Indian Institute of Information Technology Allahabad

Data Structures

Graphs

Dr. Shiv Ram Dubey
Assistant Professor
Department of Information Technology
Indian Institute of Information Technology, Allahabad

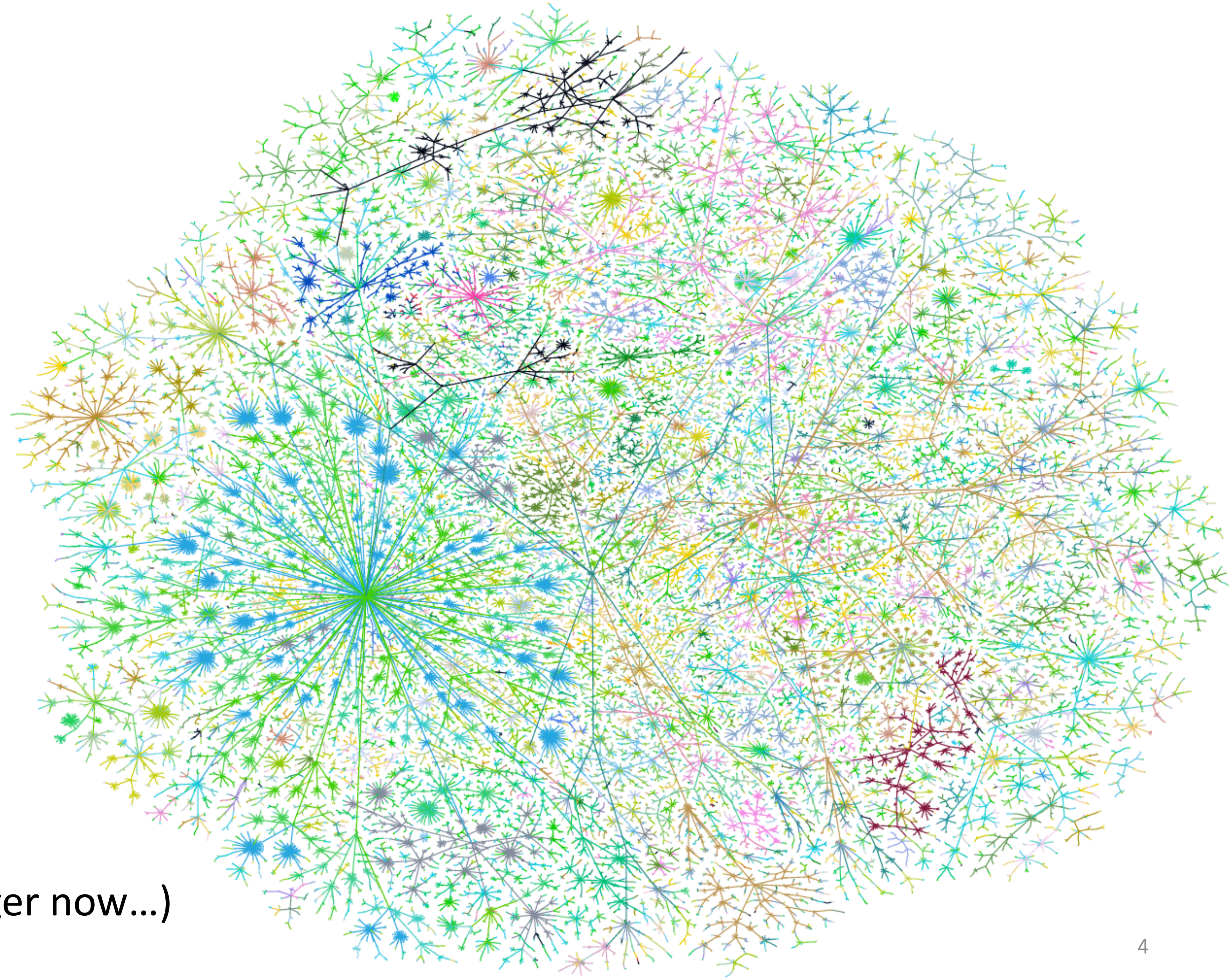
Email: srdubey@iiita.ac.in Web: https://profile.iiita.ac.in/srdubey/

DISCLAIMER

The content (text, image, and graphics) used in this slide are adopted from many sources for academic purposes. Broadly, the sources have been given due credit appropriately. However, there is a chance of missing out some original primary sources. The authors of this material do not claim any copyright of such material.

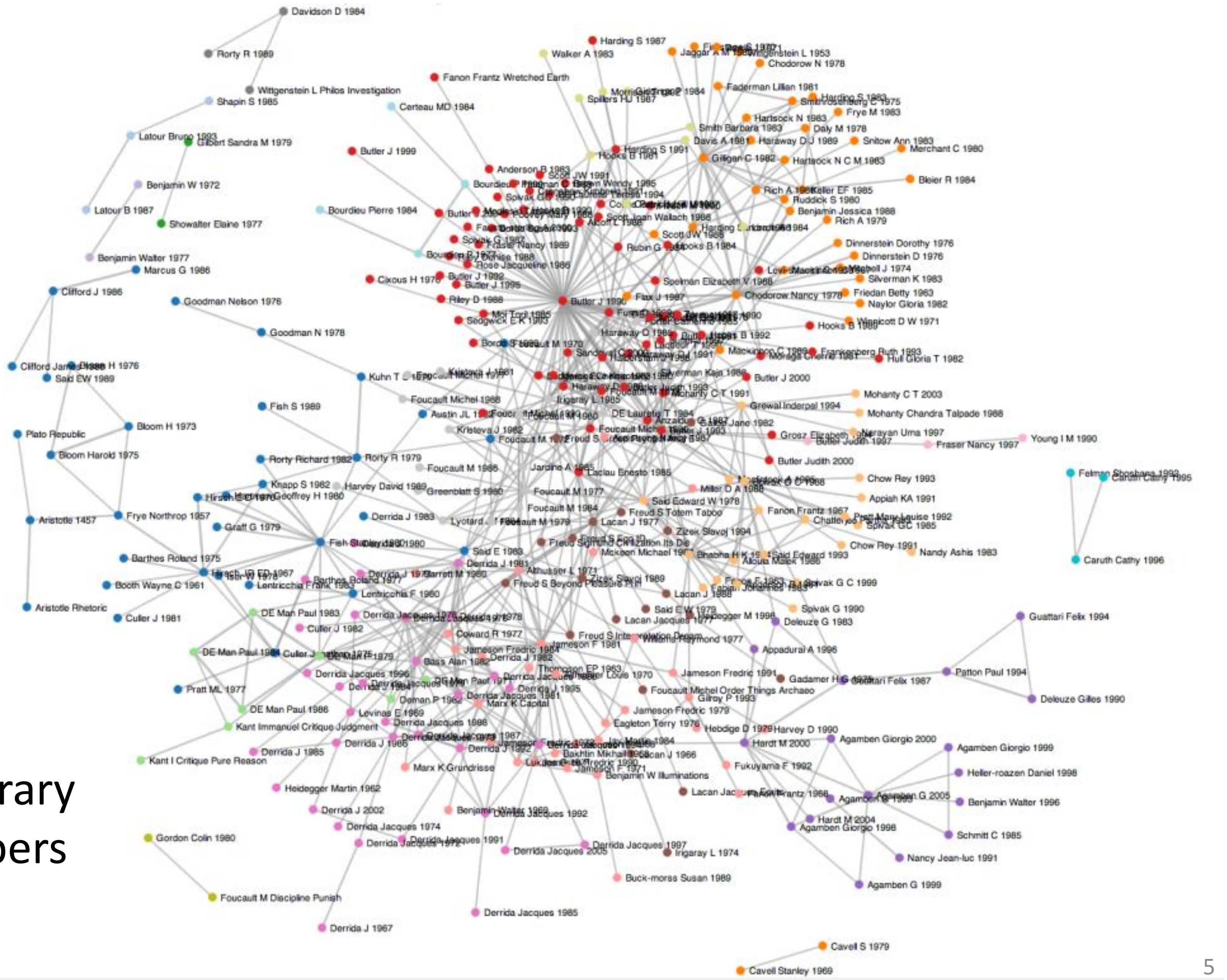
Graphs

Graphs



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

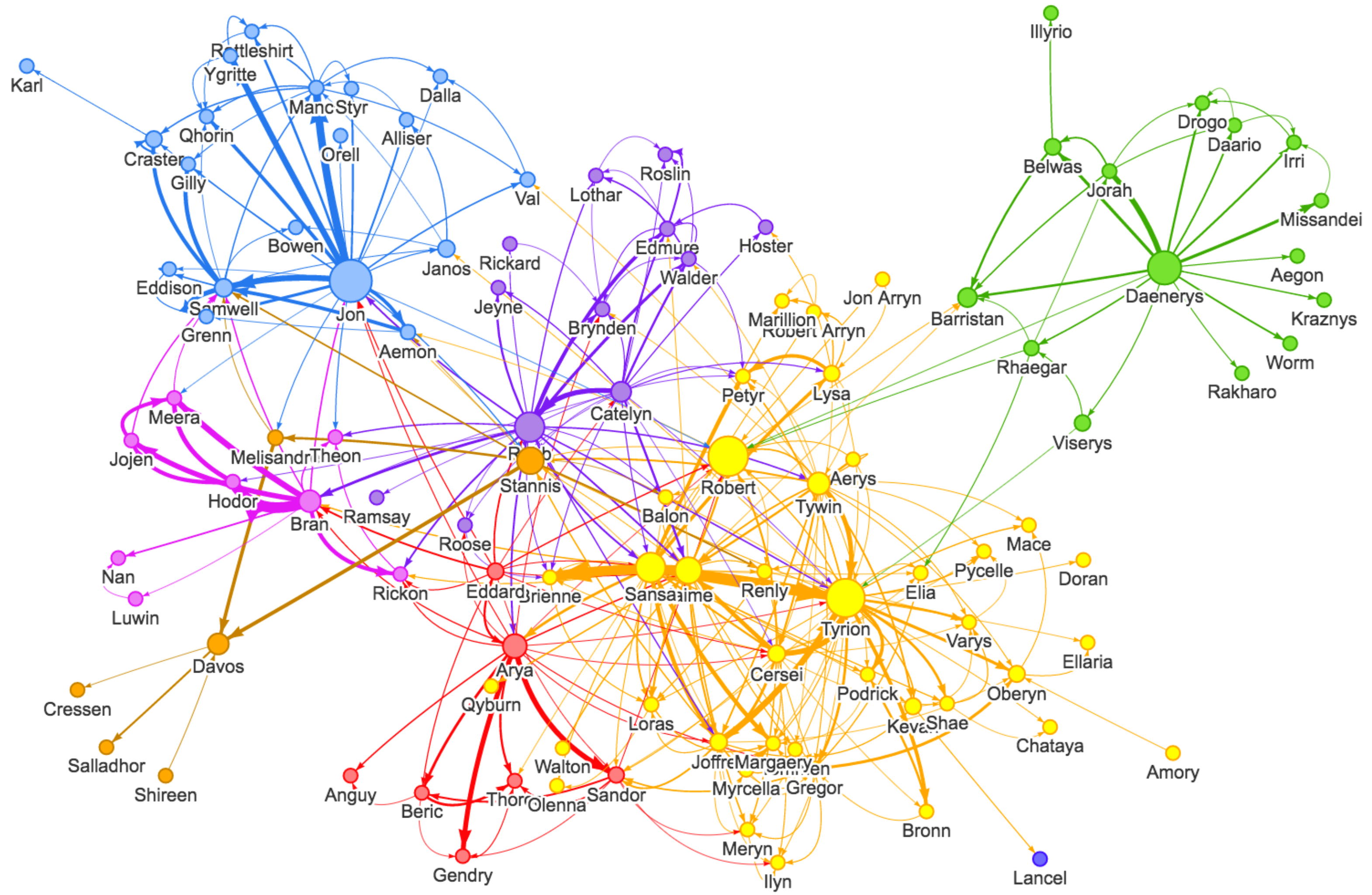
Graphs



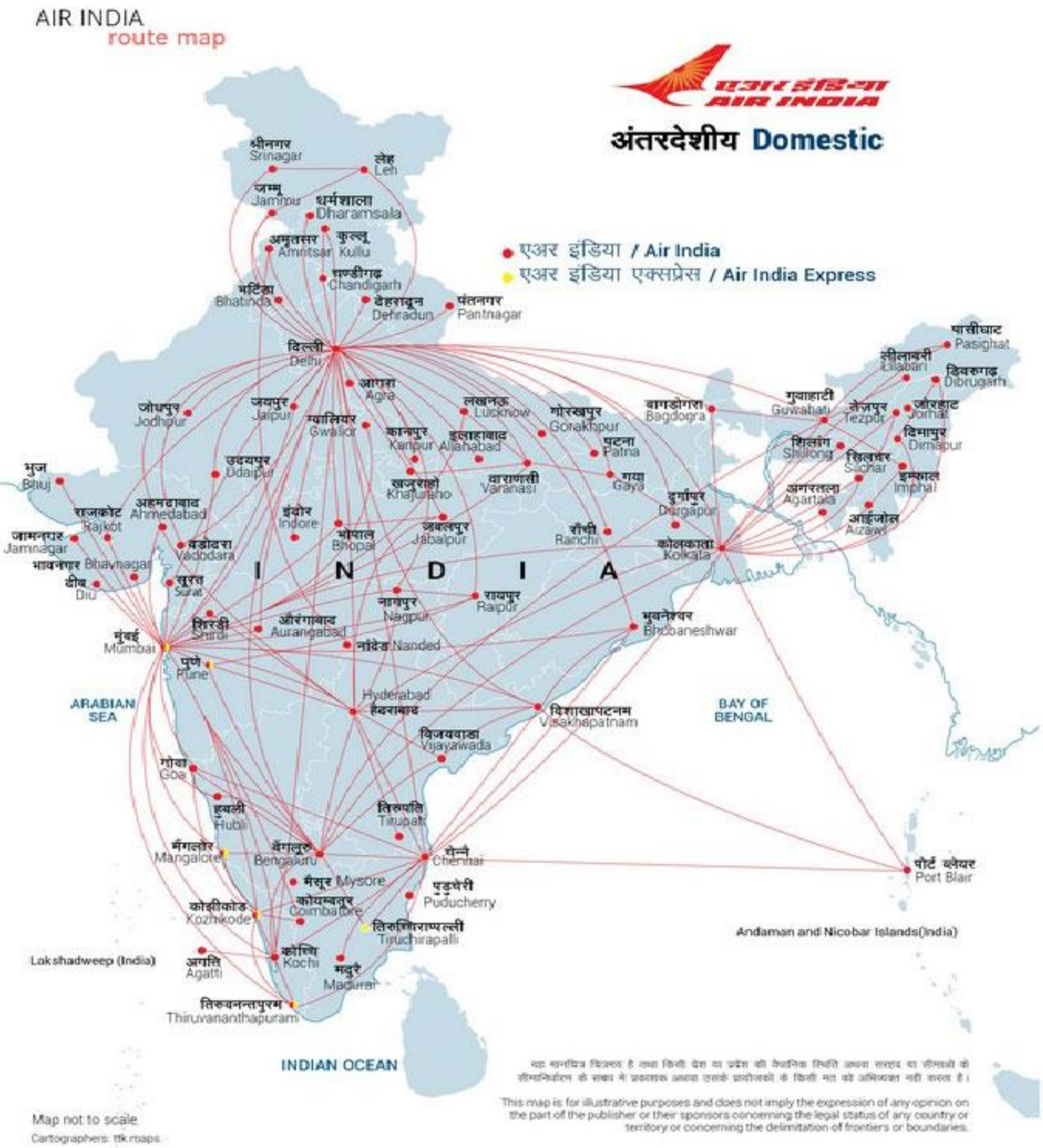
Citation graph of literary theory academic papers

Graphs

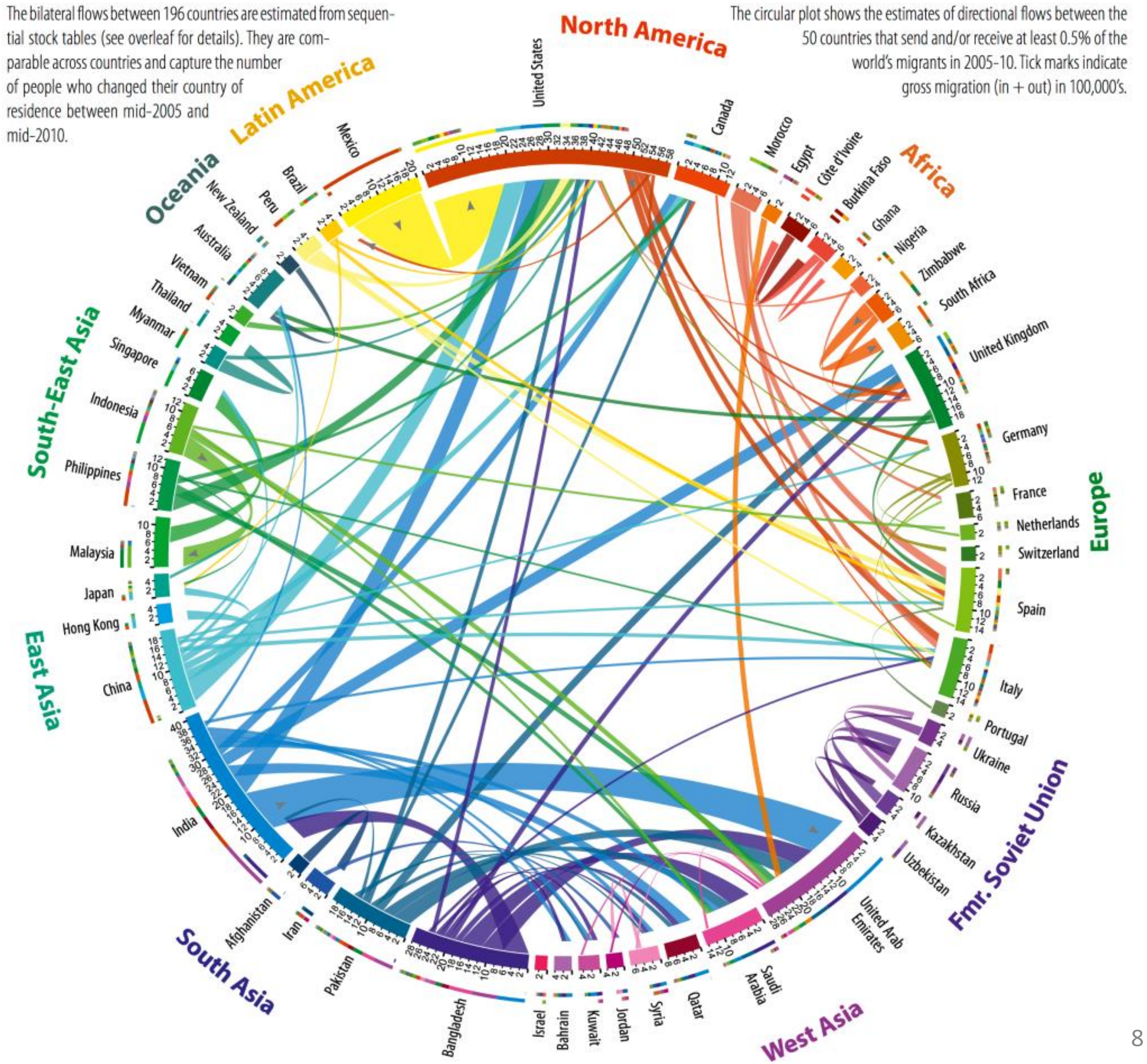
Game of Thrones
Character Interaction Network



Graphs


AIR INDIA
route map
एअर इंडिया
AIR INDIA
अंतरदेशीय Domestic
एअर इंडिया / Air India
एअर इंडिया एक्सप्रेस / Air India Express
Srinagar
Leh
Jammu
Dharamsala
Amritsar
Kullu
Chandigarh
Bhatinda
Dehradun
Pantnagar
Delhi
Agra
Jodhpur
Jaipur
Gwalior
Lucknow
Gorakhpur
Bagdogra
Guwahati
Tezpur
Jorhat
Lilabari
Pasighat
Dibrugarh
Dimapur
Shillong
Silchar
Imphal
Agartala
Aizawl
Kanpur
Allahabad
Patna
Varanasi
Gaya
Khajuraho
Durgapur
Bhuj
Udaipur
Rajkot
Ahmedabad
Jamnagar
Indore
Bhopal
Jabalpur
Ranchi
Kolkata
Vadodara
Bhavnagar
Diu
Surat
I N D I A
Raipur
Nagpur
Shirdi
Aurangabad
Bhubaneshwar
Mumbai
Nanded
Pune
Hyderabad
Visakhapatnam
ARABIAN SEA
BAY OF BENGAL
Vijayawada
Goa
Hubli
Tirupati
Mangalore
Bengaluru
Chennai
Port Blair
Mysore
Puducherry
Kozhikode
Coimbatore
Tiruchirapalli
Andaman and Nicobar Islands(India)
Lakshadweep (India)
Agatti
Kochi
Madurai
Thiruvananthapuram
INDIAN OCEAN
This map is for illustrative purposes and does not imply the expression of any opinion on the part of the publisher or their sponsors concerning the legal status of any country or territory or concerning the delimitation of frontiers or boundaries.
Map not to scale
Cartographers: ttk maps

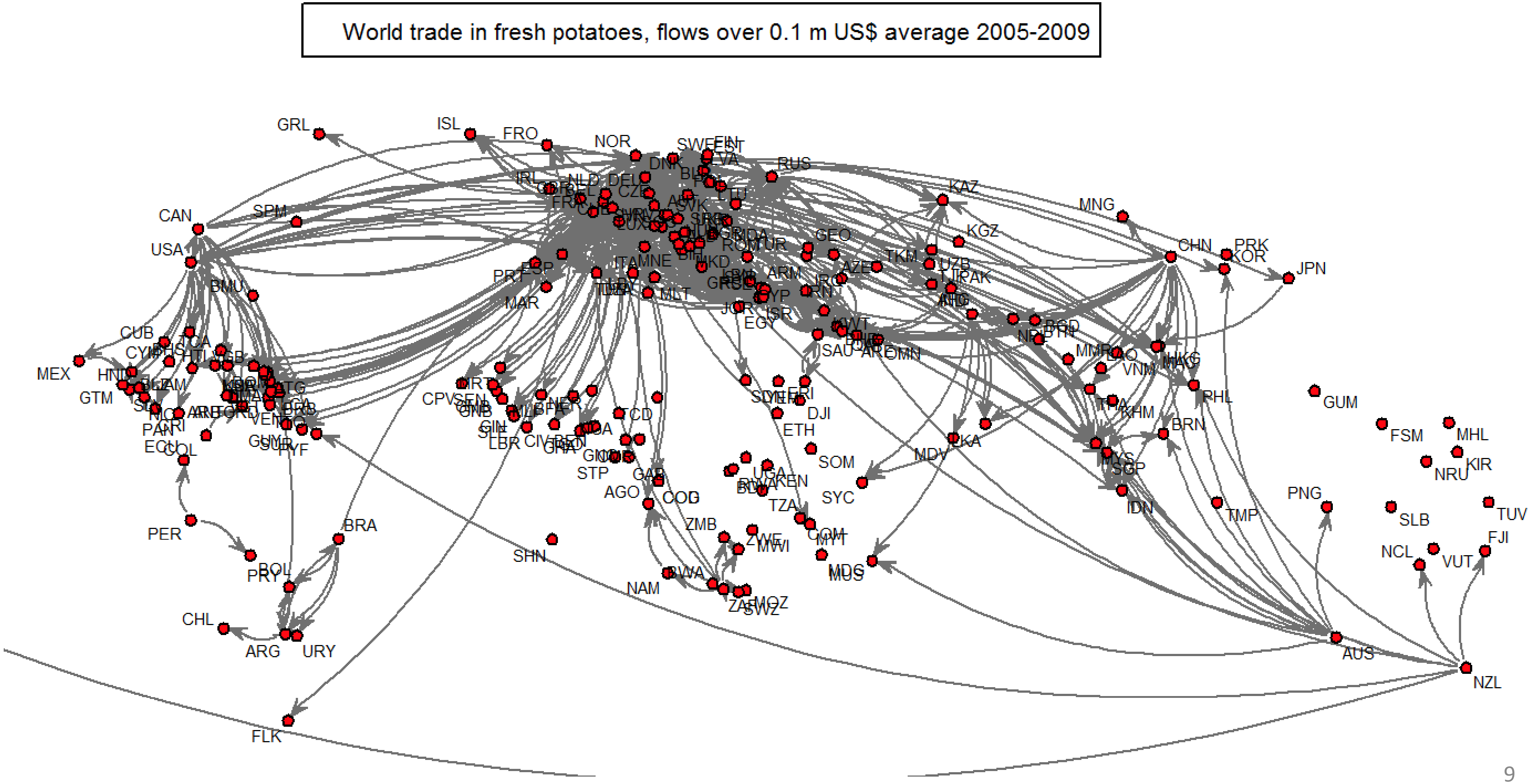
Graphs



Immigration flows

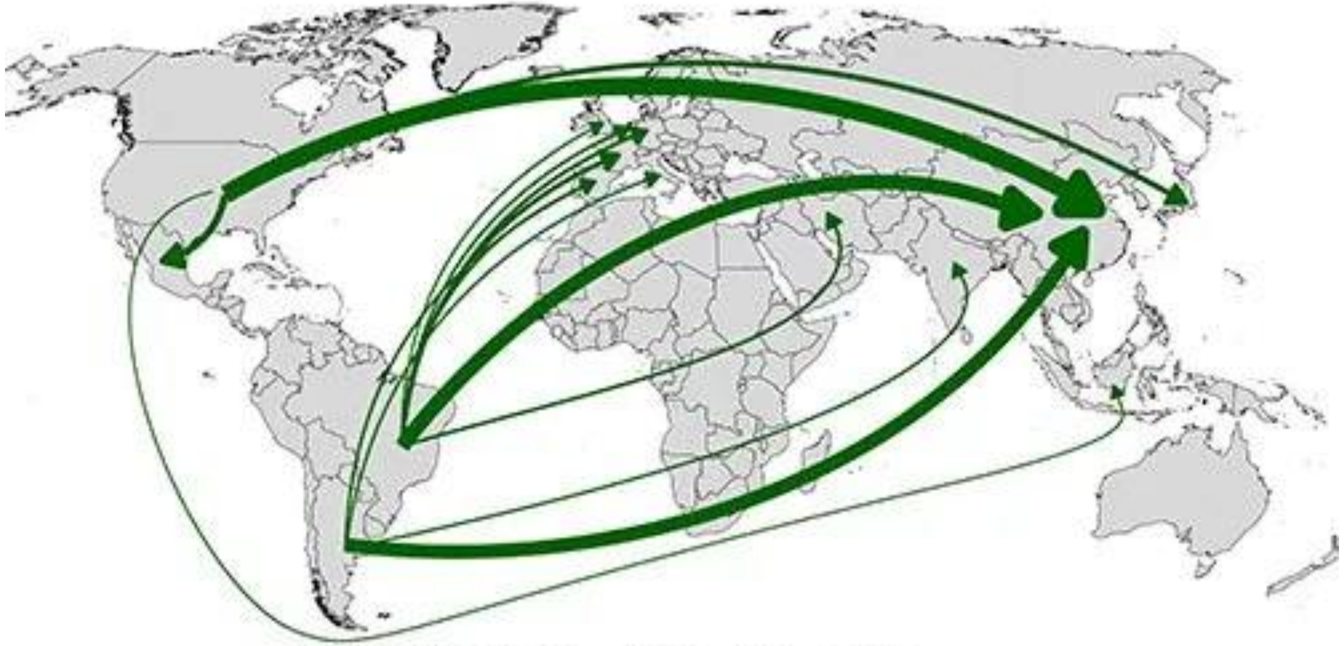
Graphs

Potato trade

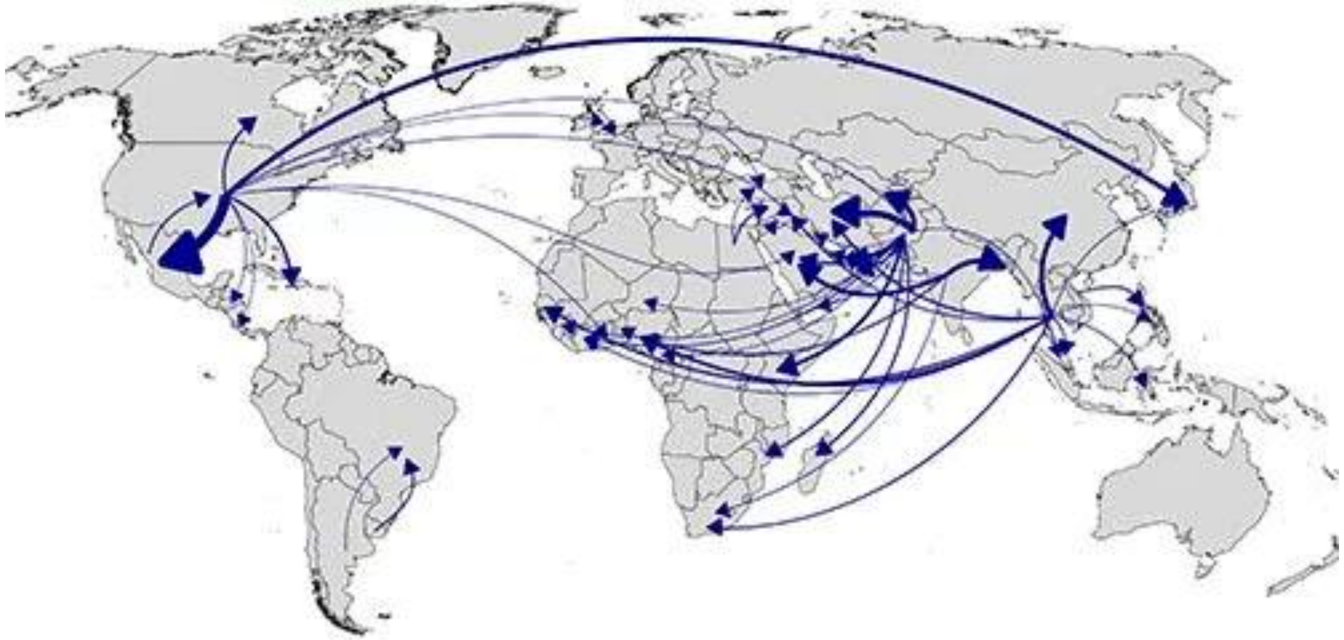


Graphs

Soybeans

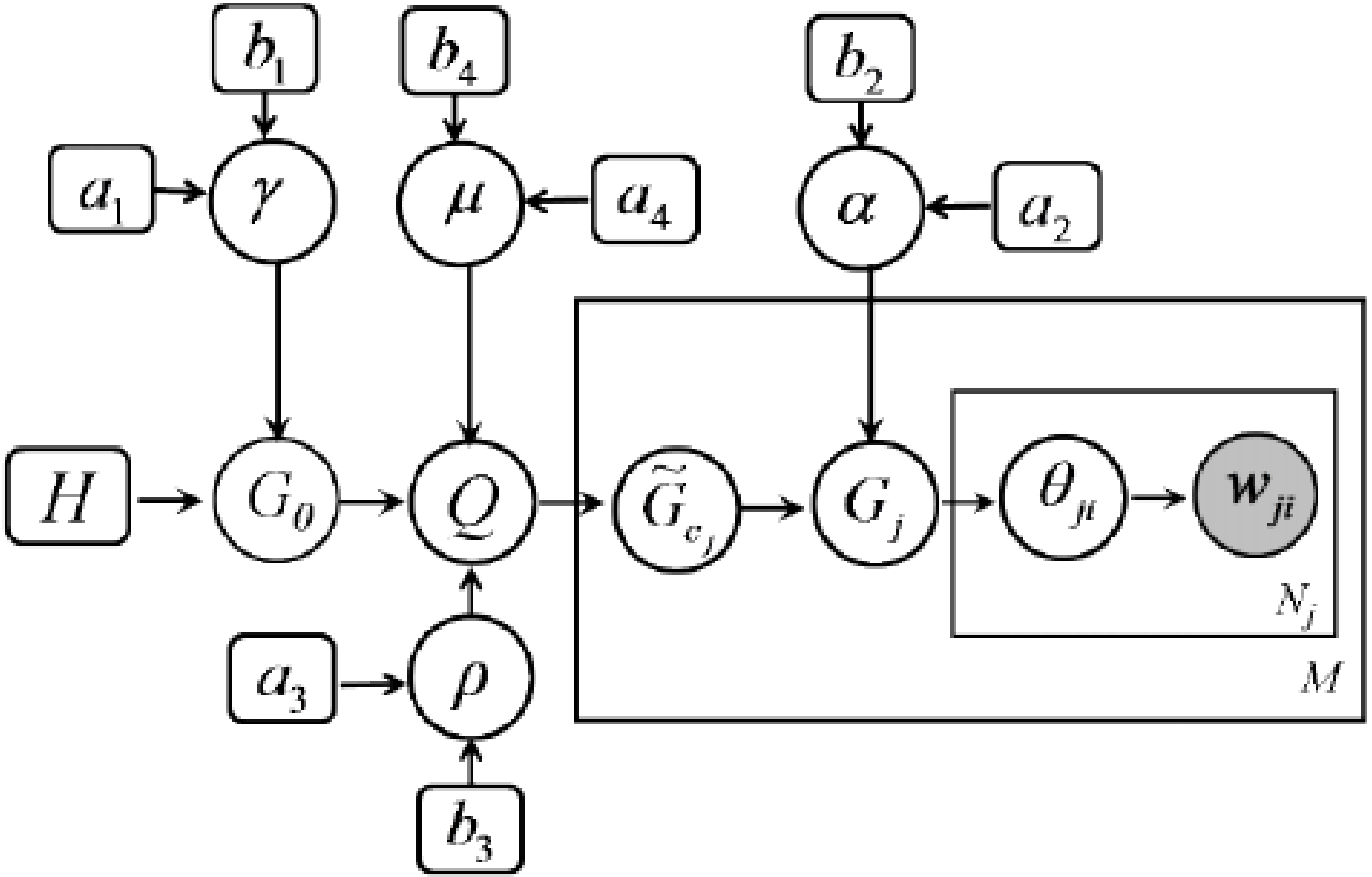


Water



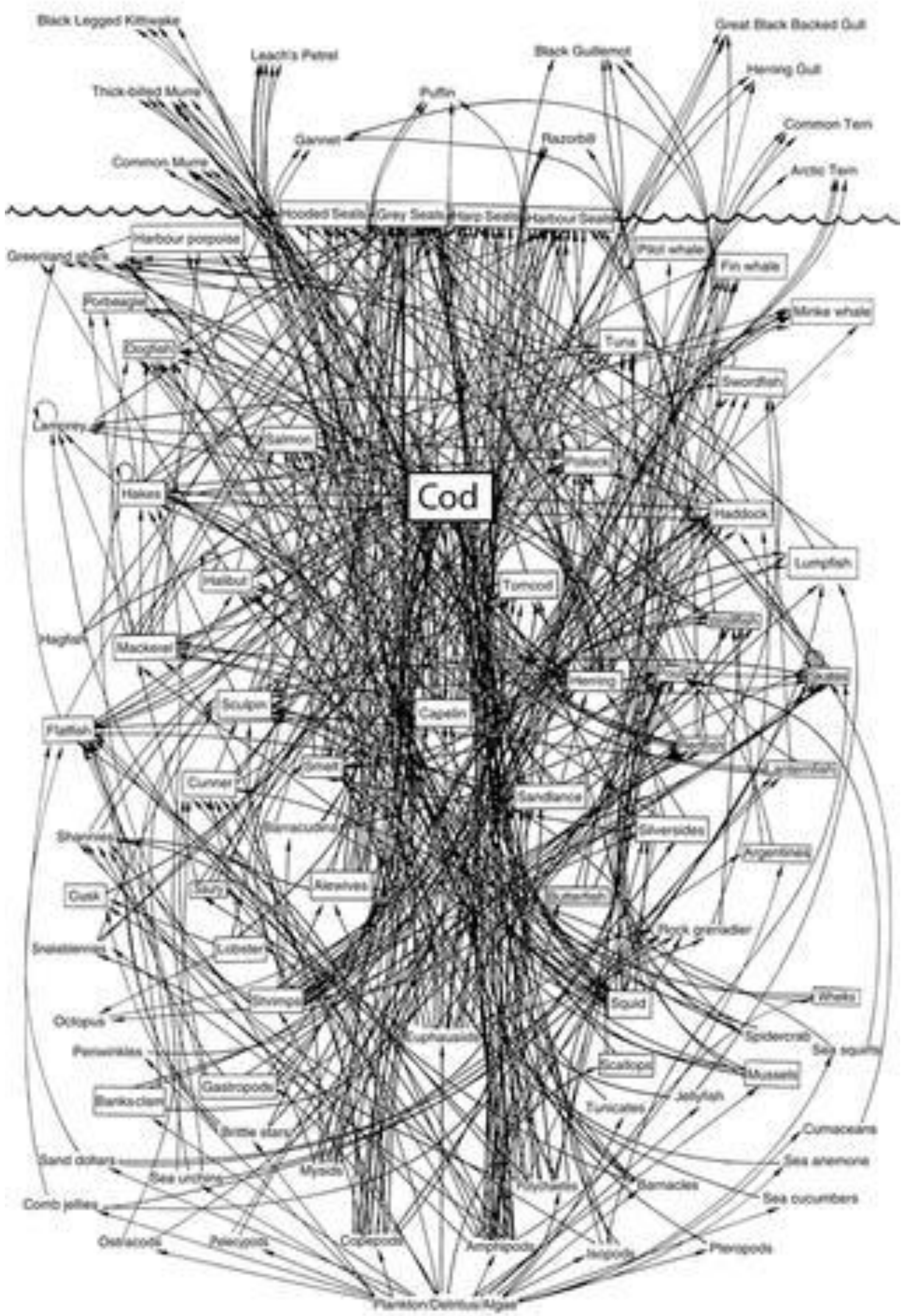
Graphs

Graphical models



Graphs

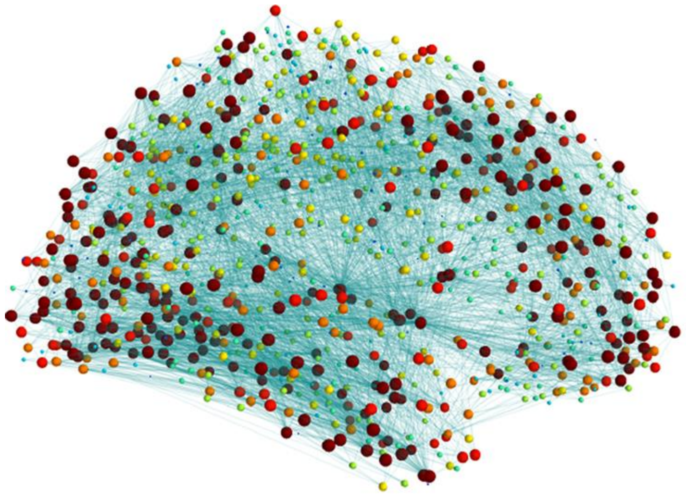
What eats what in the Atlantic ocean?

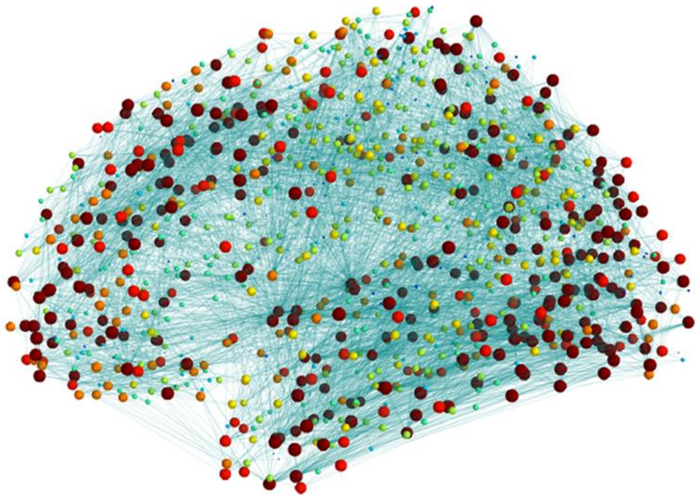


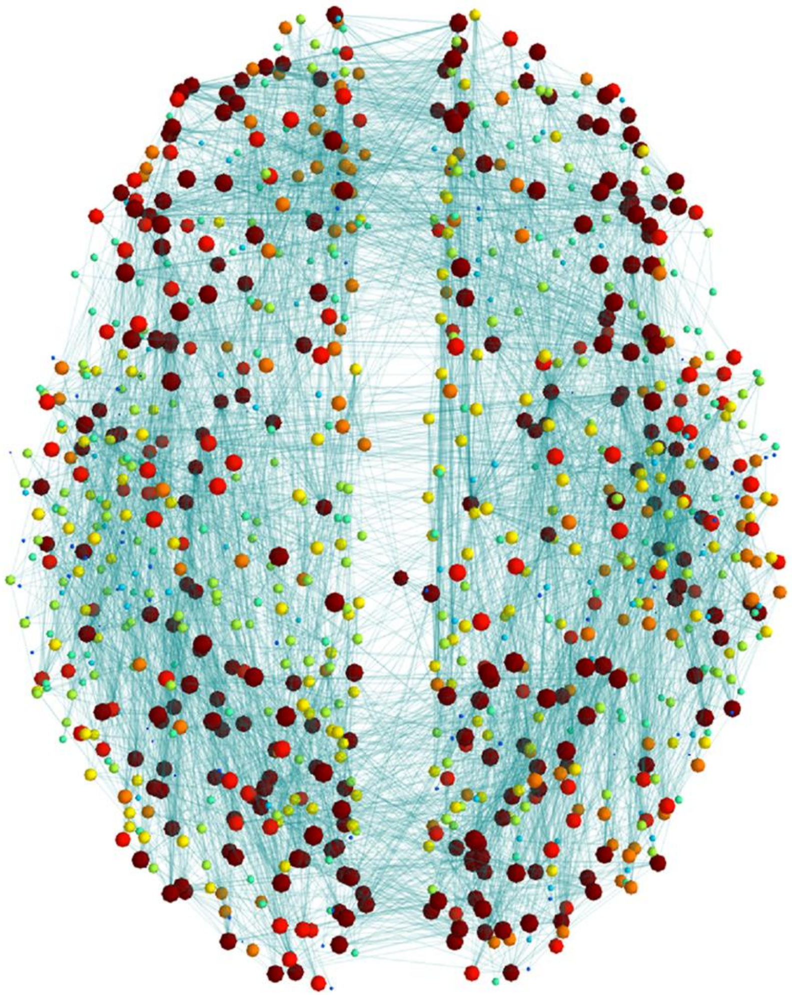
A simplified food web for the Northwest Atlantic. © IMMA

Graphs

Neural connections in the brain





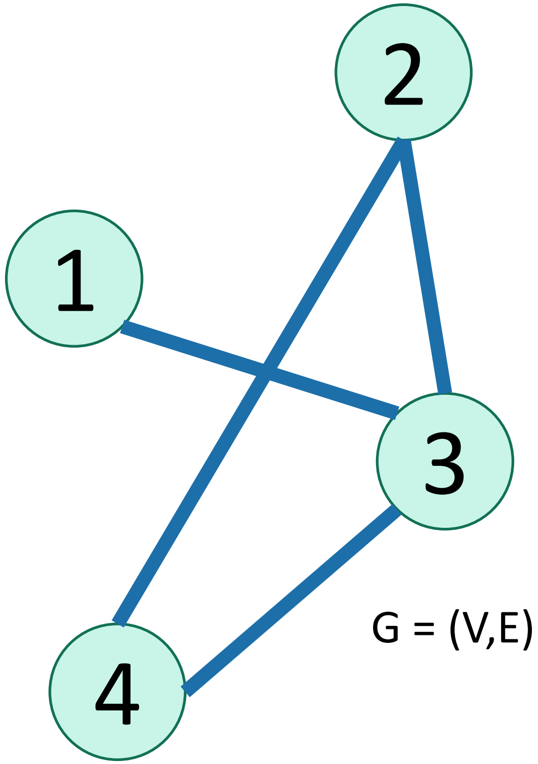


Graphs

- **There are a lot of graphs.**

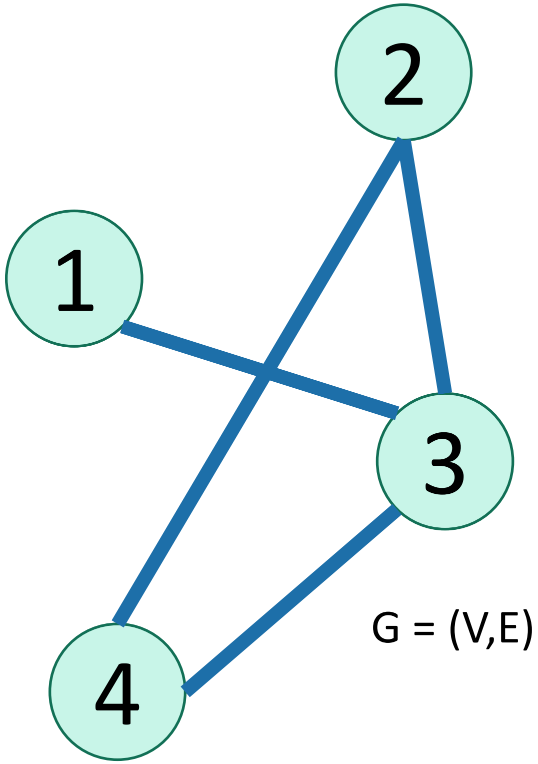
- We want to answer questions about them.
    - Efficient routing?
    - Community detection/clustering?
    - Signing up for classes without violating pre-req constraints
    - How to distribute fish in tanks so that none of them will fight.

Undirected Graphs



Undirected Graphs

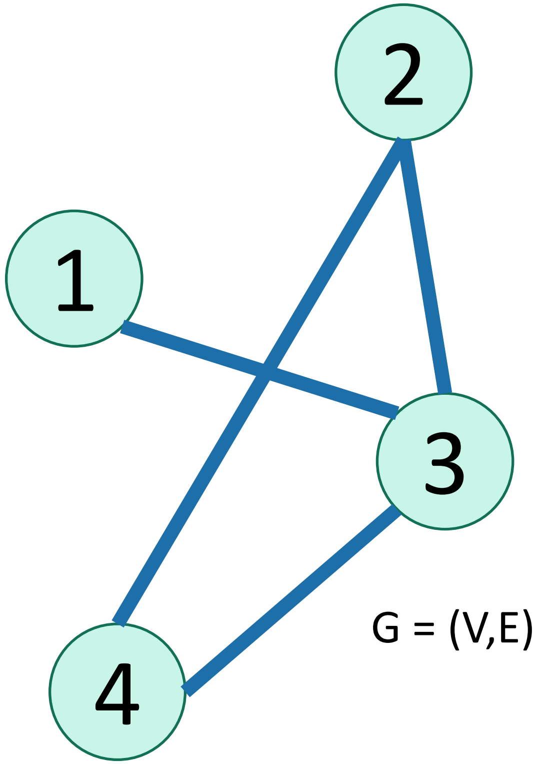
- Has vertices and edges
  - V is the set of vertices
  - E is the set of edges
  - Formally, a graph is G = (V,E)



Undirected Graphs

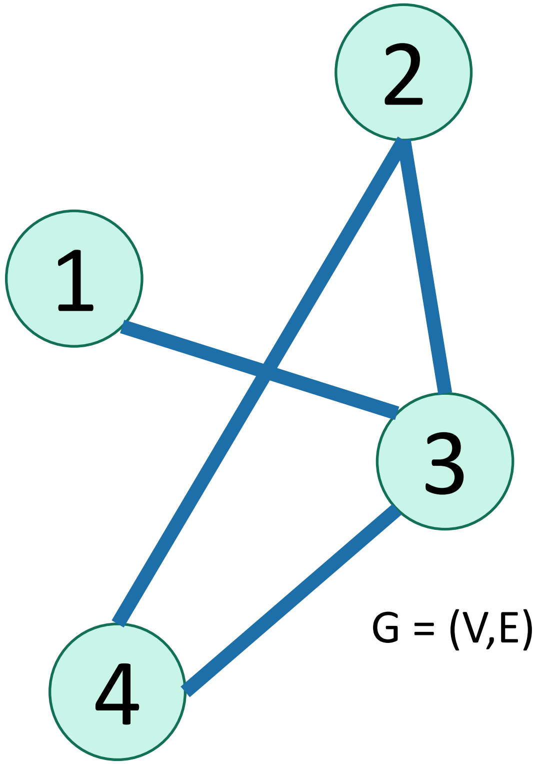
- Has vertices and edges
  - V is the set of vertices
  - E is the set of edges
  - Formally, a graph is G = (V,E)

- Example
  - V = {1,2,3,4}
  - E = { {1,3}, {2,4}, {3,4}, {2,3} }



Undirected Graphs

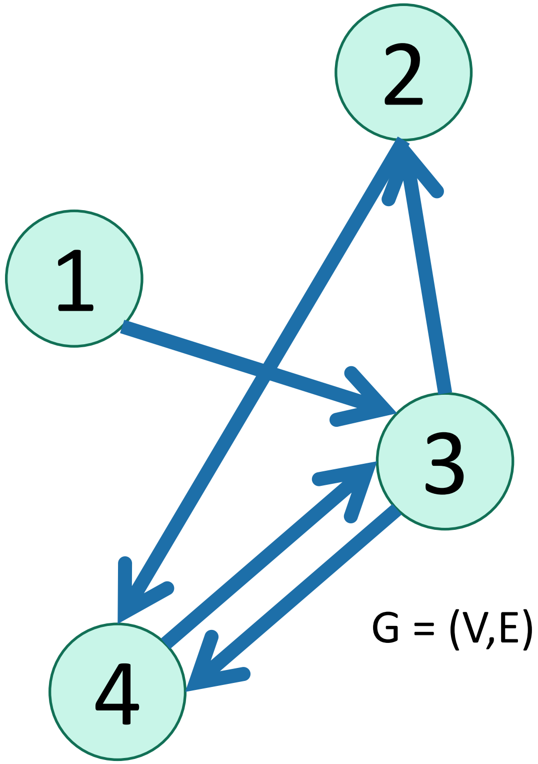
- Has vertices and edges
  - V is the set of vertices
  - E is the set of edges
  - Formally, a graph is G = (V,E)



- Example
  - V = {1,2,3,4}
  - E = { {1,3}, {2,4}, {3,4}, {2,3} }

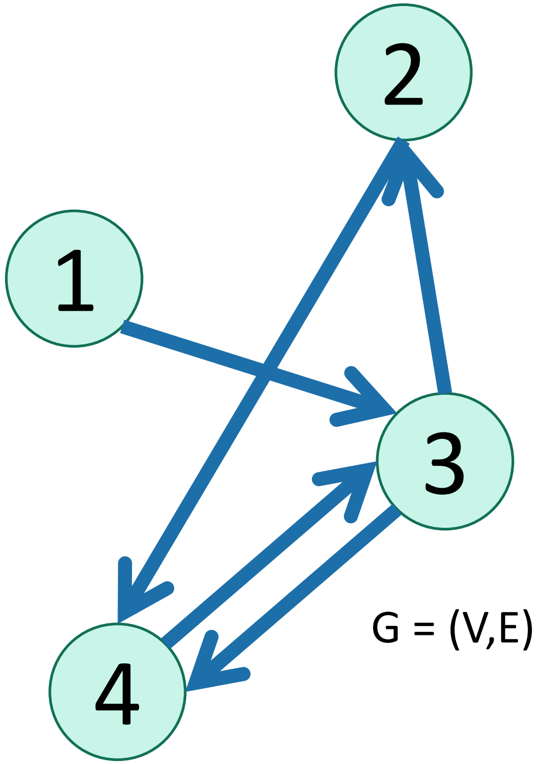
- The **degree** of vertex 4 is 2.
  - There are 2 edges coming out.
- Vertex 4's **neighbors** are 2 and 3

Directed Graphs



Directed Graphs

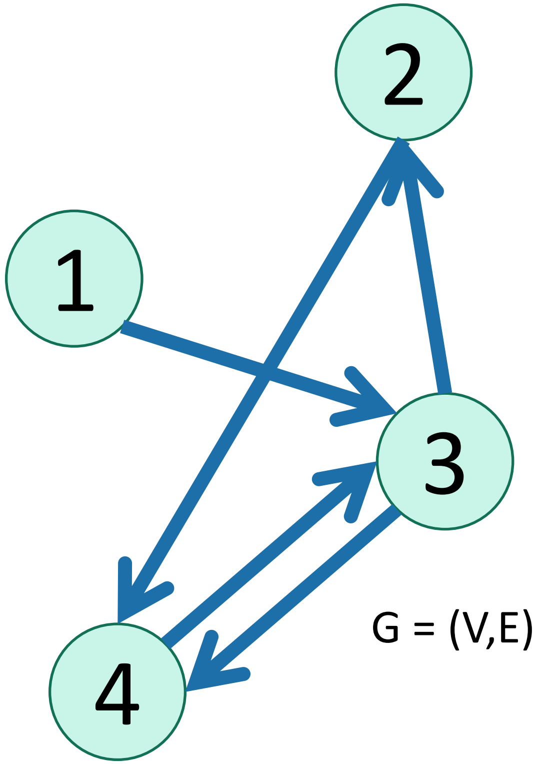
- Has vertices and edges
  - V is the set of vertices
  - E is the set of **DIRECTED** edges
  - Formally, a graph is G = (V,E)



Directed Graphs

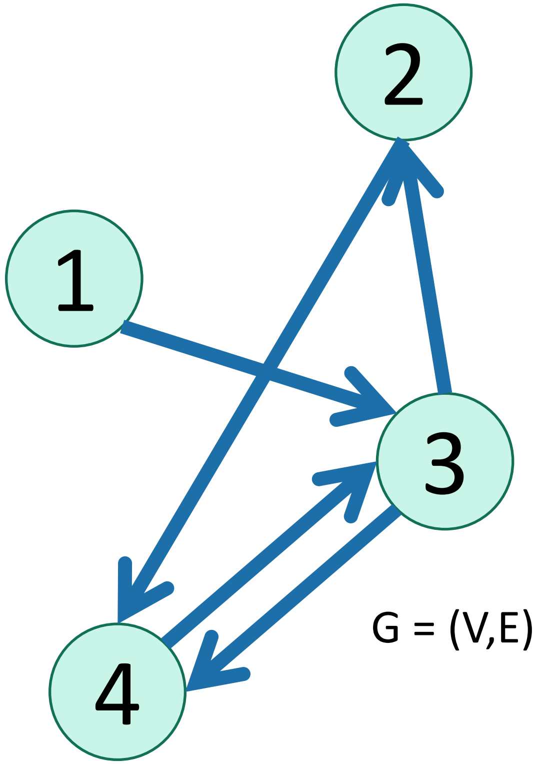
- Has vertices and edges
  - V is the set of vertices
  - E is the set of **DIRECTED** edges
  - Formally, a graph is G = (V,E)

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



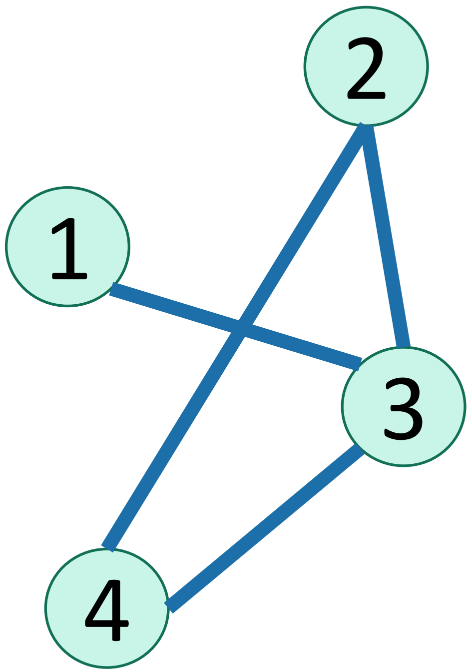
Directed Graphs

- 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) }



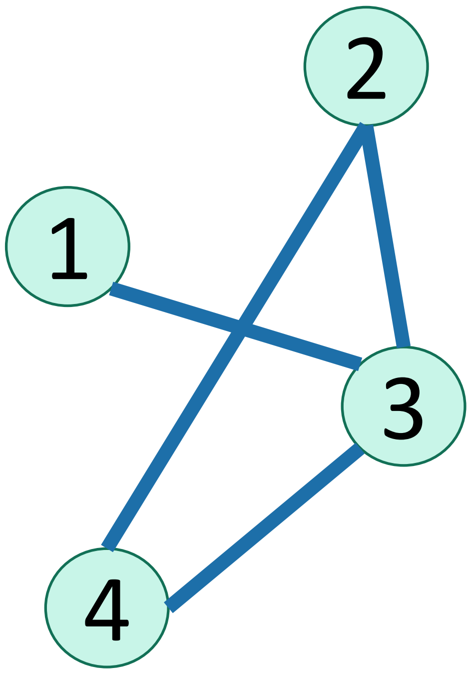
- The **in-degree** of vertex 4 is 2.
- The **out-degree** of vertex 4 is 1.
- Vertex 4's **incoming neighbors** are 2,3
- Vertex 4's **outgoing neighbor** is 3.

How do we represent graphs?



How do we represent graphs?

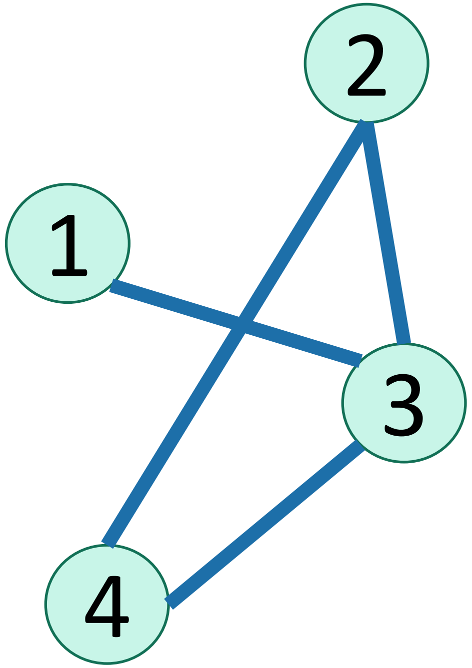
- Option 1: adjacency matrix



How do we represent graphs?

- Option 1: adjacency matrix

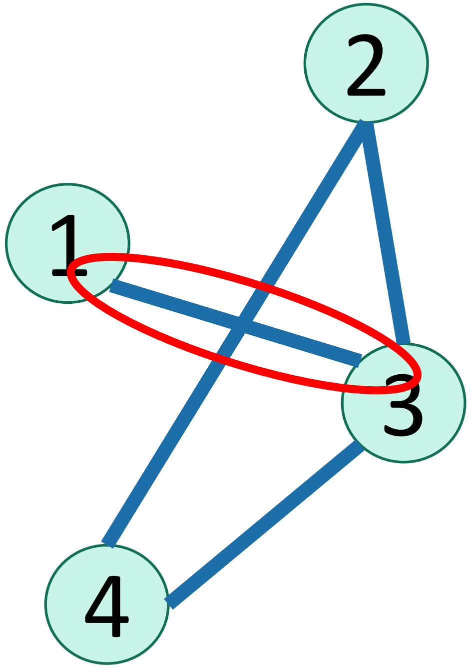
$$
\begin{array}{c c}
 & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\
\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}
\end{array}
$$



How do we represent graphs?

- Option 1: adjacency matrix

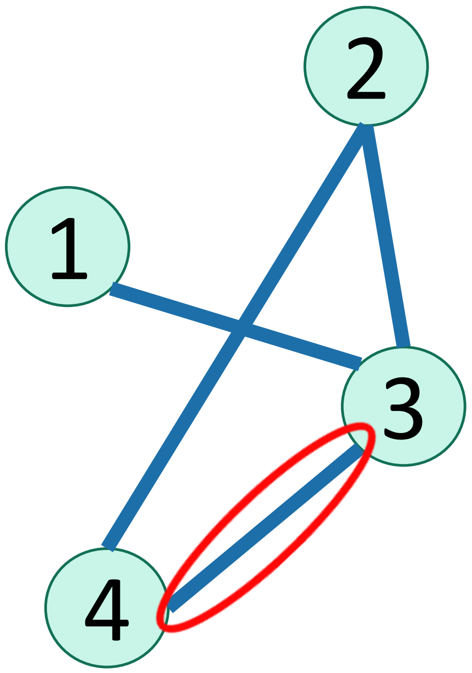
$$
\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array}
\begin{array}{c} \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\
\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}
\end{array}
$$



How do we represent graphs?

- Option 1: adjacency matrix

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



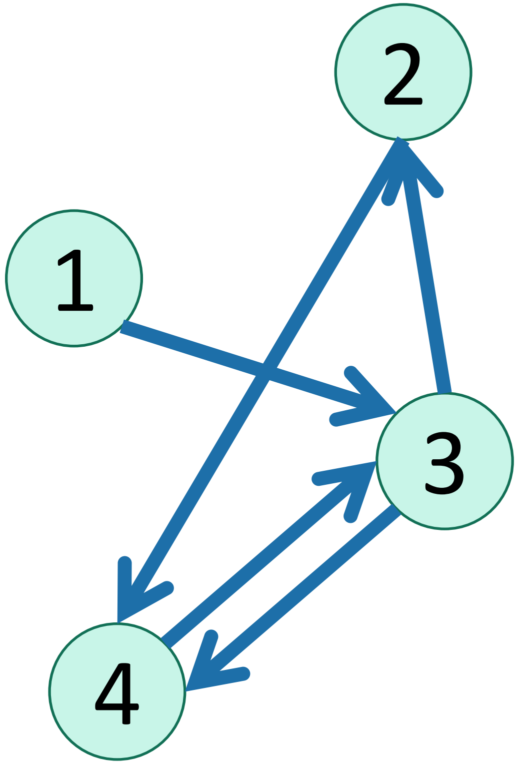
How do we represent graphs?

- Option 1: adjacency matrix (directed graph)

Destination: 1 2 3 4

Source: 1 2 3 4

$$
\begin{bmatrix}
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 \\
0 & 1 & 0 & 1 \\
0 & 0 & 1 & 0
\end{bmatrix}
$$

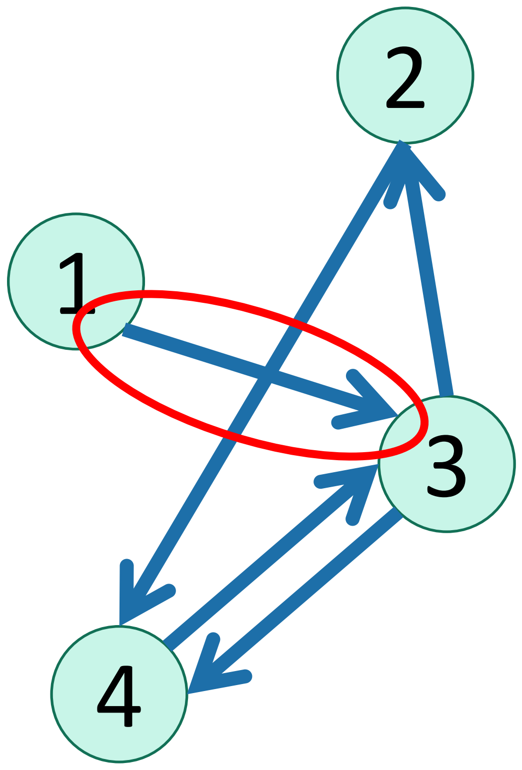


How do we represent graphs?

- Option 1: adjacency matrix (directed graph)

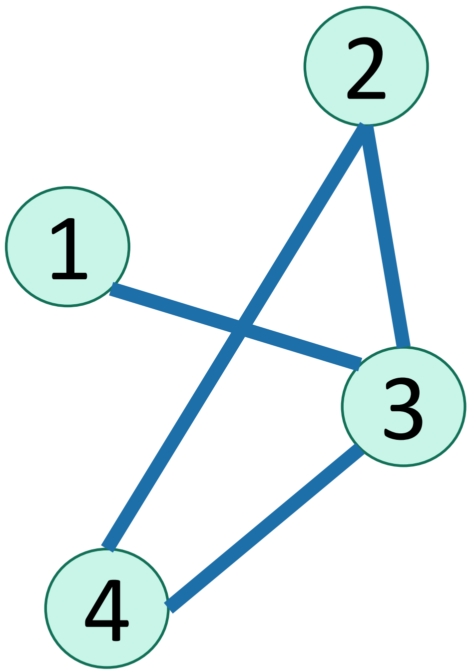
Destination

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



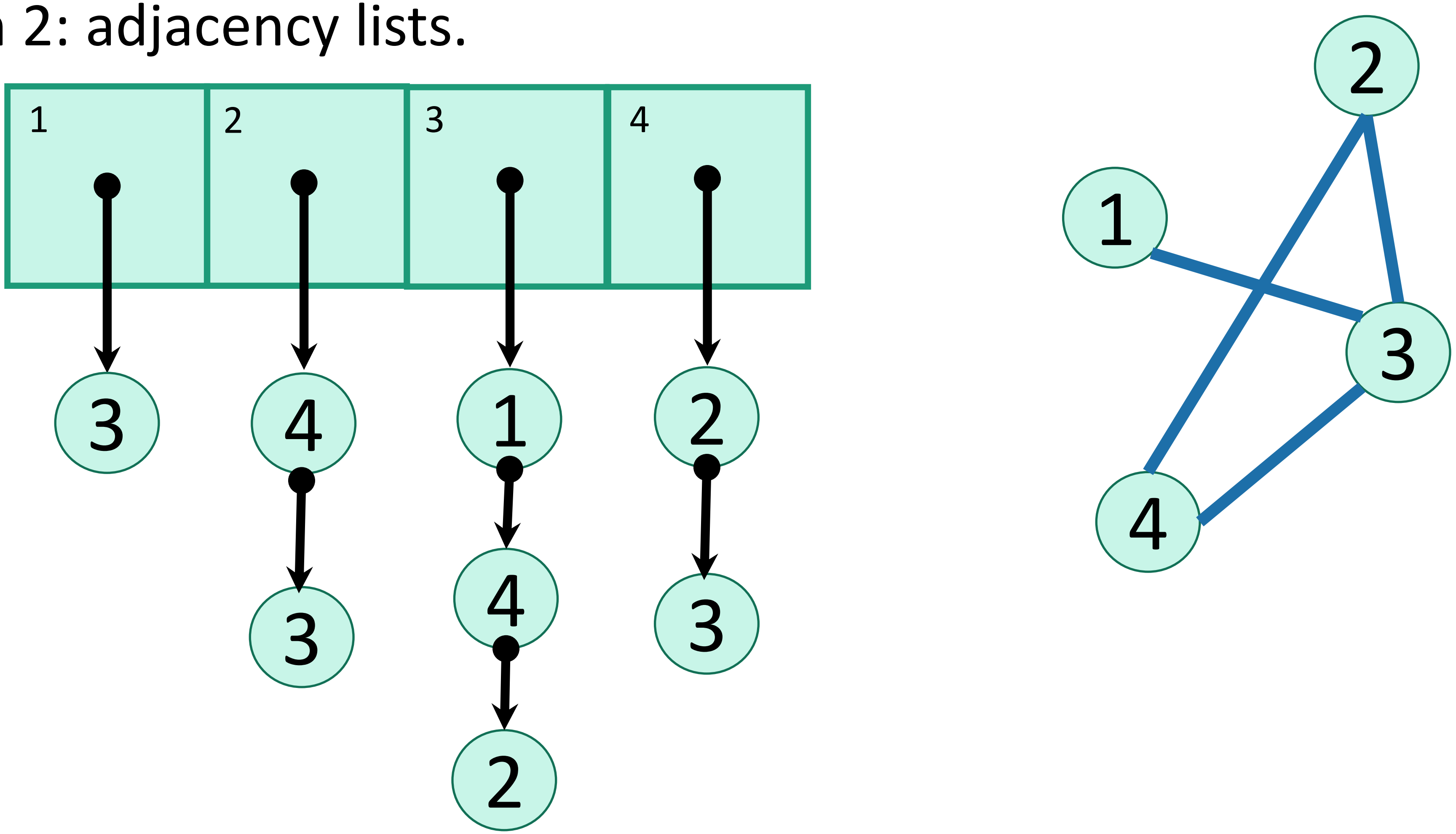
How do we represent graphs?

- Option 2: adjacency lists.



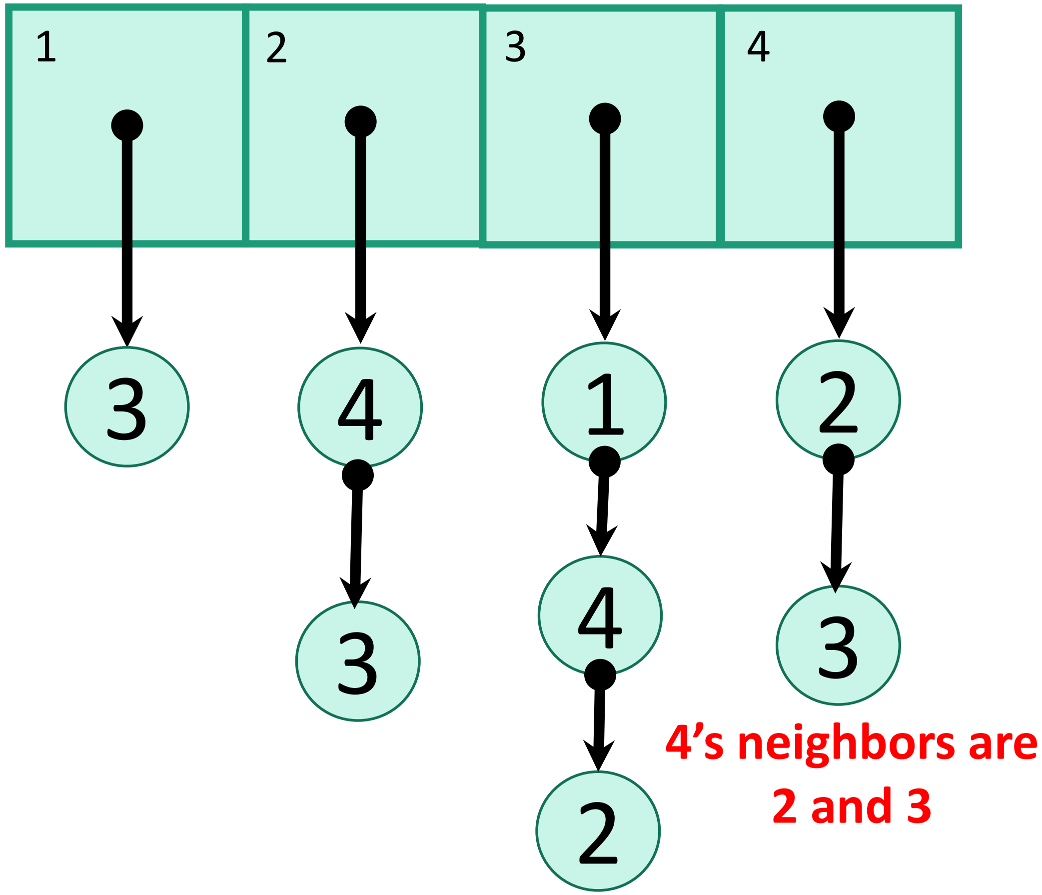
How do we represent graphs?

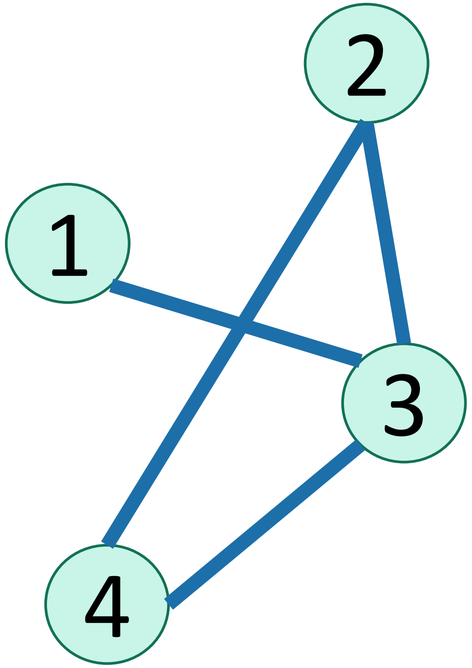
- Option 2: adjacency lists.



How do we represent graphs?

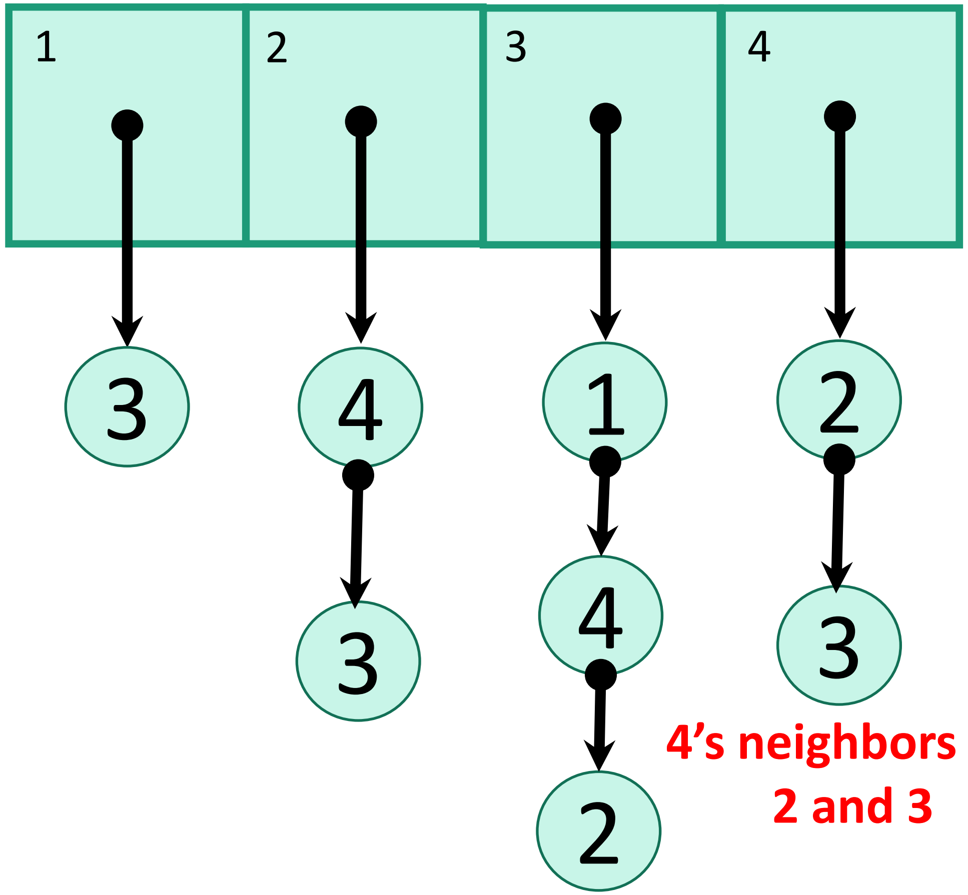
- Option 2: adjacency lists.



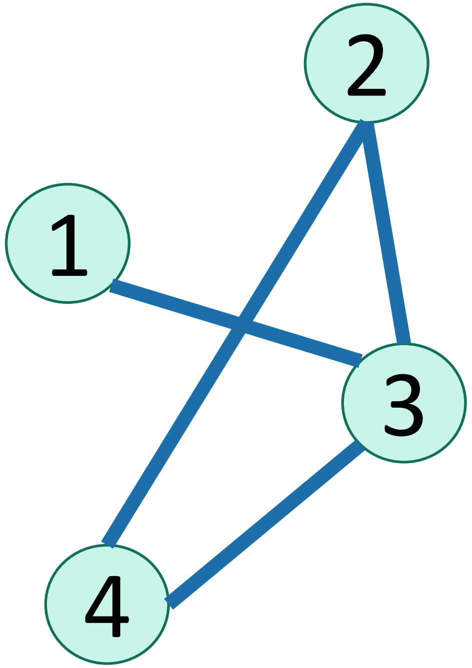


How do we represent graphs?

- Option 2: adjacency lists.



4's neighbors are 2 and 3



How would you modify this for directed graphs?

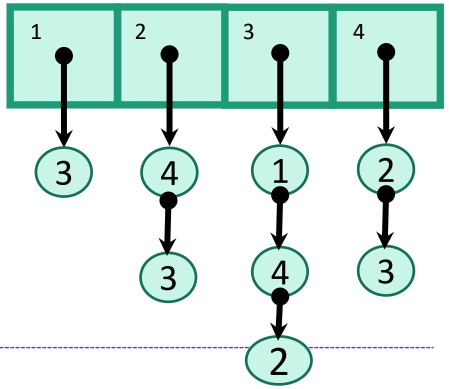
In either case

- Vertices can store other information
  - Attributes (name, IP address, …)
  - helper info for algorithms that we will perform on the graph

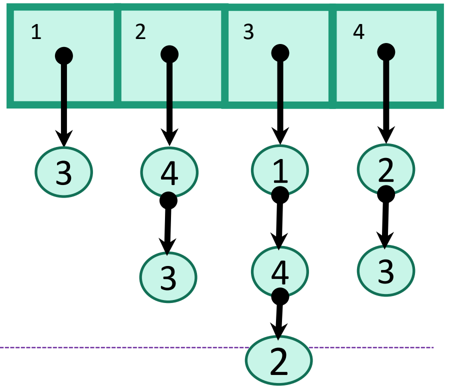
In either case

- 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?

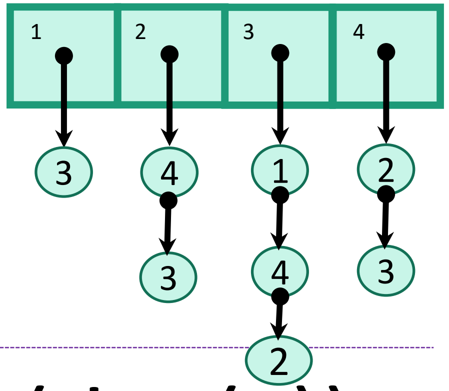
Trade-offs

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?		
Neighbor query Give me v's neighbors.		
Space requirements		

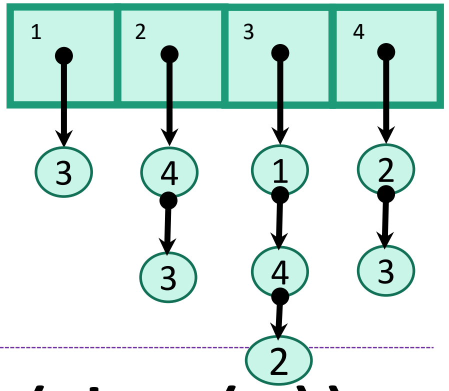
Trade-offs

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)	
Neighbor query Give me v's neighbors.		
Space requirements		

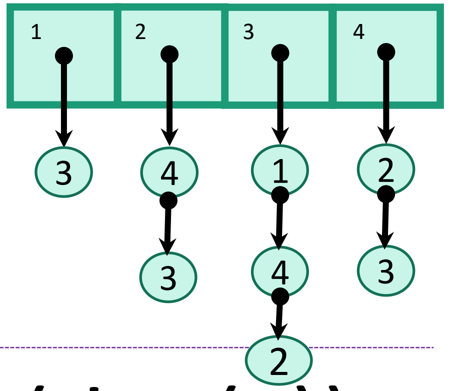
Trade-offs

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.		
Space requirements		

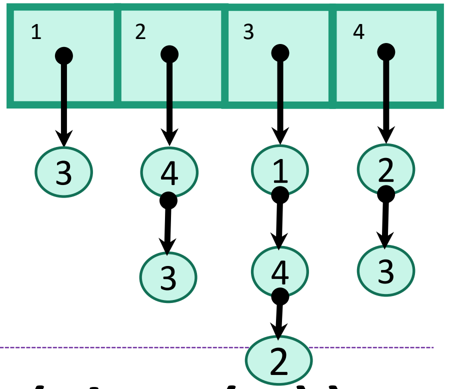
Trade-offs

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)	
Space requirements		

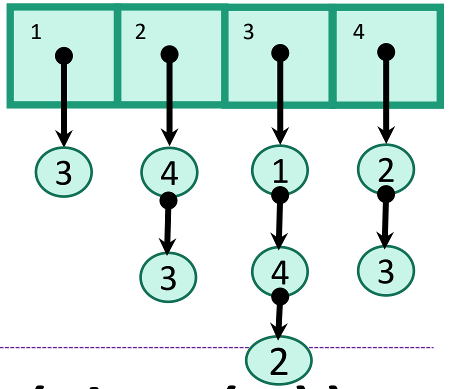
Trade-offs

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		

Trade-offs

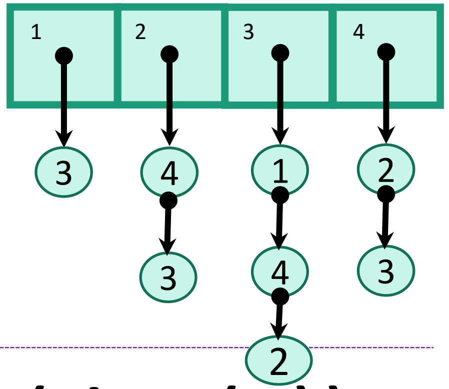
Say there are n vertices and m edges.	$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$	
Edge membership Is e = {v,w} in E?	O(1)	O(deg(v)) or O(deg(w))
Neighbor query Give me v's neighbors.	O(n)	O(deg(v))
Space requirements	$O(n^2)$	

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)

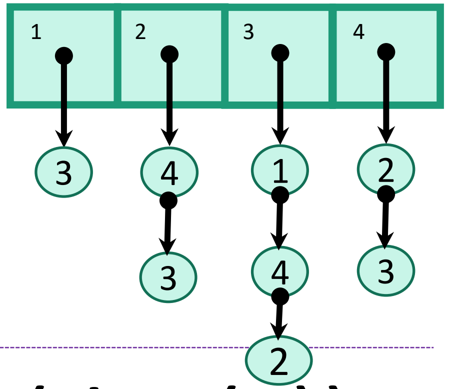
Trade-offs

Generally better for **sparse** graphs

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)

Trade-offs

Generally better for **sparse** graphs

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

Thank You