleaf for details). They are comparable across countries and capture the number of people who changed their country of residence between mid-2005 and mid-2010.

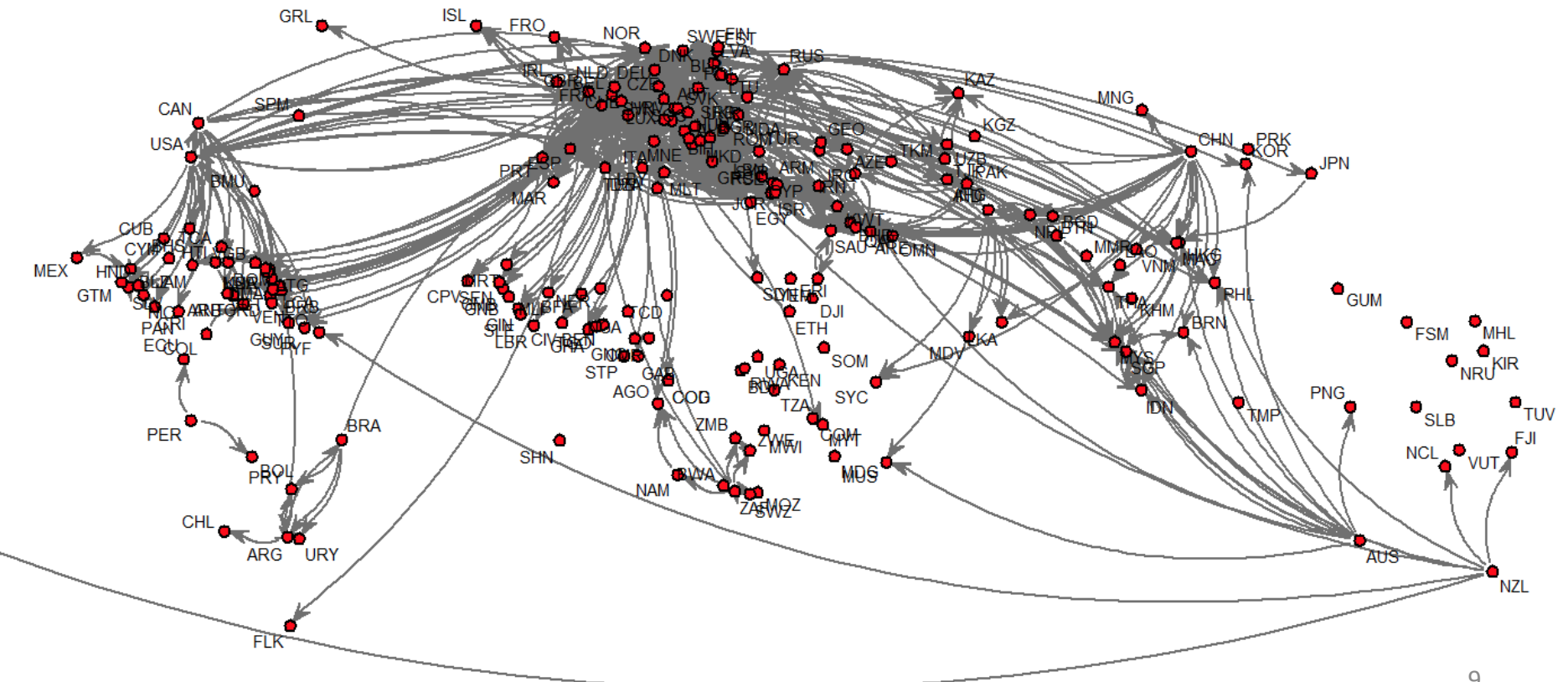


Immigration
flows

Graphs

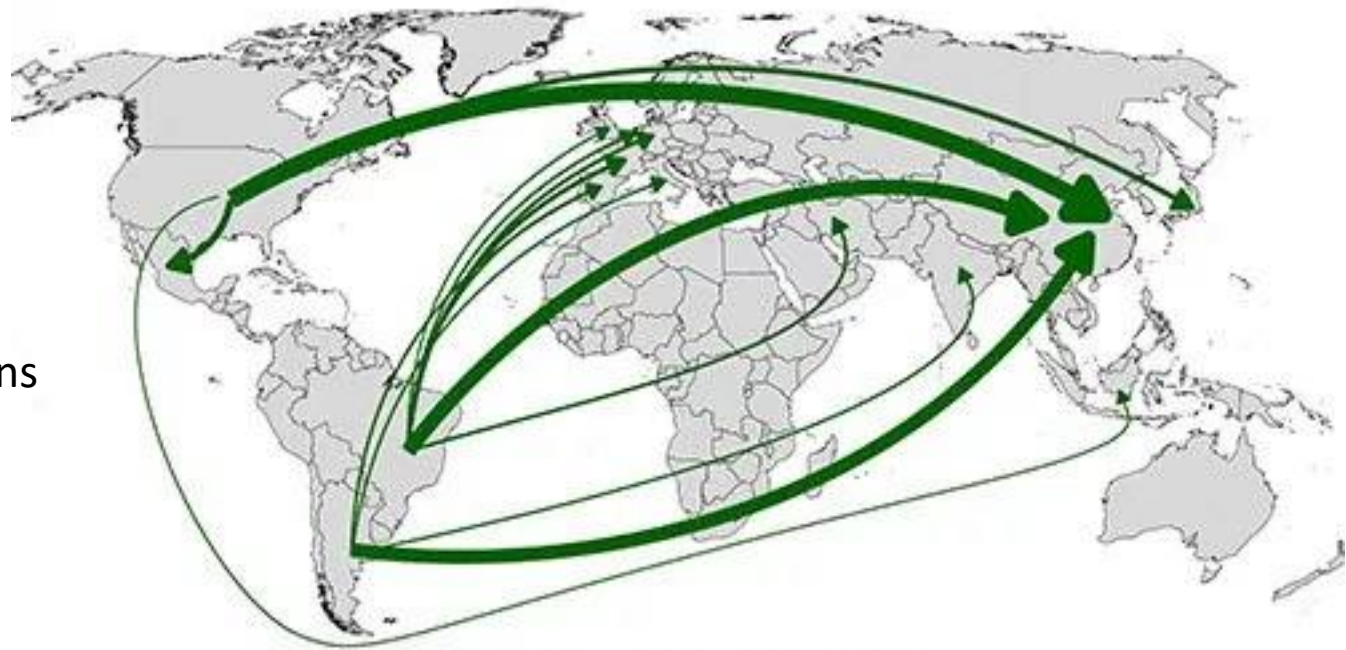
Potato trade

World trade in fresh potatoes, flows over 0.1 m US\$ average 2005-2009

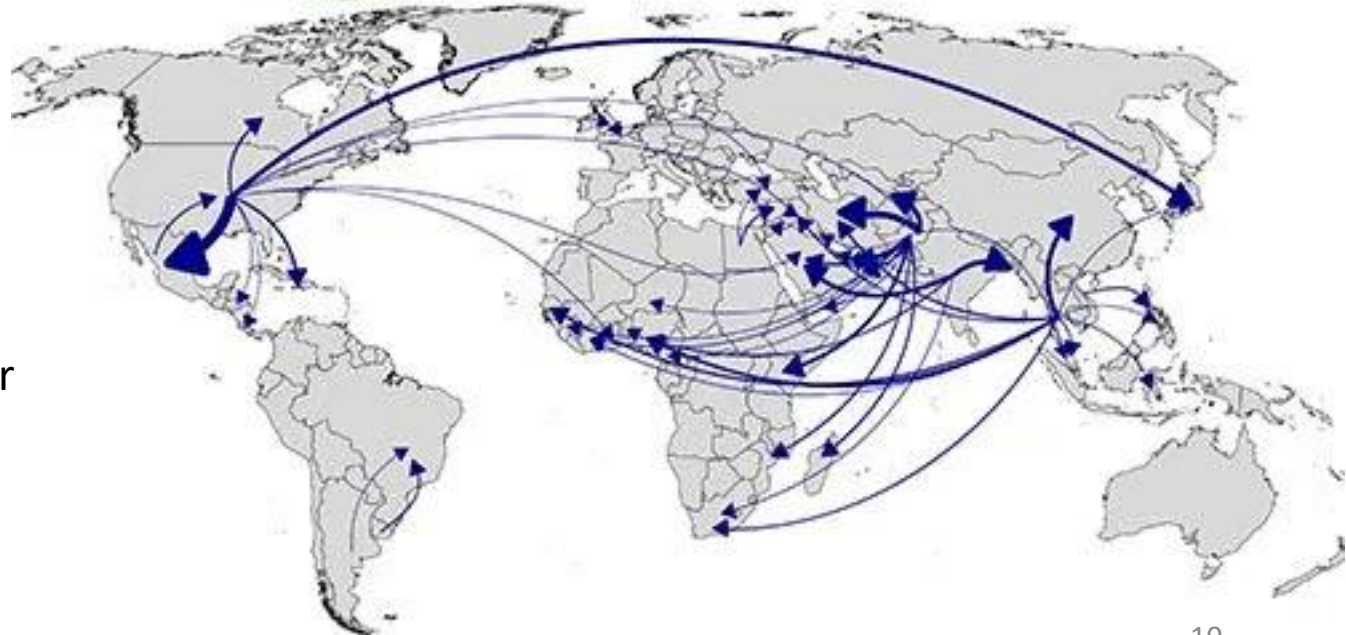


Graphs

Soybeans

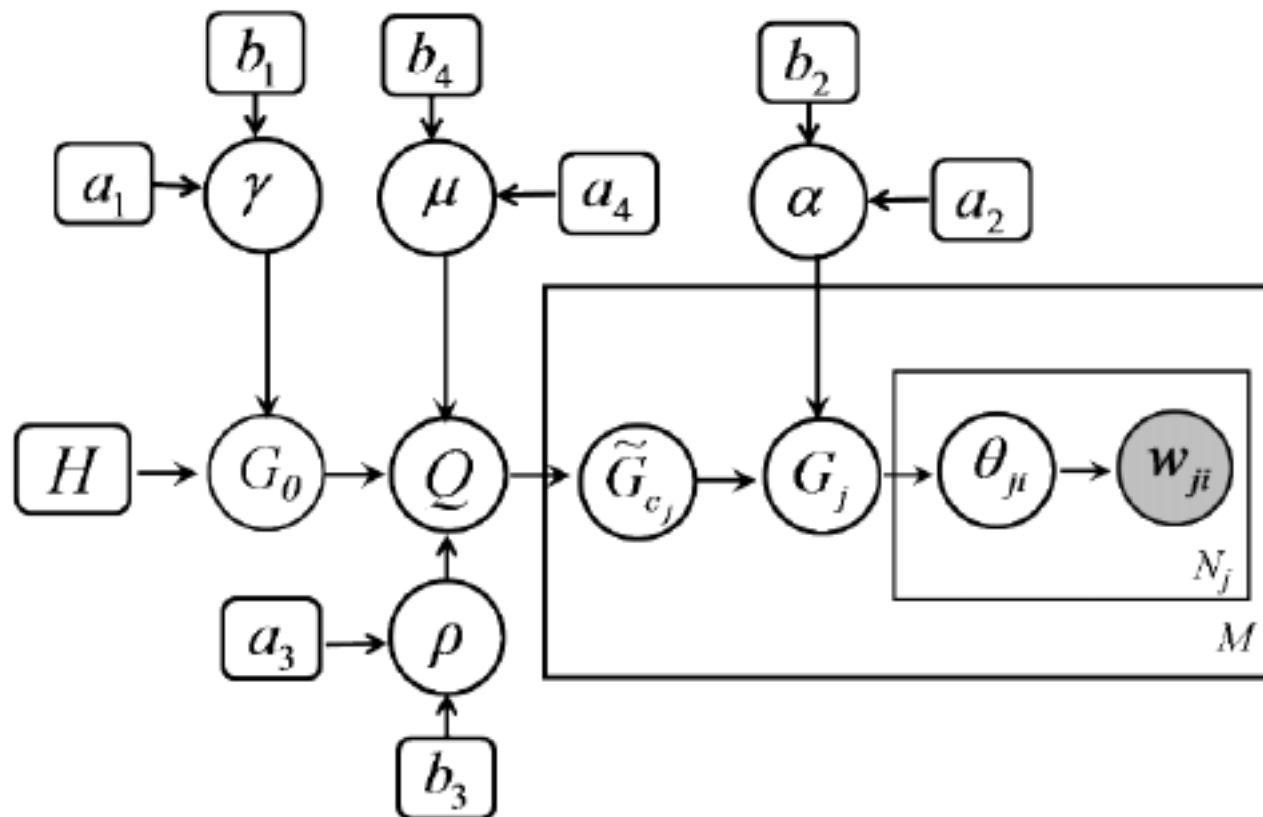


Water



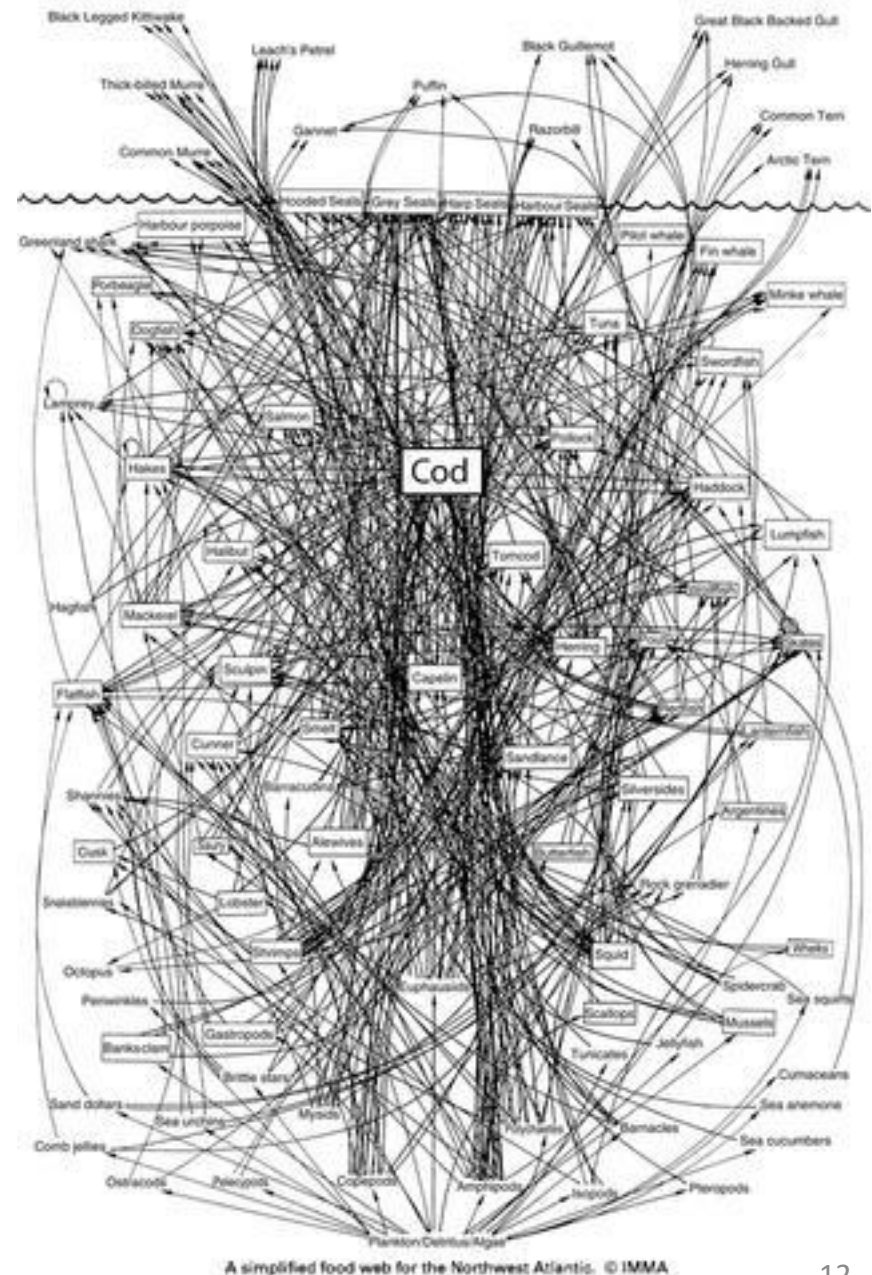
Graphs

Graphical models



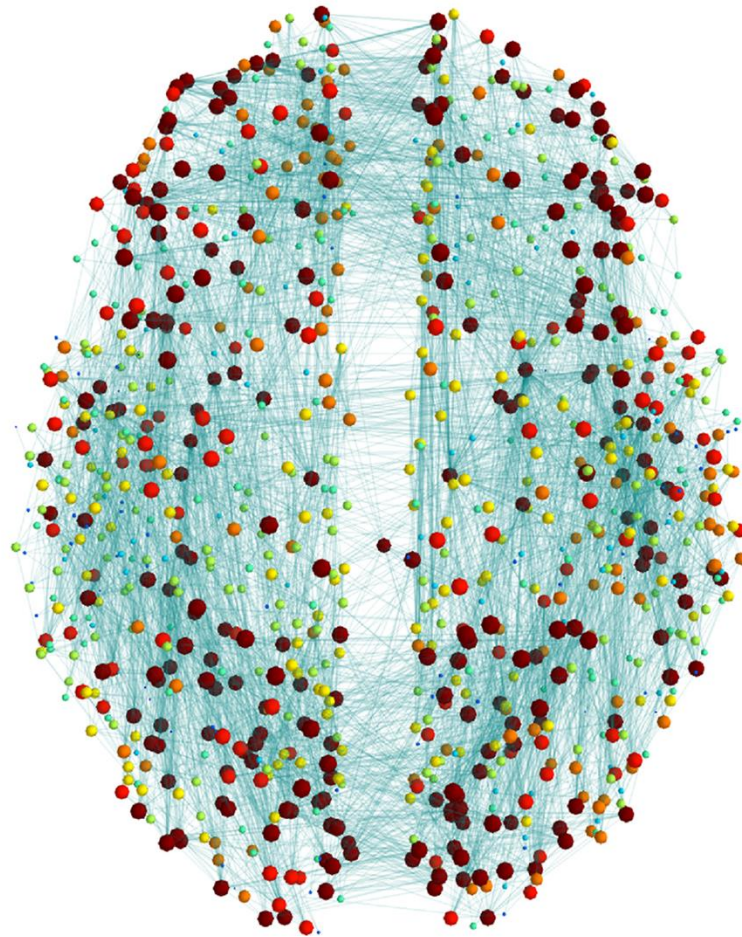
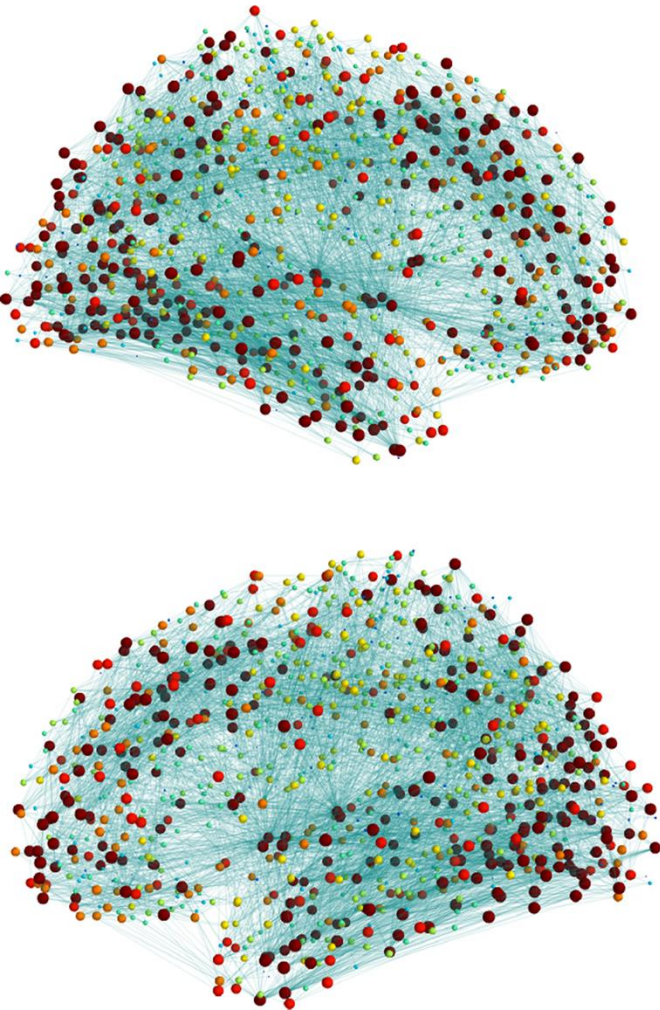
Graphs

What eats what in
the Atlantic ocean?



Graphs

Neural connections
in the brain

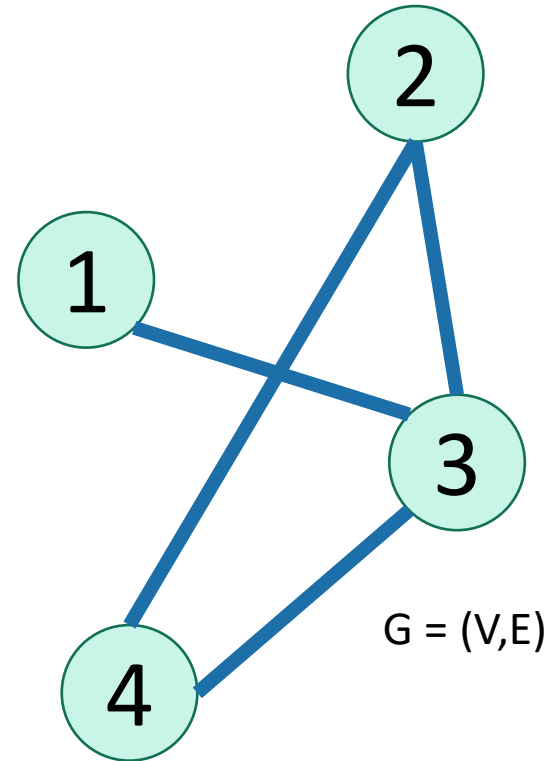


Graphs

- **There are a lot of graphs.**
- We want to answer questions about them.
 - Efficient routing?
 - Community detection/clustering?
 - 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.

Undirected Graphs

- Has vertices and edges
 - V is the set of vertices
 - E is the set of edges
 - Formally, a graph is $G = (V, E)$
- Example
 - $V = \{1, 2, 3, 4\}$
 - $E = \{ \{1, 3\}, \{2, 4\}, \{3, 4\}, \{2, 3\} \}$



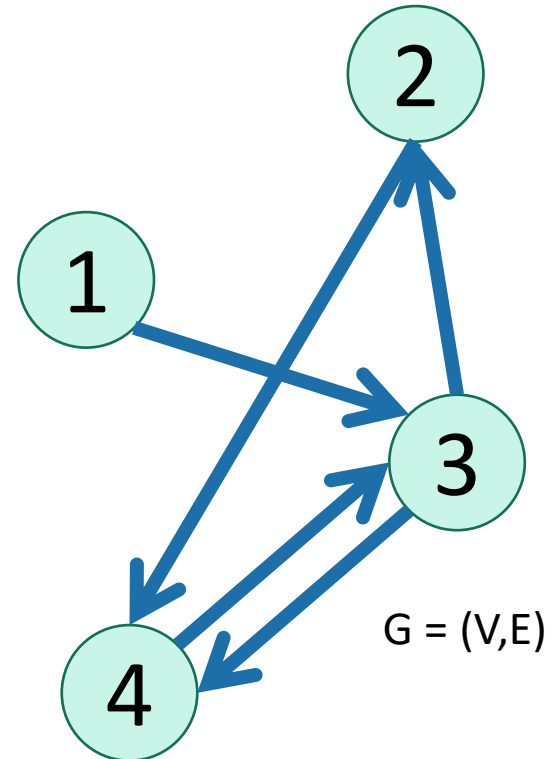
- The **degree** of vertex 4 is 2.
 - There are 2 edges coming out.
- Vertex 4's **neighbors** are 2 and 3

Directed Graphs

- Has vertices and edges
 - V is the set of vertices
 - E is the set of **DIRECTED** edges
 - Formally, a graph is $G = (V, E)$

- Example

- $V = \{1, 2, 3, 4\}$
- $E = \{ (1, 3), (2, 4), (3, 4), (4, 3), (3, 2) \}$

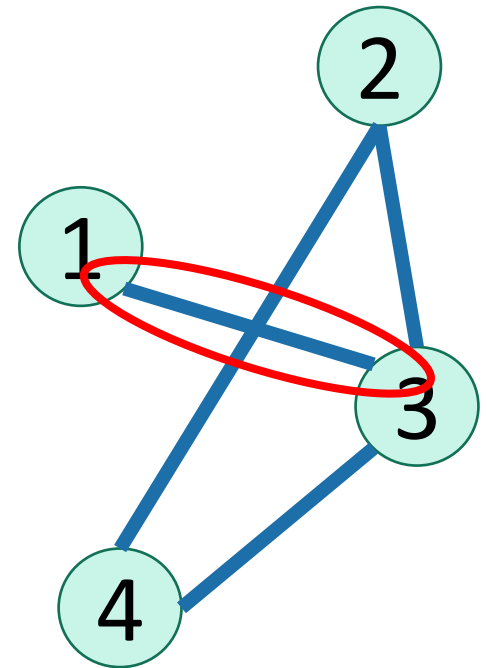


- The **in-degree** of vertex 4 is 2.
- The **out-degree** of vertex 4 is 1.
- Vertex 4's **incoming neighbors** are 2, 3
- Vertex 4's **outgoing neighbor** is 3.

How do we represent graphs?

- Option 1: adjacency matrix

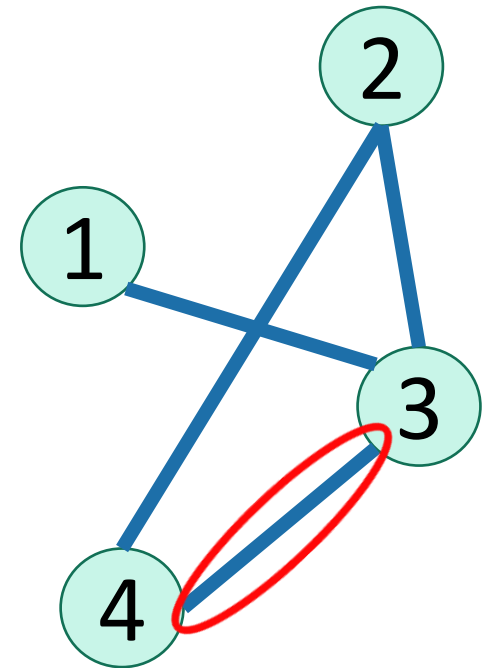
	1	2	3	4
1	0	0	1	0
2	0	0	1	1
3	1	1	0	1
4	0	1	1	0



How do we represent graphs?

- Option 1: adjacency matrix

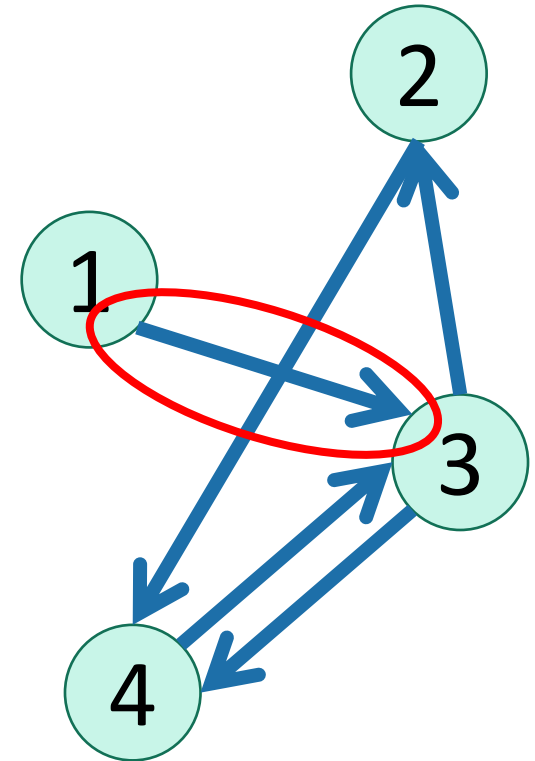
	1	2	3	4
1	0	0	1	0
2	0	0	1	1
3	1	1	0	1
4	0	1	1	0



How do we represent graphs?

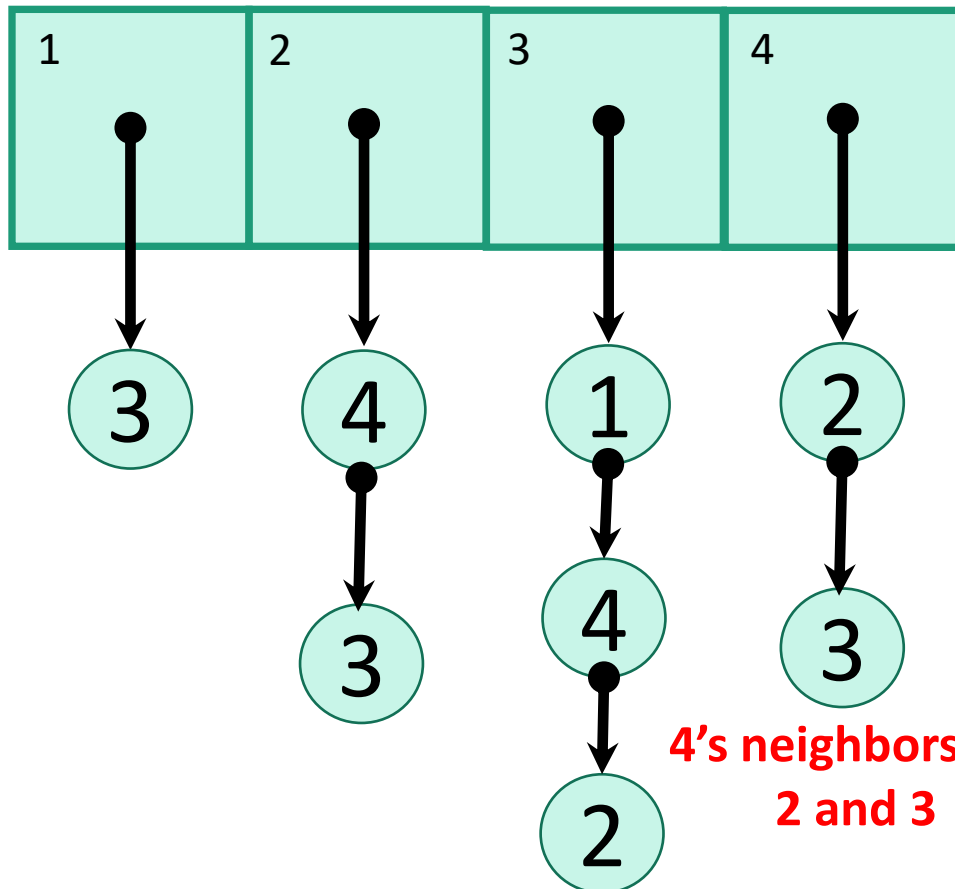
- Option 1: adjacency matrix

		Destination			
		1	2	3	4
Source	1	0	0	1	0
	2	0	0	0	1
	3	0	1	0	1
	4	0	0	1	0

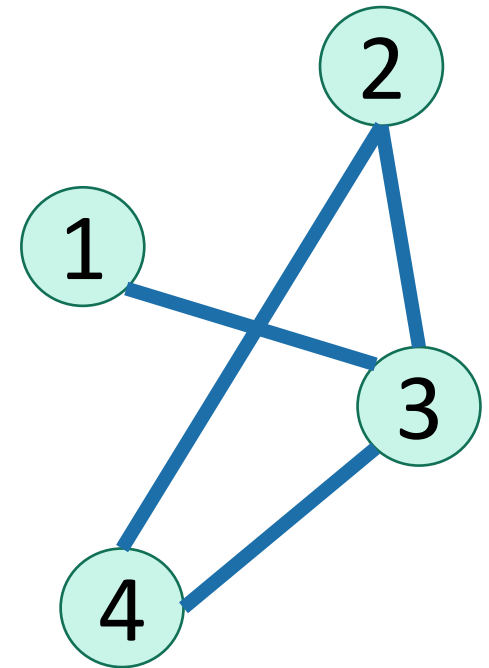


How do we represent graphs?

- Option 2: adjacency lists.



4's neighbors are
2 and 3



How would you
modify this for
directed graphs?



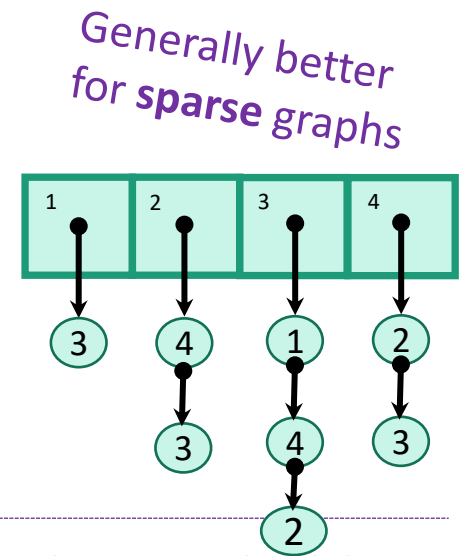
In either case

- Vertices can store other information
 - Attributes (name, IP address, ...)
 - 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

Say there are n vertices
and m 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
Give me v 's neighbors.

$O(n)$

$O(\deg(v))$

Space requirements

$O(n^2)$

$O(n + m)$

We'll assume this
representation for
the rest of the class

Acknowledgement

- Stanford University