

# **Social and Information Networks**

**CSCC46H, Fall 2025**

**Lecture 3**

Prof. Ashton Anderson  
[ashton@cs.toronto.edu](mailto:ashton@cs.toronto.edu)



# Logistics

**A1 out this week, due Thursday, Oct 9 on MarkUs**

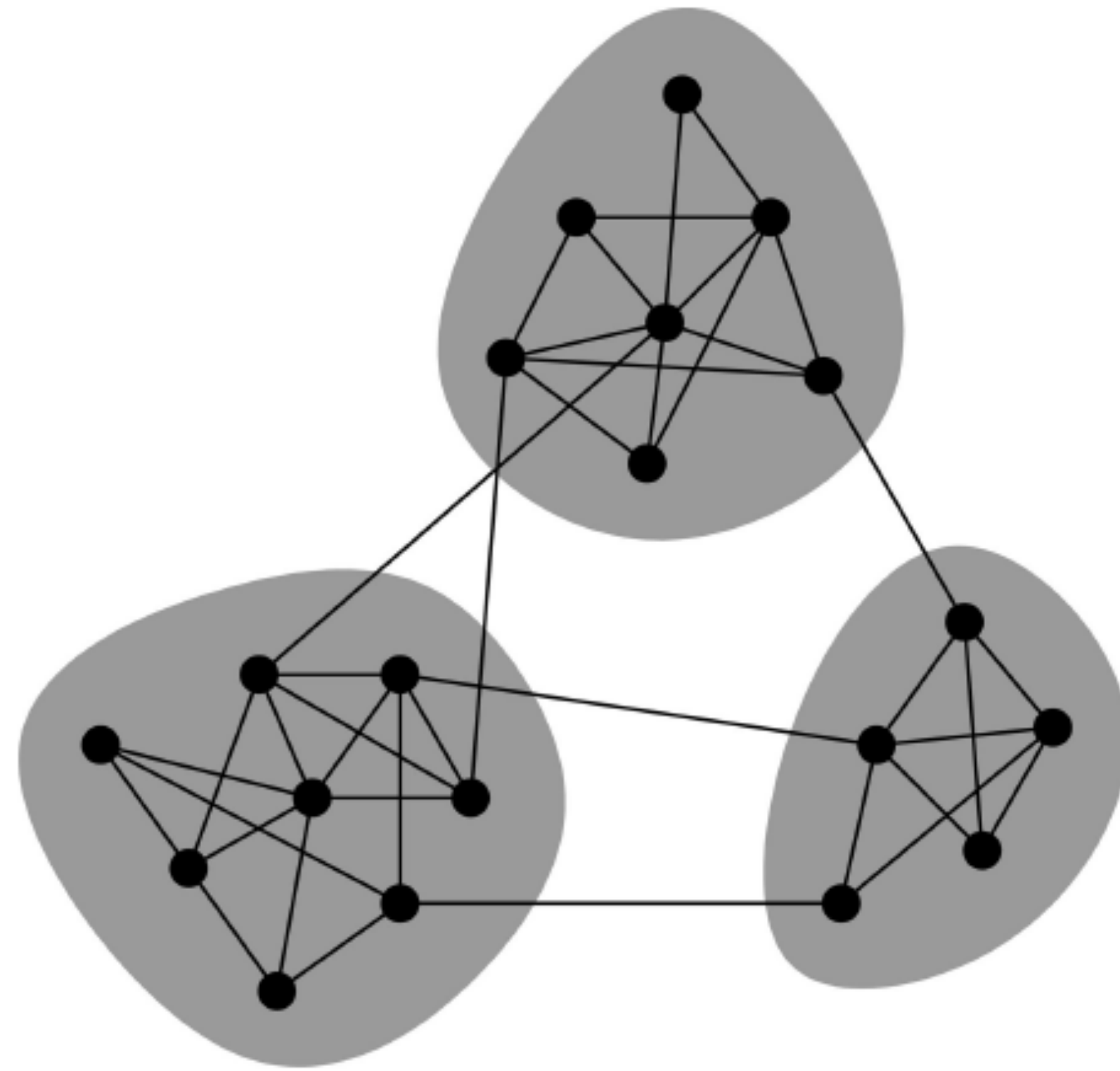
**Tutorials have begun**

# Strong and weak ties

**Modeling relationships of varying strength**

# Networks & Communities

We often think of networks “looking” like this:



**What can lead to such a conceptual picture?**



# Networks: Flow of Information

- **How does information flow through networks?**

What structurally distinct roles do nodes play?

What roles do different **links** (short vs. long) play?

- **How people find out about new jobs?**

Mark Granovetter, part of his PhD in 1960s

People find the information through personal contacts

- **But:** Contacts were often **acquaintances**  
rather than close friends

**This is surprising:** One would expect your friends to help you out more than casual acquaintances

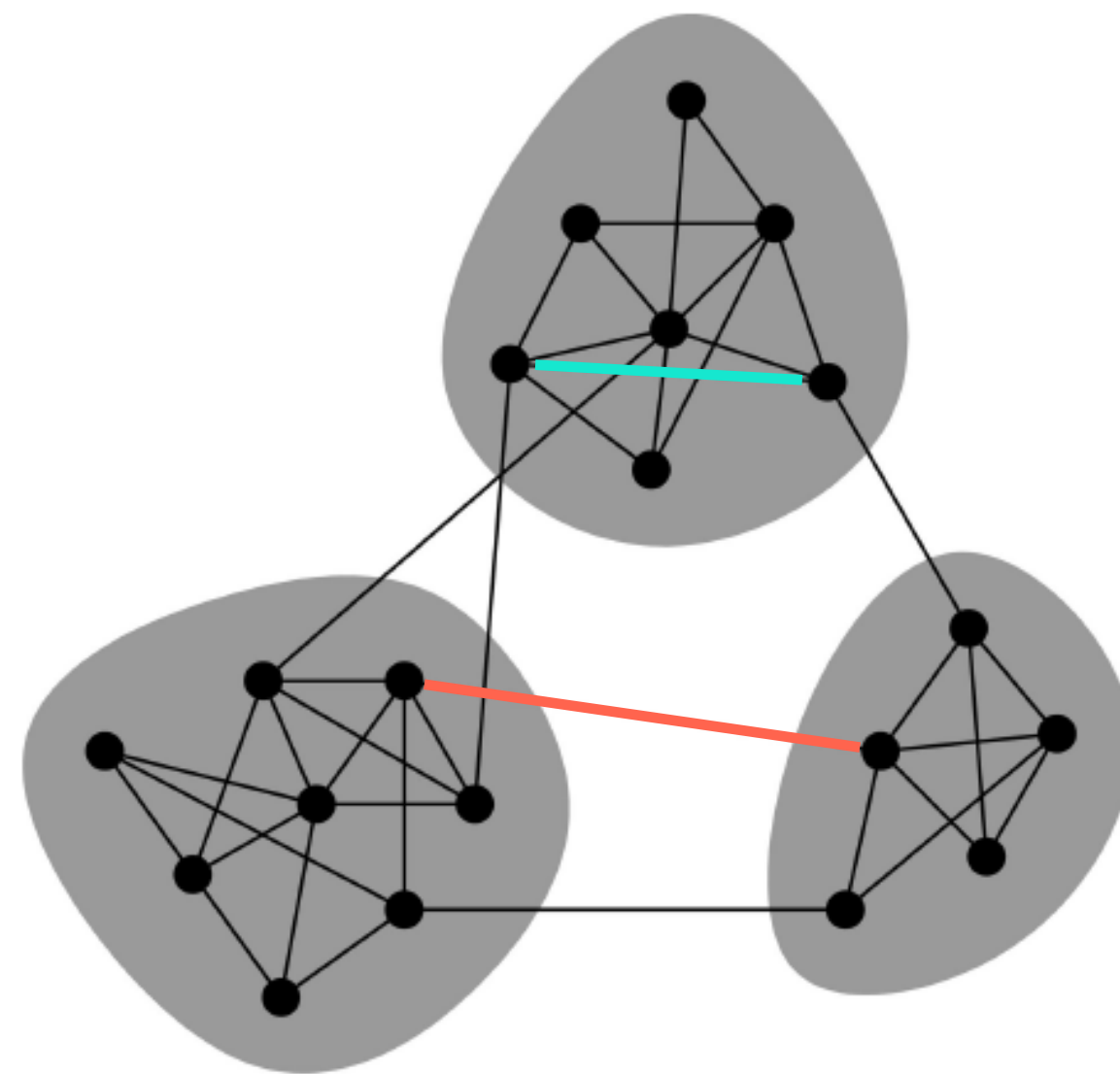
**Why is it that acquaintances are most helpful?**



# Granovetter's Answer

Two perspectives on **friendships**:

**Structural:** Friendships span different parts of the network

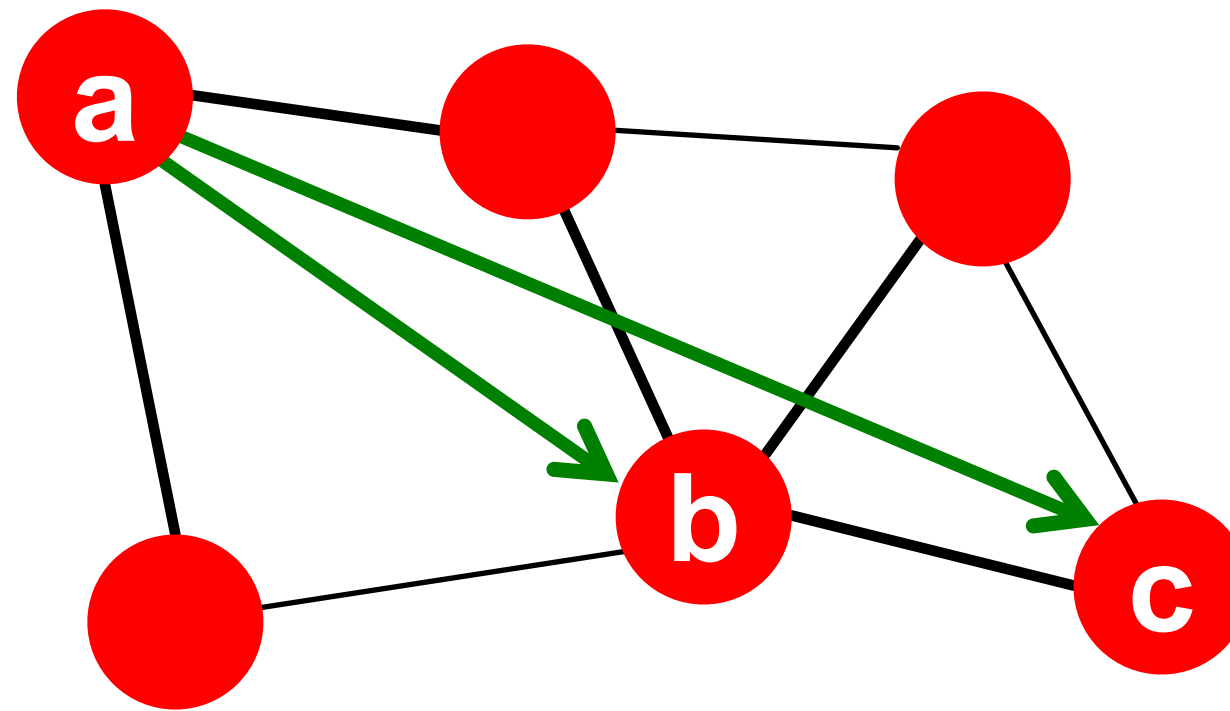


The two highlighted edges are structurally different: one spans two different “communities” and the other is inside a community

**Interpersonal:** Friendship between two people vary in strength, you can be close or not so close to someone



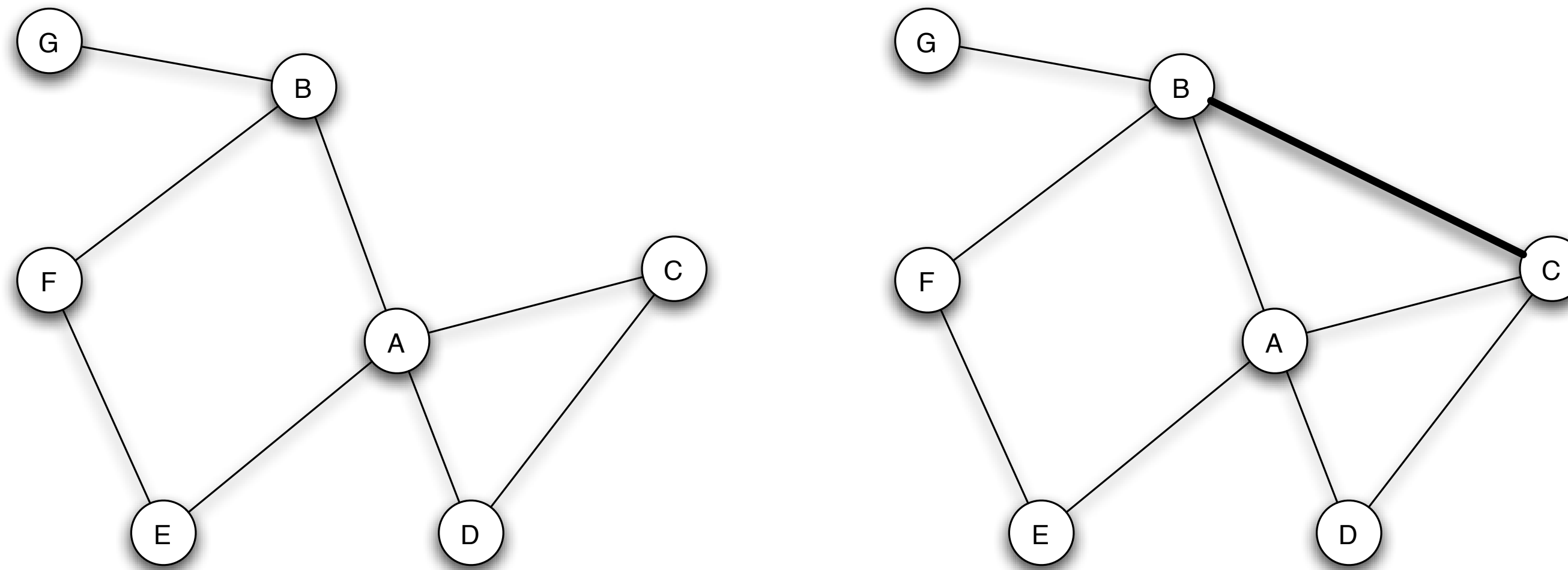
# How do new edges form?



Which edge is more likely:  
a–b or a–c?



# Triadic closure



**Informally:** If two people in a social network have a friend in common, then there is an **increased likelihood** that they will become friends themselves at some point in the future.



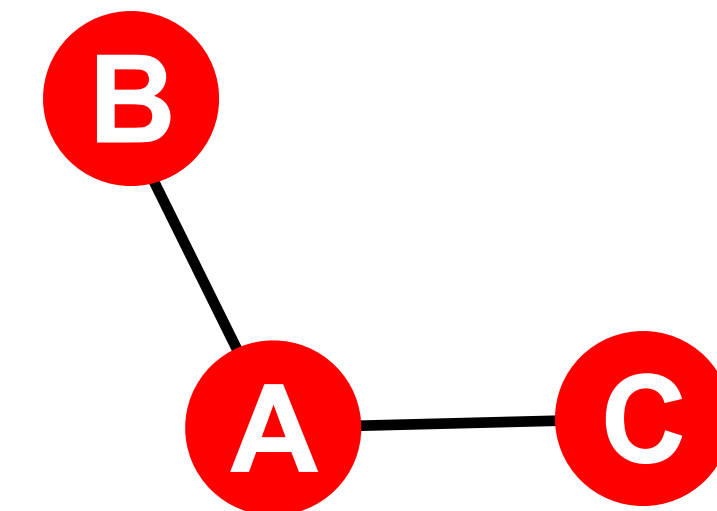
# Triadic Closure

**Triadic closure == High clustering coefficient**

## Reasons for triadic closure:

If **B** and **C** have a friend **A** in common:

- **B** is **more likely to meet C**  
(both **spend time** with **A**)
- **B** and **C** **trust each other more**  
(they have a **friend in common**)
- **A** has an **incentive to bring B and C together**  
(**easier** for **A** to **maintain** two disjoint relationships)





# Granovetter's Explanation

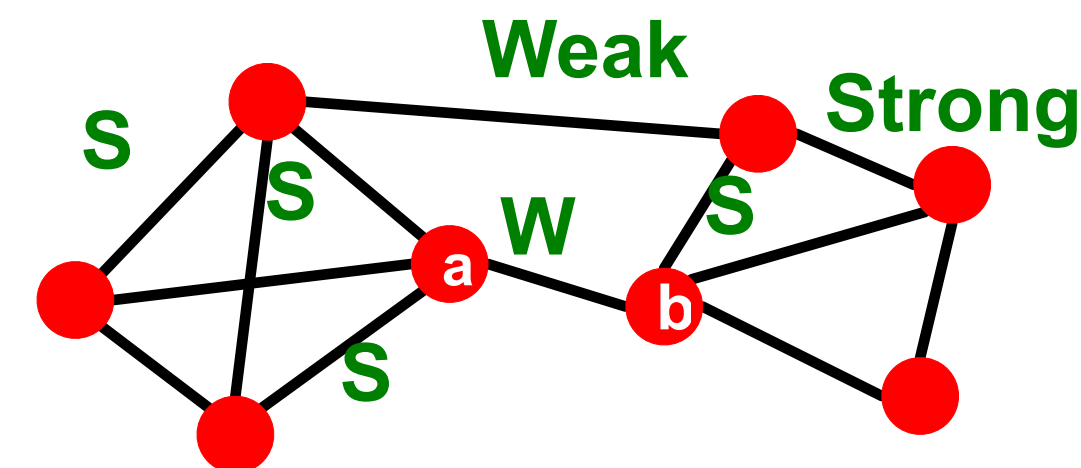
**Granovetter makes a connection between the social and structural roles of an edge**

## ■ **First point: Structure**

- Structurally embedded edges are also socially strong
- Long-range edges spanning different parts of the network are socially weak

## ■ **Second point: Information**

- Long-range edges allow you to gather information from different parts of the network and get a job
- Structurally embedded edges are heavily redundant in terms of information access

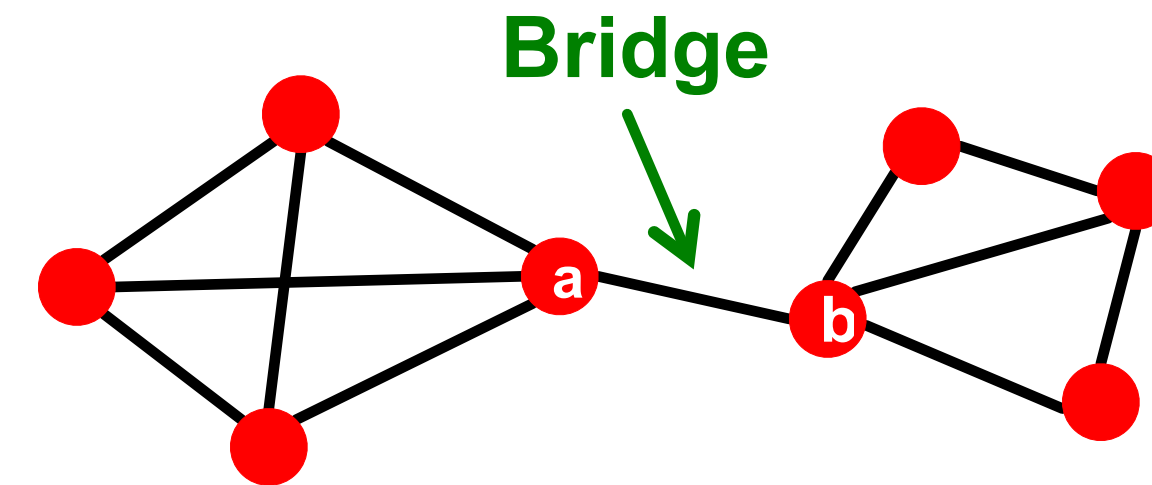




# Network Vocabulary: Span and Bridges

## Define: **Span**

The **Span** of an edge is the distance of the edge endpoints if the edge is deleted.



## Define: **Bridge edge**

If removed, it disconnects the graph

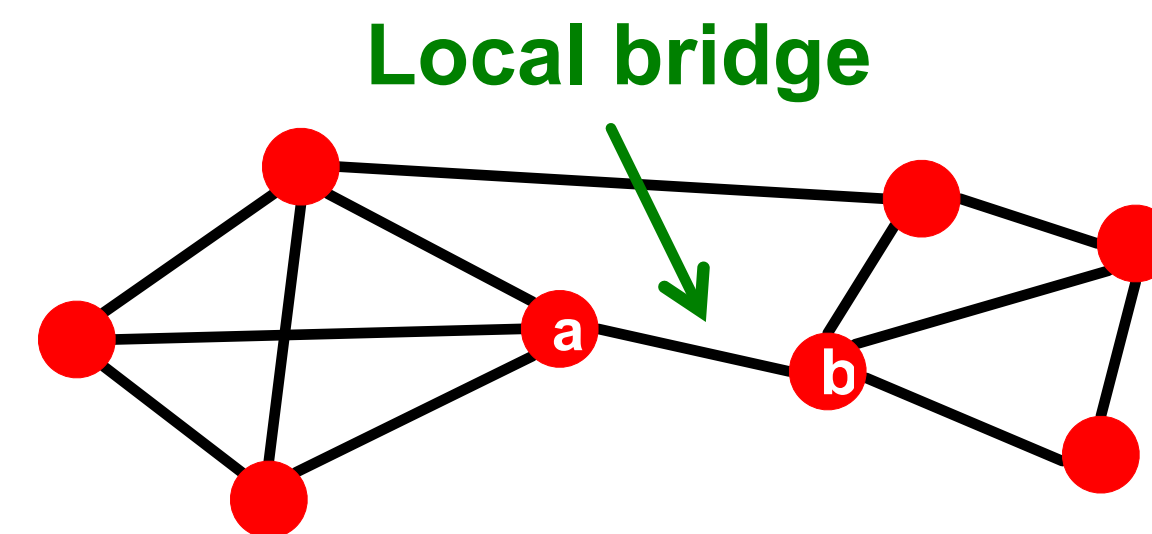
Span of a bridge edge =  $\infty$

## Define: **Local bridge**

Edge of **Span**  $> 2$

(any edge that doesn't close a triangle)

Idea: Local bridges with long span are like real bridges

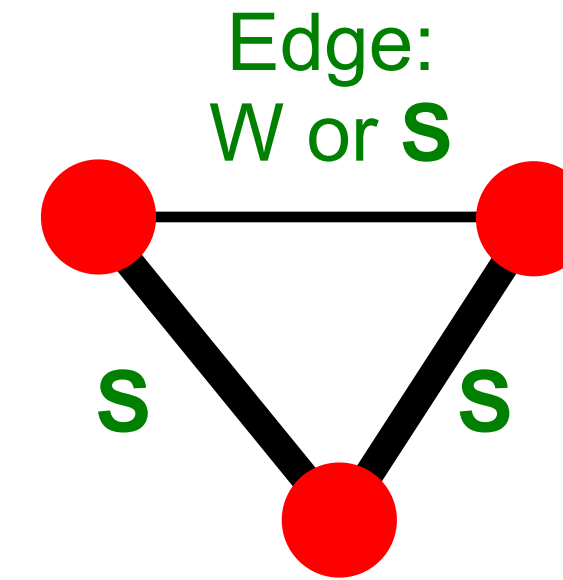




# Granovetter's Explanation

Model: Two types of edges:

**Strong** (friend), **Weak** (acquaintance)

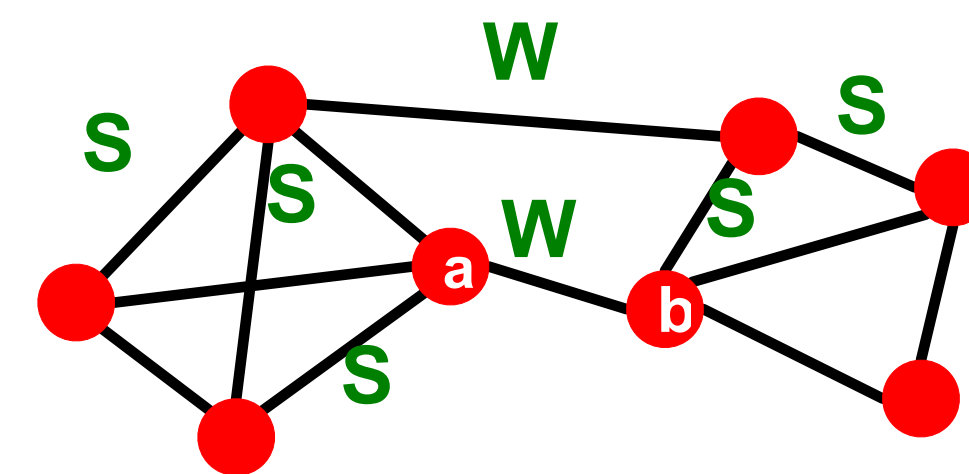


Model: **Strong Triadic Closure property**:

**Two strong ties imply a third edge**

If node A has strong ties to both nodes B and C, then there must be an edge (strong or weak) between B and C

Fact: If strong triadic closure is satisfied then **local bridges are weak ties!**

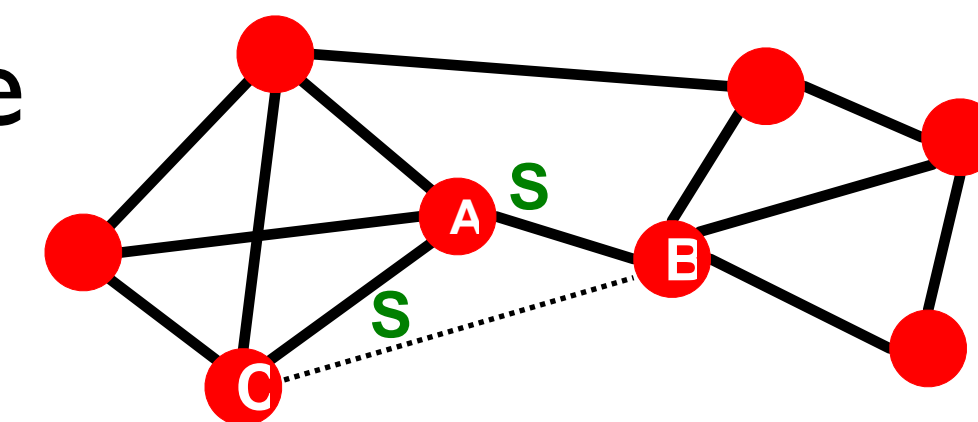
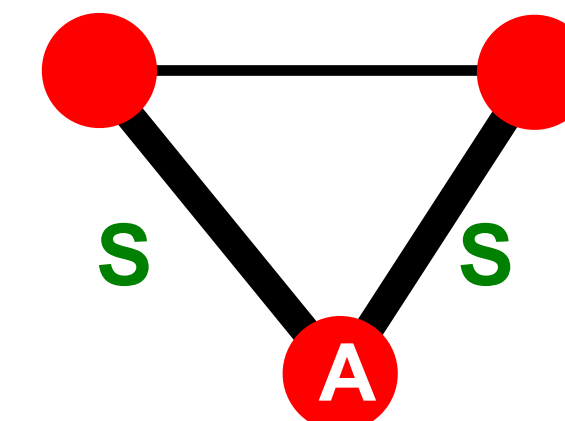


# Local Bridges and Weak ties

**Claim:** if node A satisfies **Strong Triadic Closure** and has two strong ties, then **any local bridge adjacent to A must be a weak tie**

**Proof:** By contradiction:

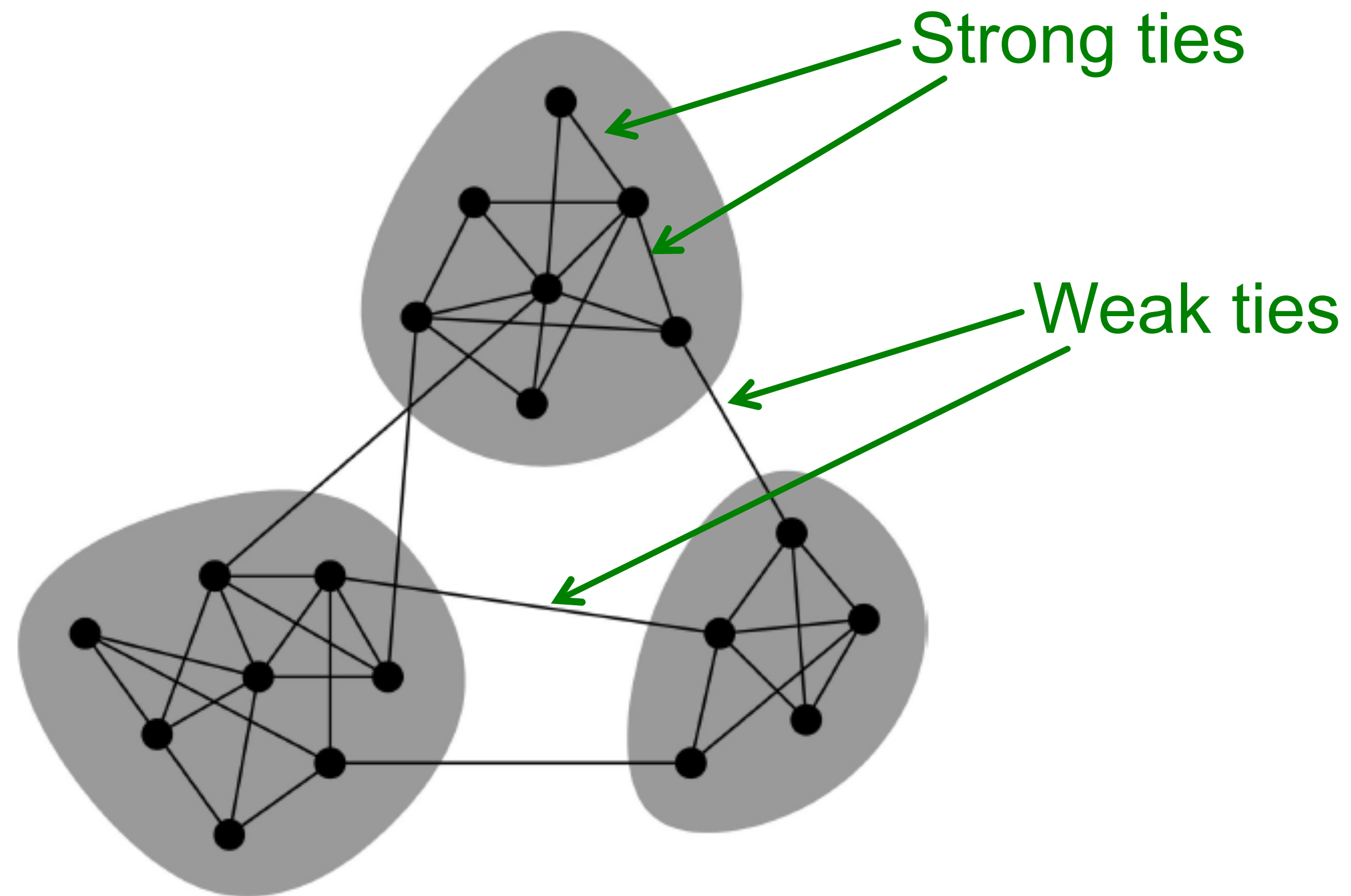
- Assume A satisfies **Strong Triadic Closure** and has **two strong ties**
- Let A–B be a **local bridge**, and assume it is a **strong tie** (to try to derive a contradiction)
- Then B–C must exist because of **Strong Triadic Closure**
- But then **A–B is not a local bridge**, because its span is 2 (without A–B, A–C–B is the shortest path)





# Conceptual Picture of Networks

Granovetter's theory leads to the following conceptual picture of networks



# Granovetter's Explanation

**Weak** ties have access to **different parts of the network!** Access to other sources and other kinds of information

**Strong** ties have **redundant information**



# Tie strength in real data

**For many years Granovetter's theory was not tested**

But, today we have large who-talks-to-whom graphs:  
Email, Messenger, Cell phones, Facebook

**Onnela et al. 2007:**

Cell-phone network of 20% of country's population

**Edge strength:** # phone calls

# Neighborhood Overlap

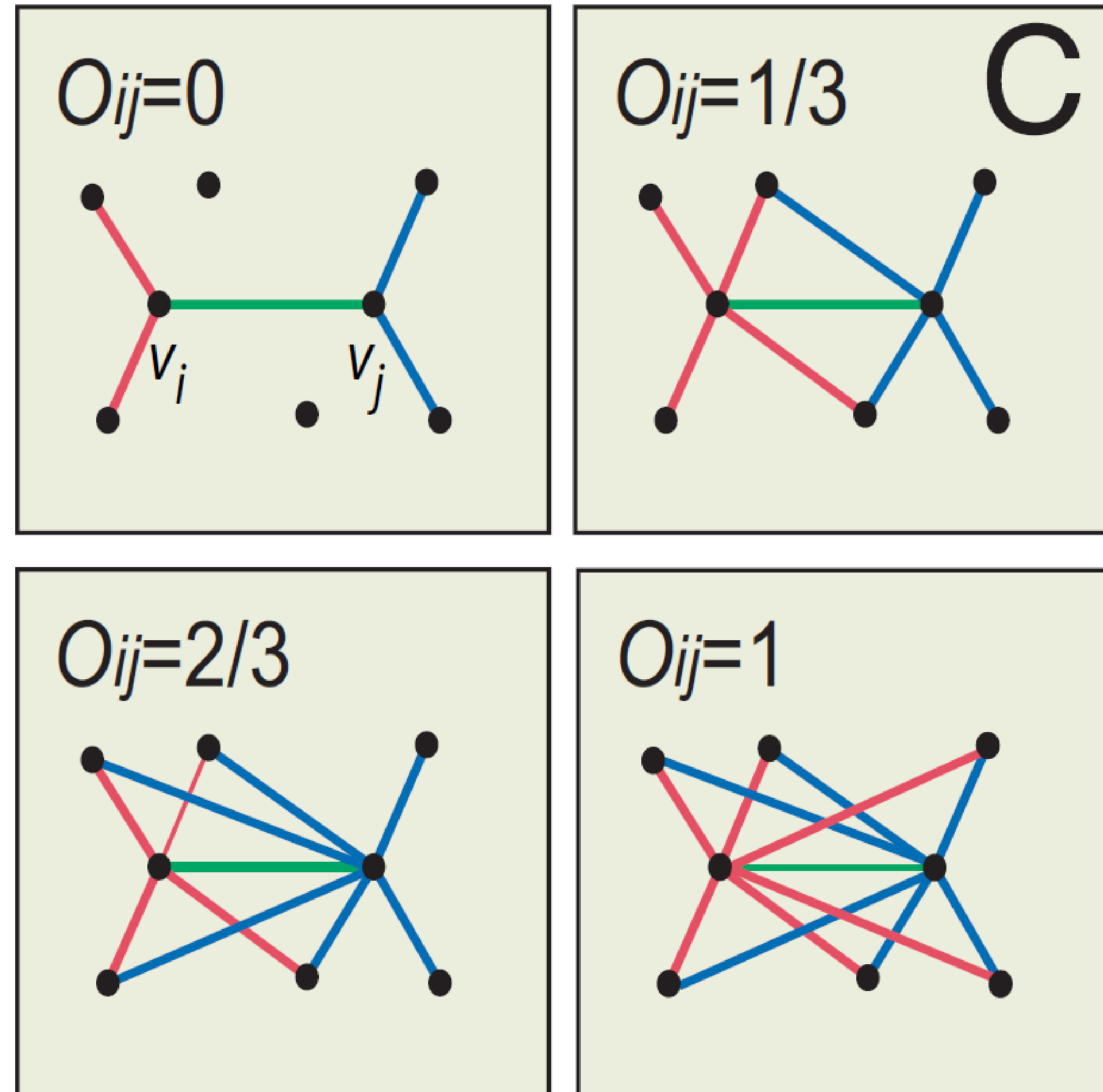
Define: **Edge overlap** as the number of shared neighbours divided by the union of neighbours:

$$O_{ij} = \frac{N(i) \cap N(j)}{N(i) \cup N(j)}$$

( $N(i)$  = set of neighbours of node  $i$ )

$O_{ij} = 0$  when  $i-j$  is a local bridge

$O_{ij} = 1$  when  $i$  and  $j$  have all neighbours in common





# Phones: Edge Overlap vs. Strength

Let's measure the **empirical relationship** between **edge strength** and **overlap** in a real network!

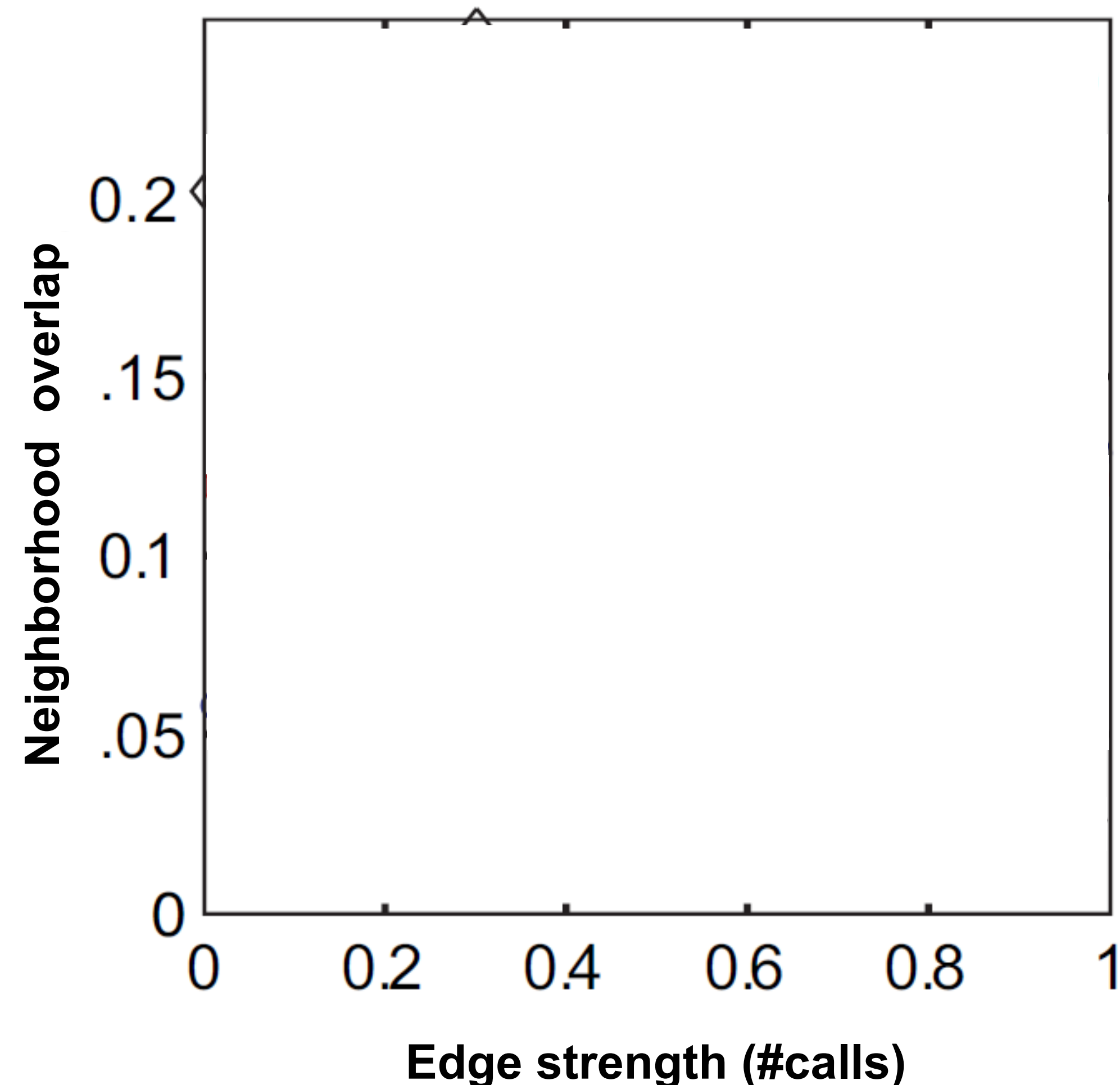
**Data:** cell phone network

## Legend:

**x-axis:** edge strength (# calls between nodes)

**y-axis:** overlap (how much edge bridges different parts of the network)

What do you think it will look like?



# Phones: Edge Overlap vs. Strength

## Legend:

**True:** The data

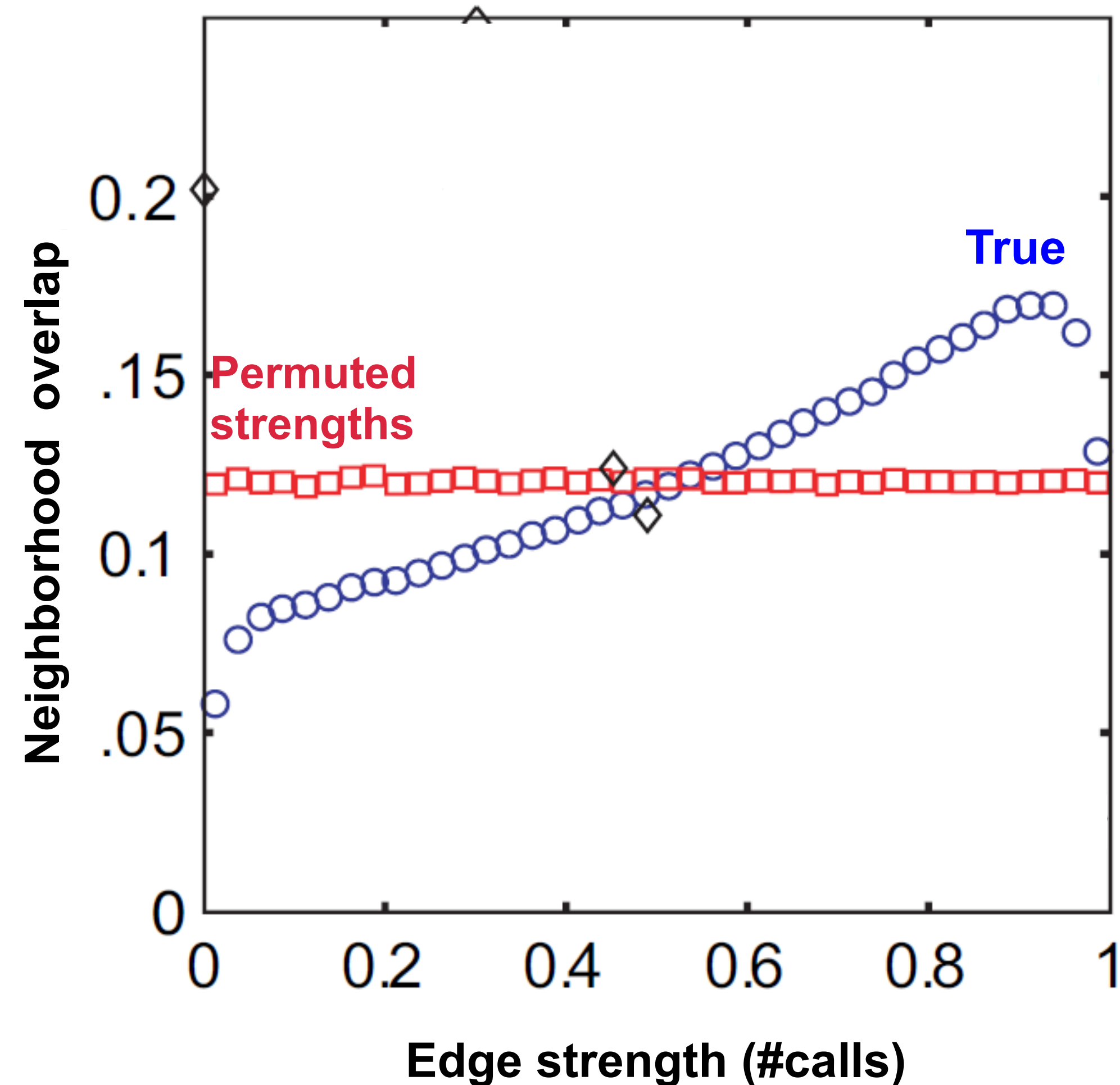
**Permuted strengths:** Keep the network structure but randomly reassign edge strengths

## Observation:

Highly used links have high overlap!

Weak links have small overlap (bridg

Granovetter was right

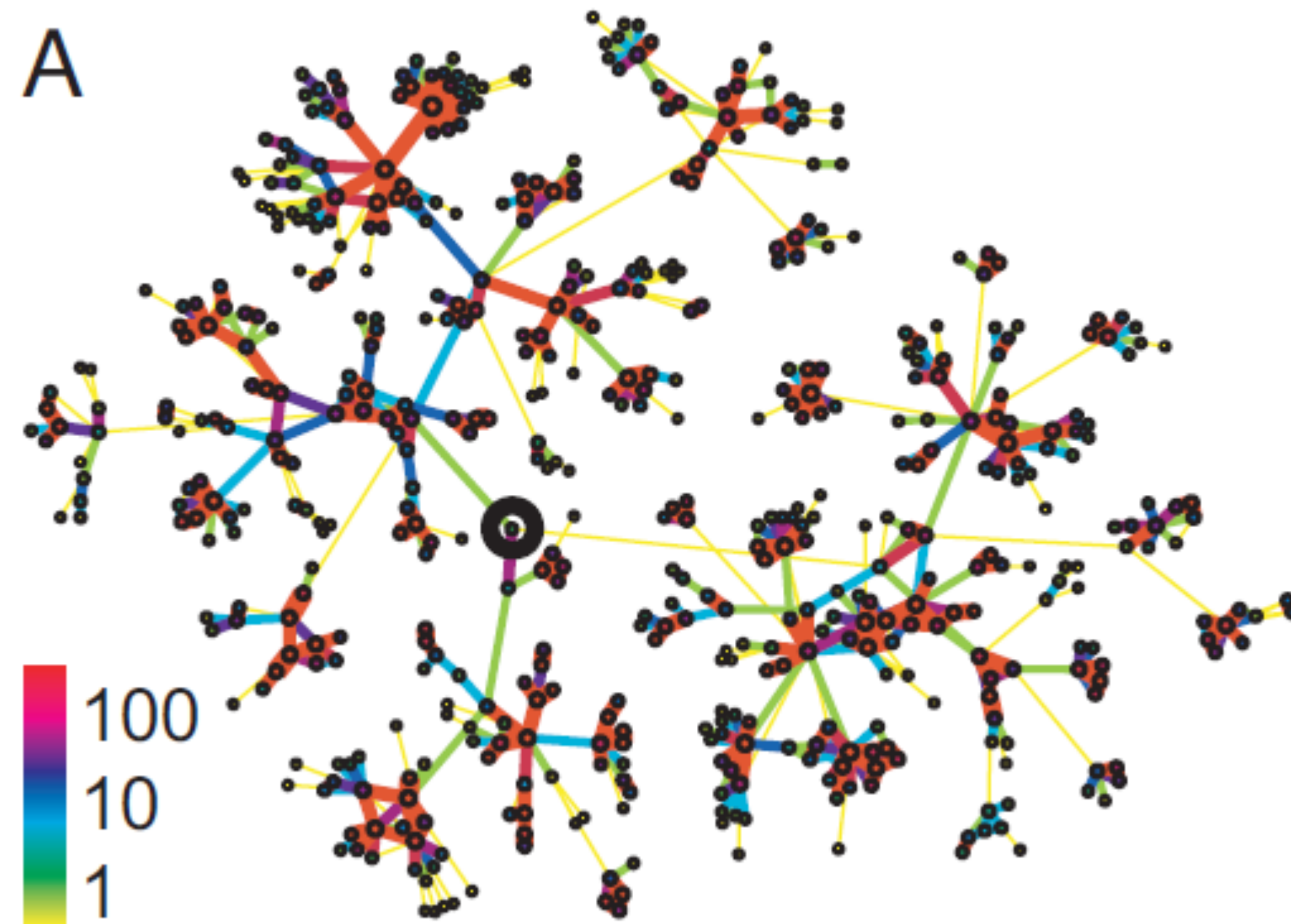




# Real Network, Real Tie Strengths

## Real edge strengths in mobile call graph

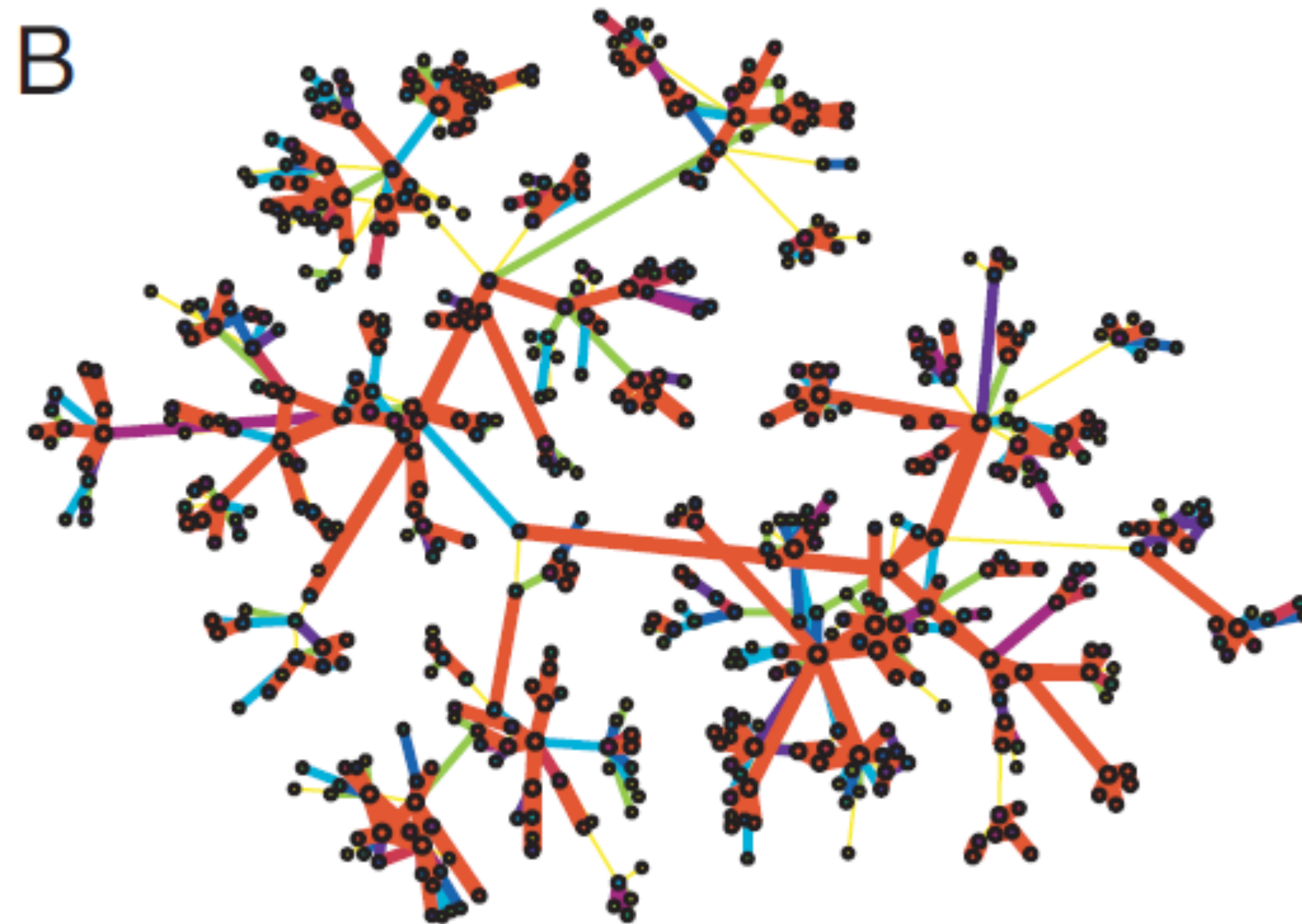
Strong ties are more embedded (have higher overlap), and occur mostly in clustered communities



# Real Net, Permuted Tie Strengths

Same network, same set of edge strengths but now strengths are randomly shuffled

Now high overlap edges are much more likely to span different parts of the network (not what we see in real life)





# Link Removal by Strength

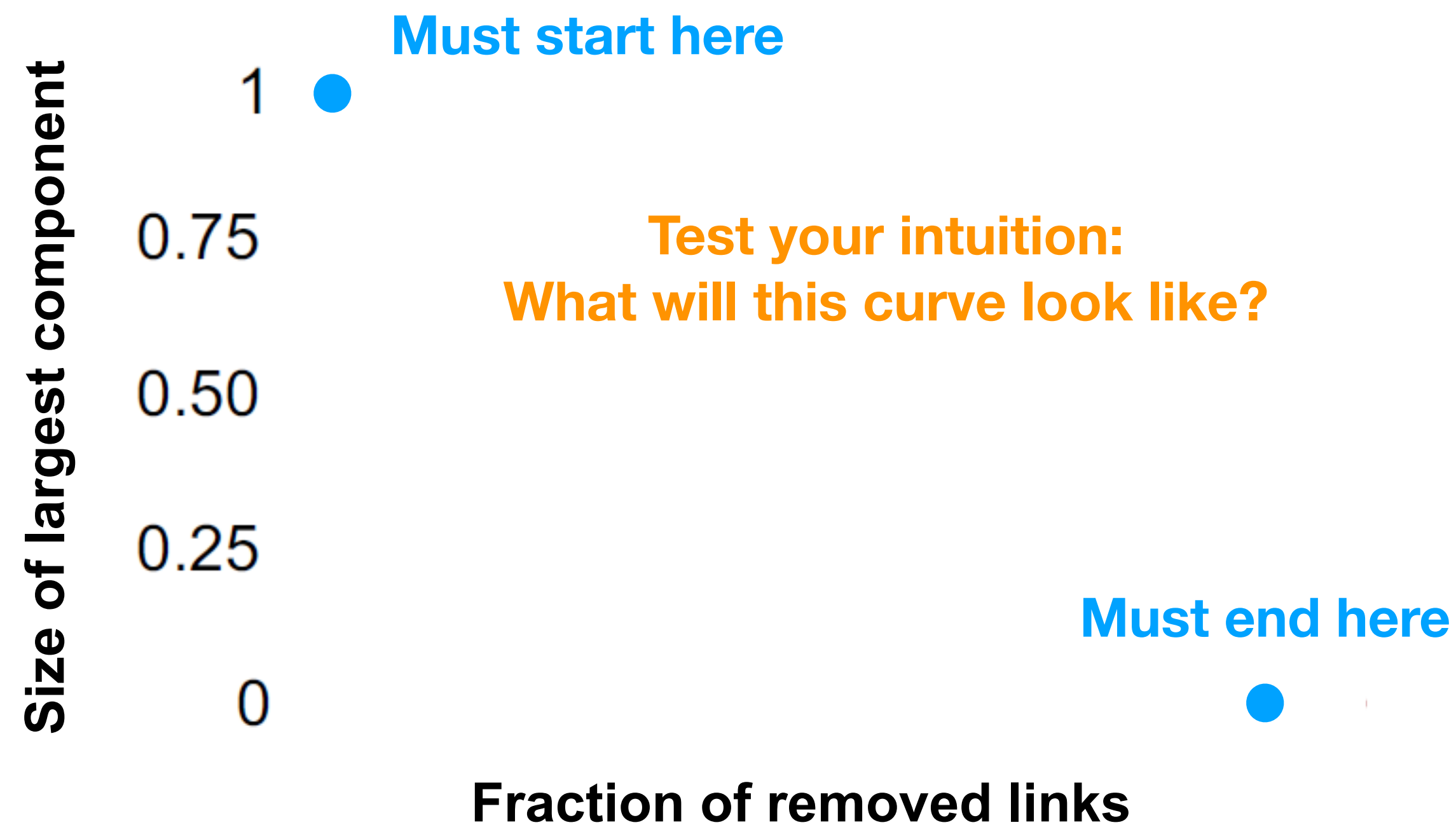
An important, recurring concept in network analysis is **network robustness**: how quickly does the graph **become disconnected** as you remove links?

The faster the network falls apart, the more prone to failure it is

Test **importance of edges** by changing the **order in which you remove them**

# Link Removal by Strength

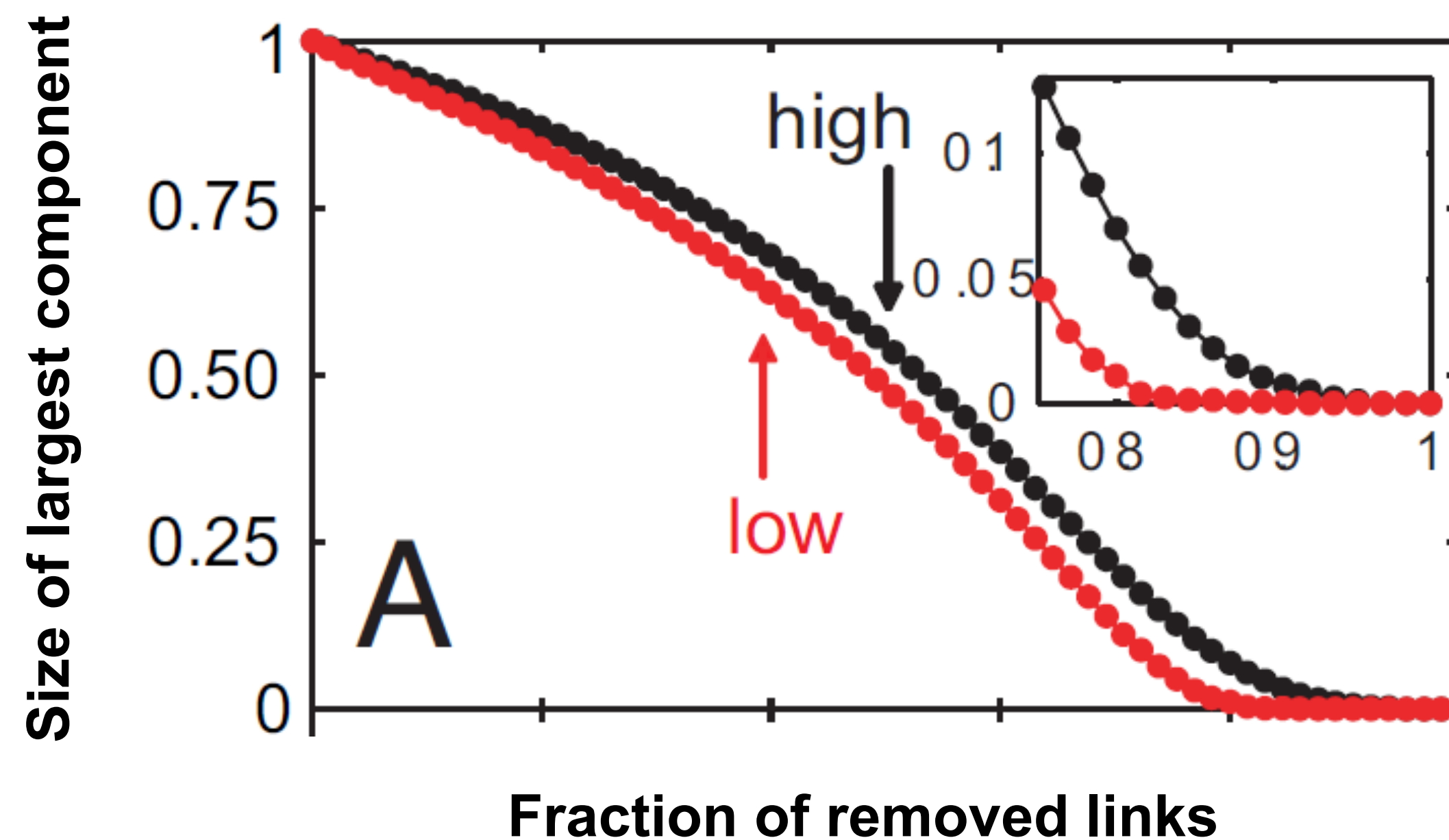
In the mobile call graph, we will test the importance of strong/weak edges, as well as high/low overlap edges, by employing this strategy



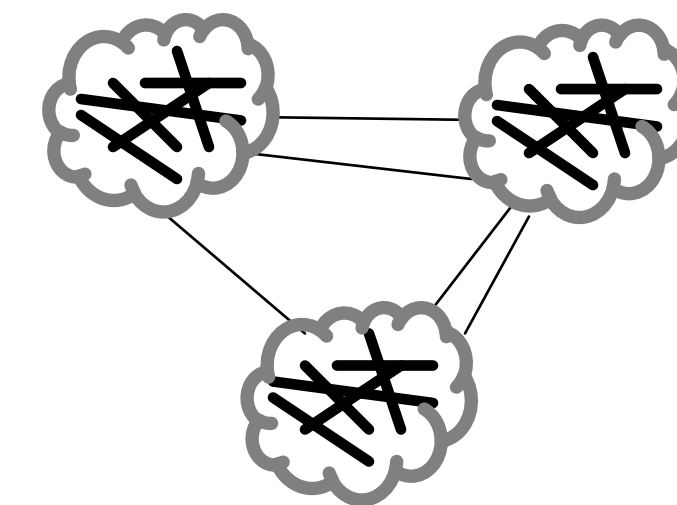
# Link Removal by Strength

Removing links by **strength (#calls)**

- Low to high
- High to low



**Low**  
disconnects  
the network  
sooner



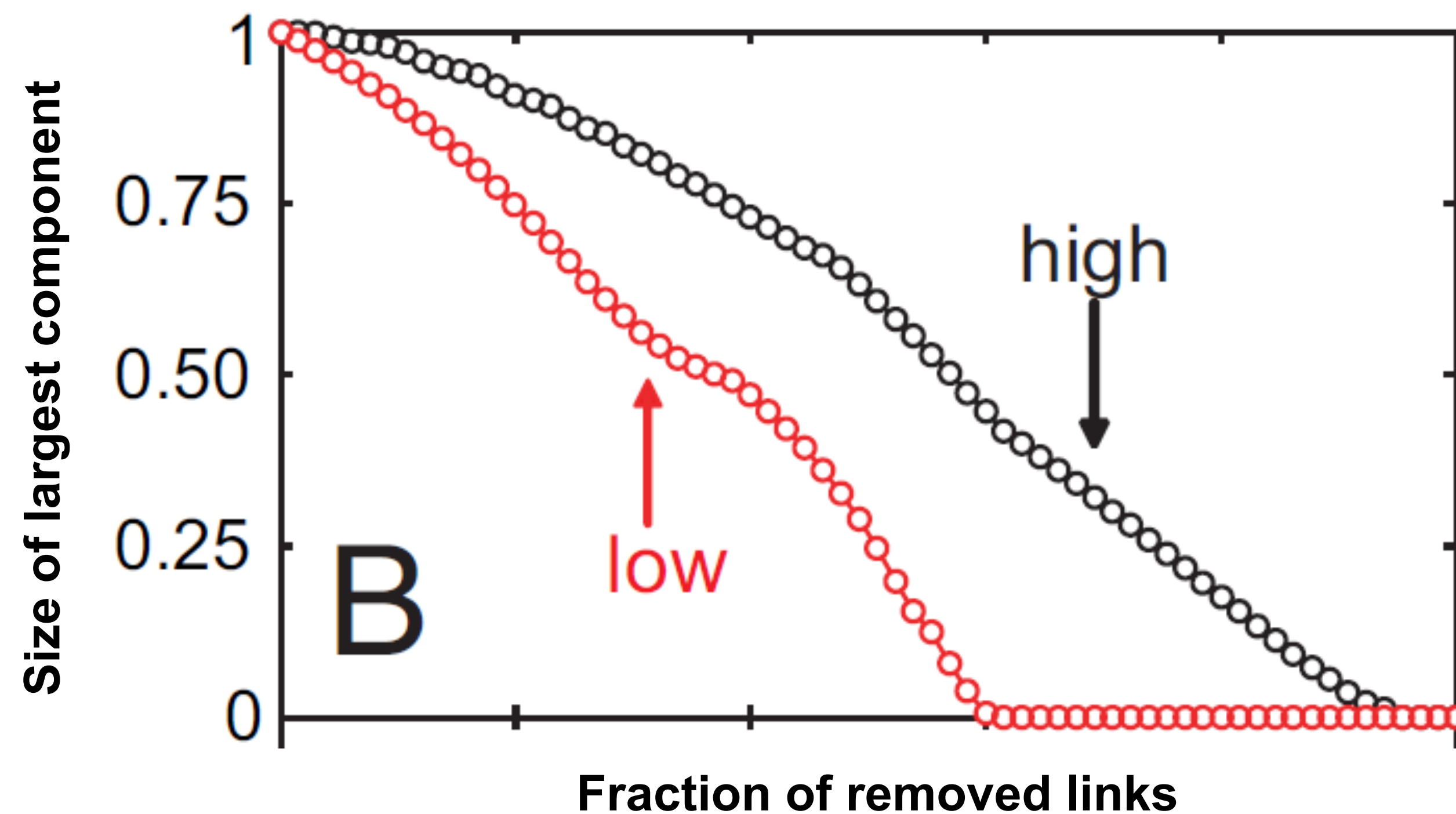
Conceptual picture  
of network structure



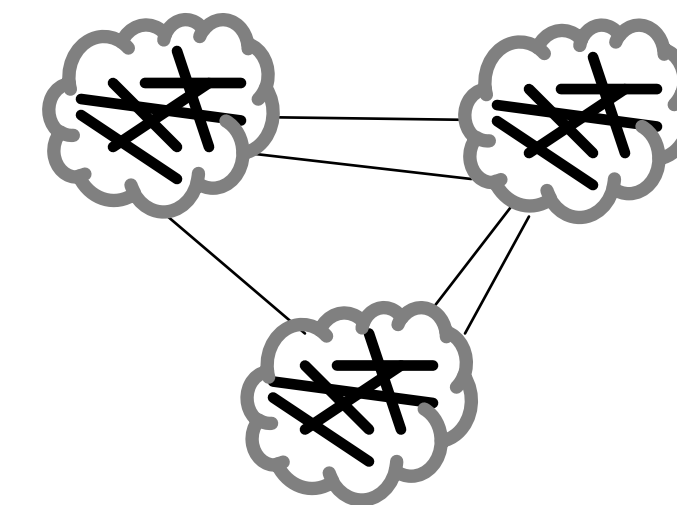
# Link Removal by Overlap

Removing links based on **overlap**

- Low to high
- High to low



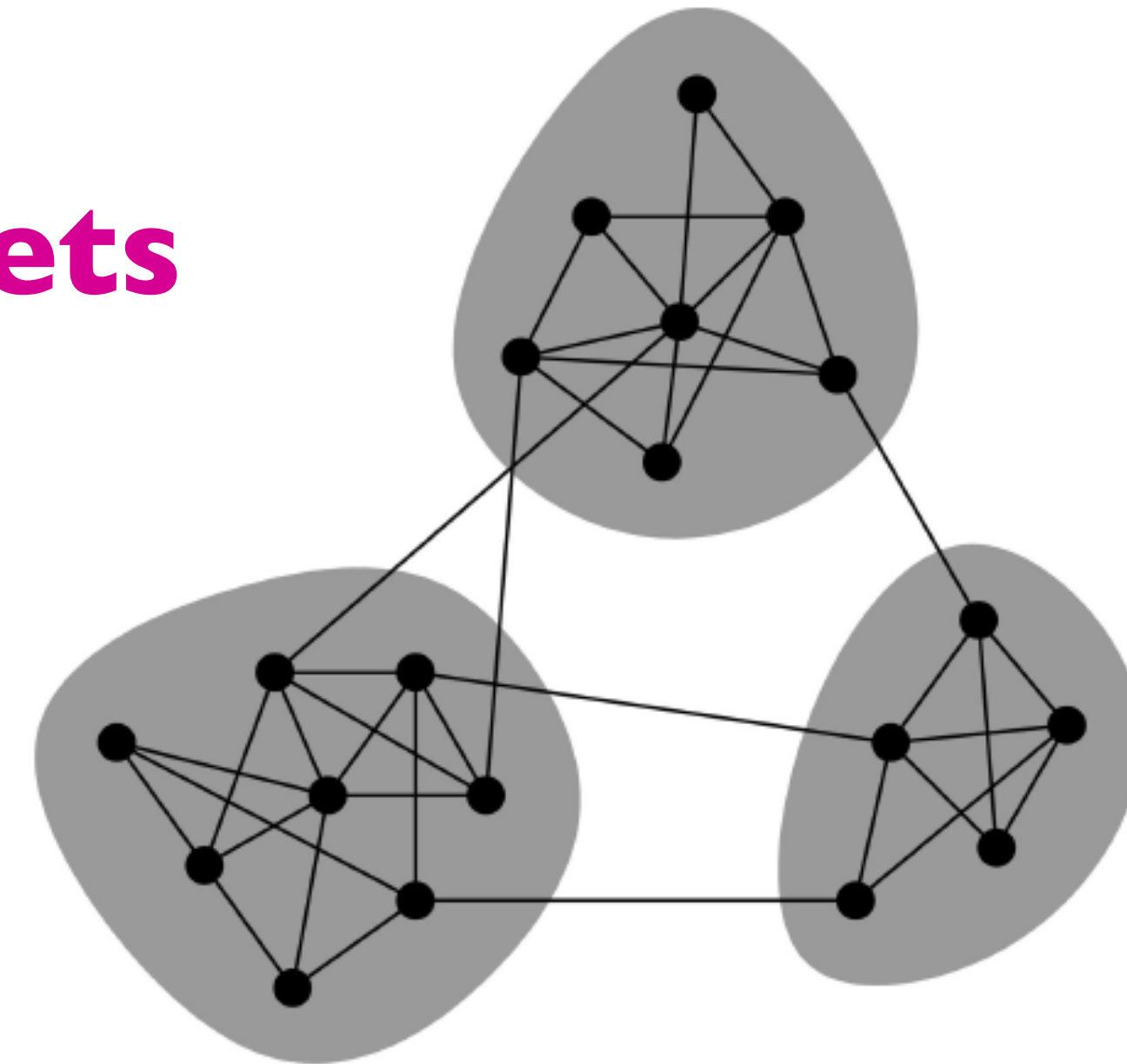
**Low** disconnects the network **much** sooner



Conceptual picture of network structure

# Network Communities

Granovetter's **strength of weak ties theory** suggests that networks are composed of **tightly connected sets of nodes**



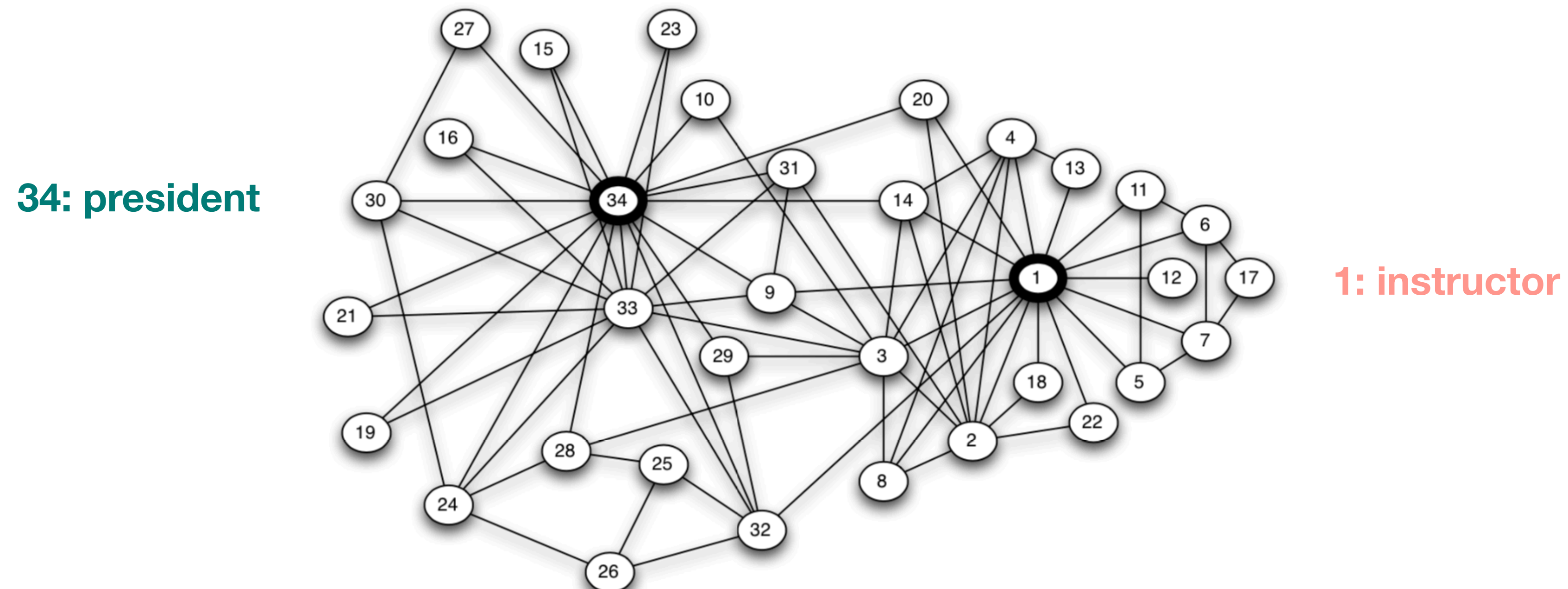
## Network communities:

Sets of nodes with **lots** of connections **inside** and **few** to **outside** (the rest of the network)

# Social Network Data

## Zachary's Karate club network:

Observe social ties and rivalries in a university karate club





# Social Network Data

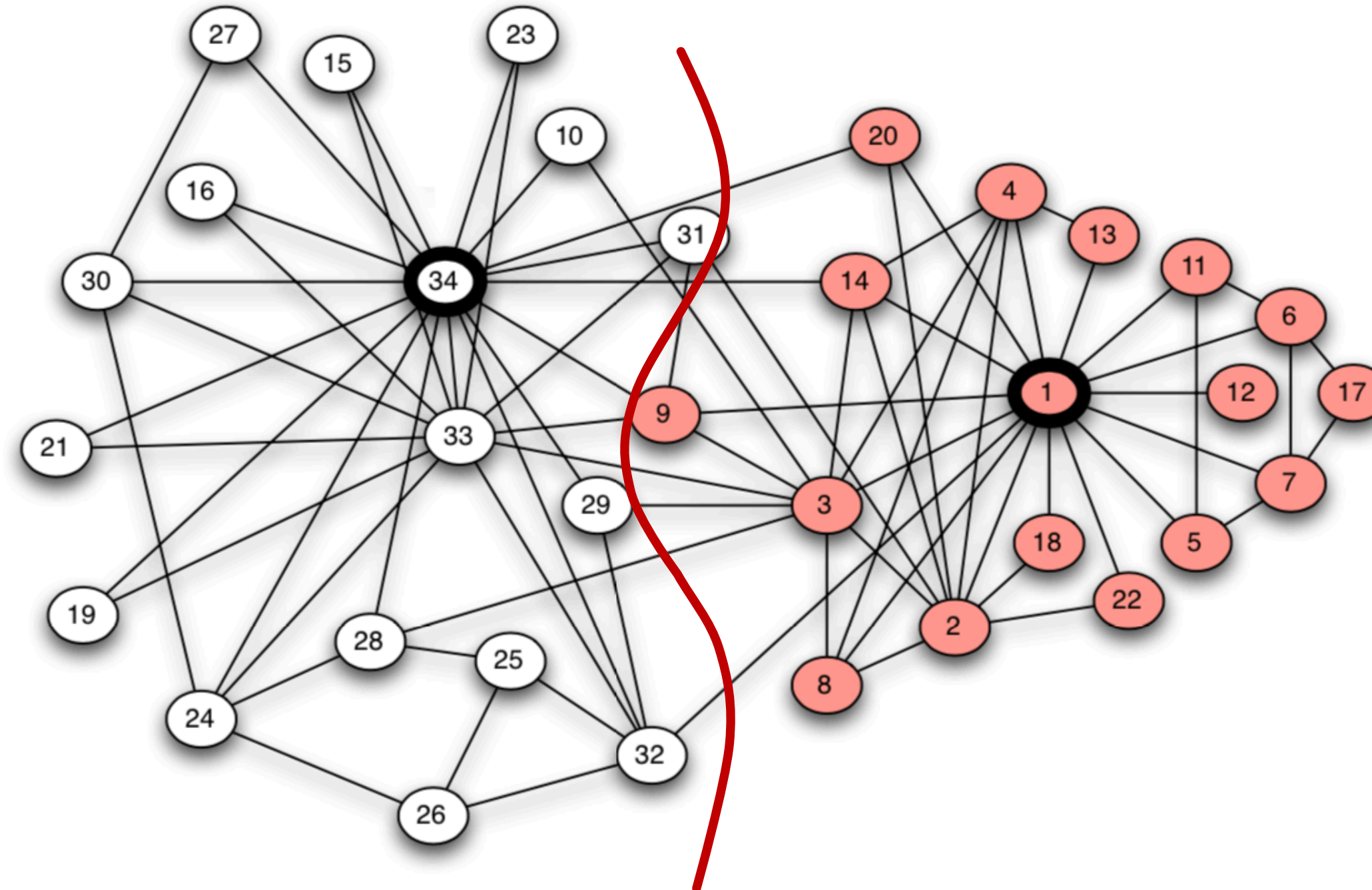
## Zachary's Karate club network:

Observe social ties and rivalries in a university karate club

During his observation, conflicts led the group to split

Split could be explained by a minimum cut in the network

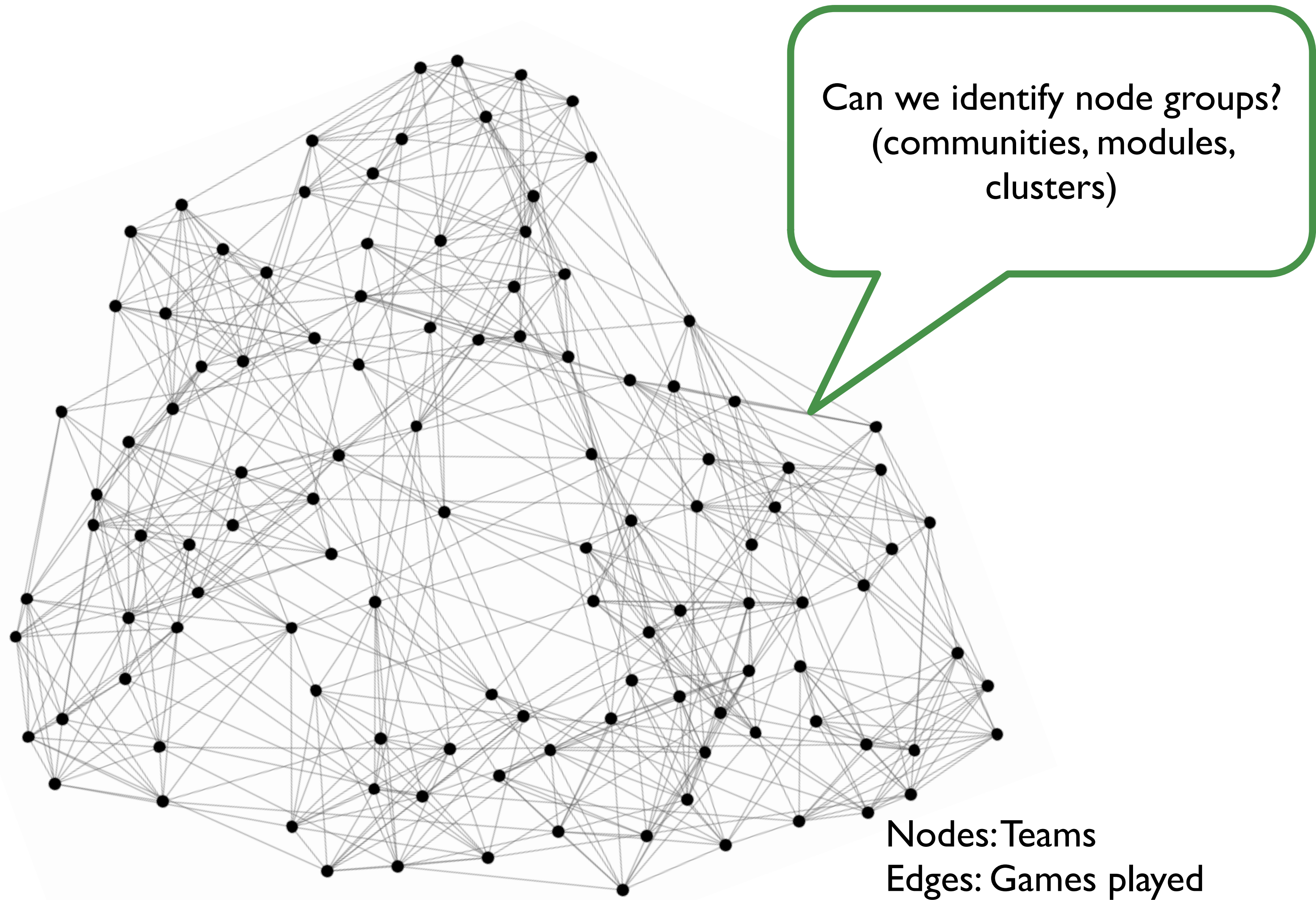
34: president



1: instructor

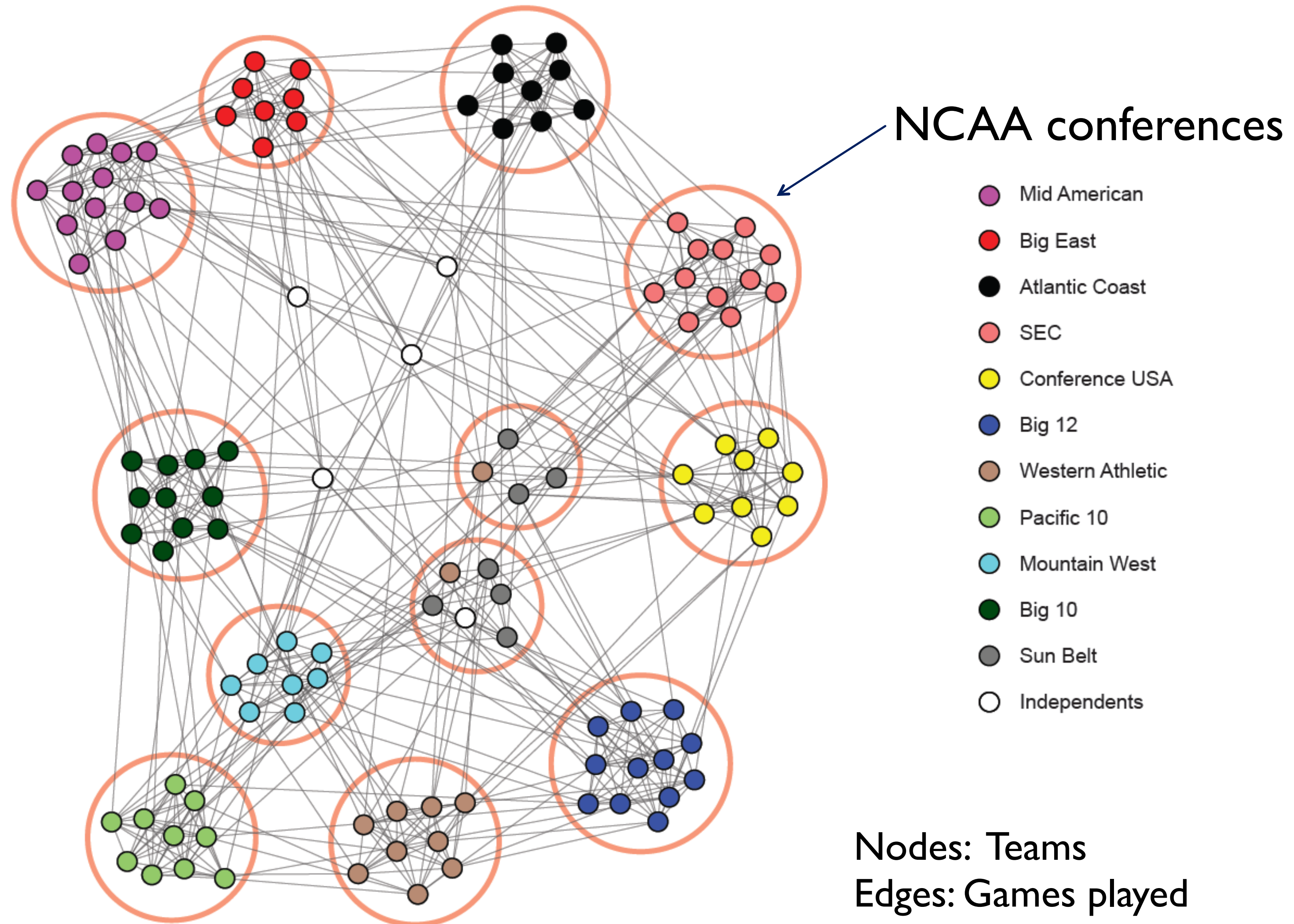


# NCAA Football Network



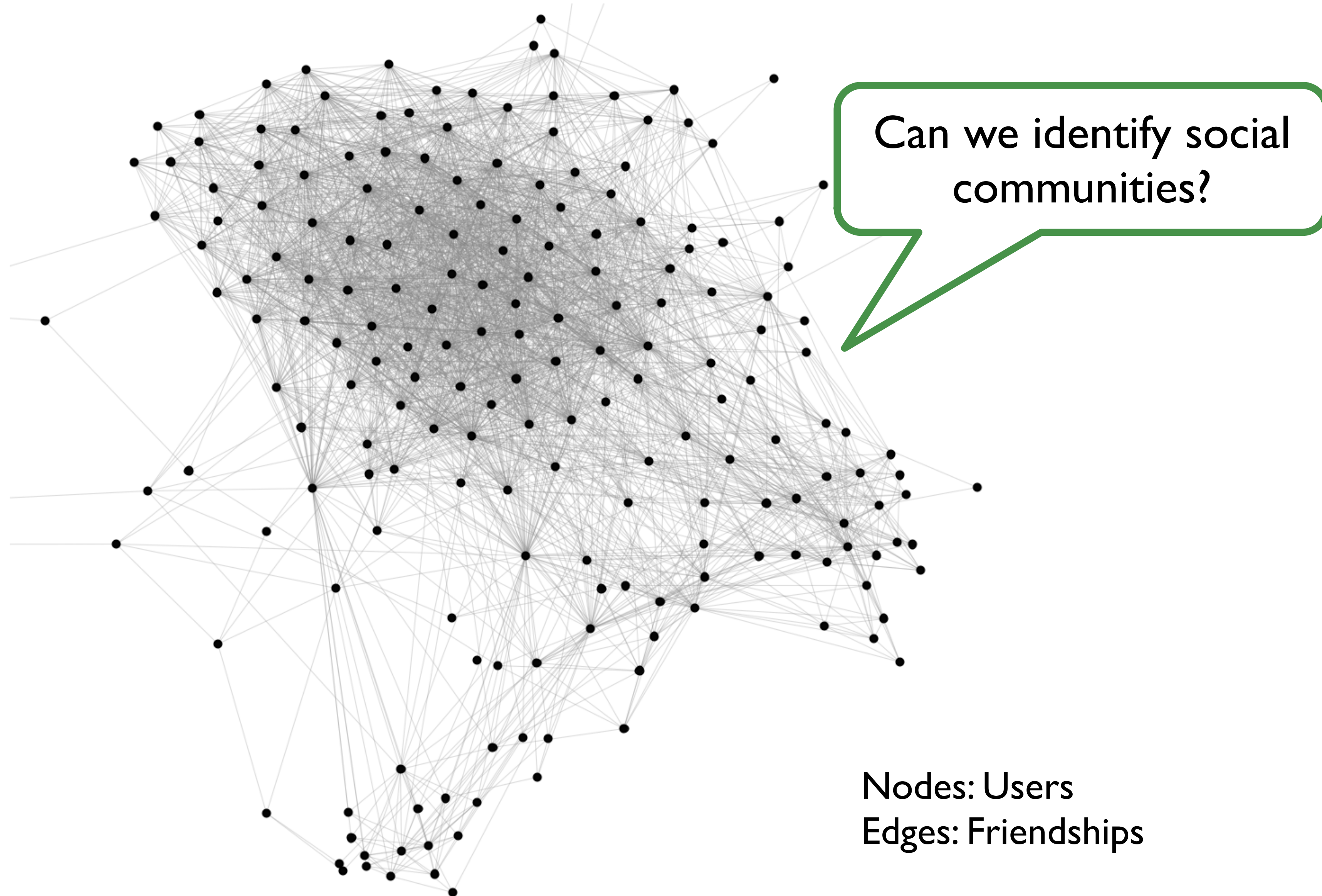


# NCAA Football Network



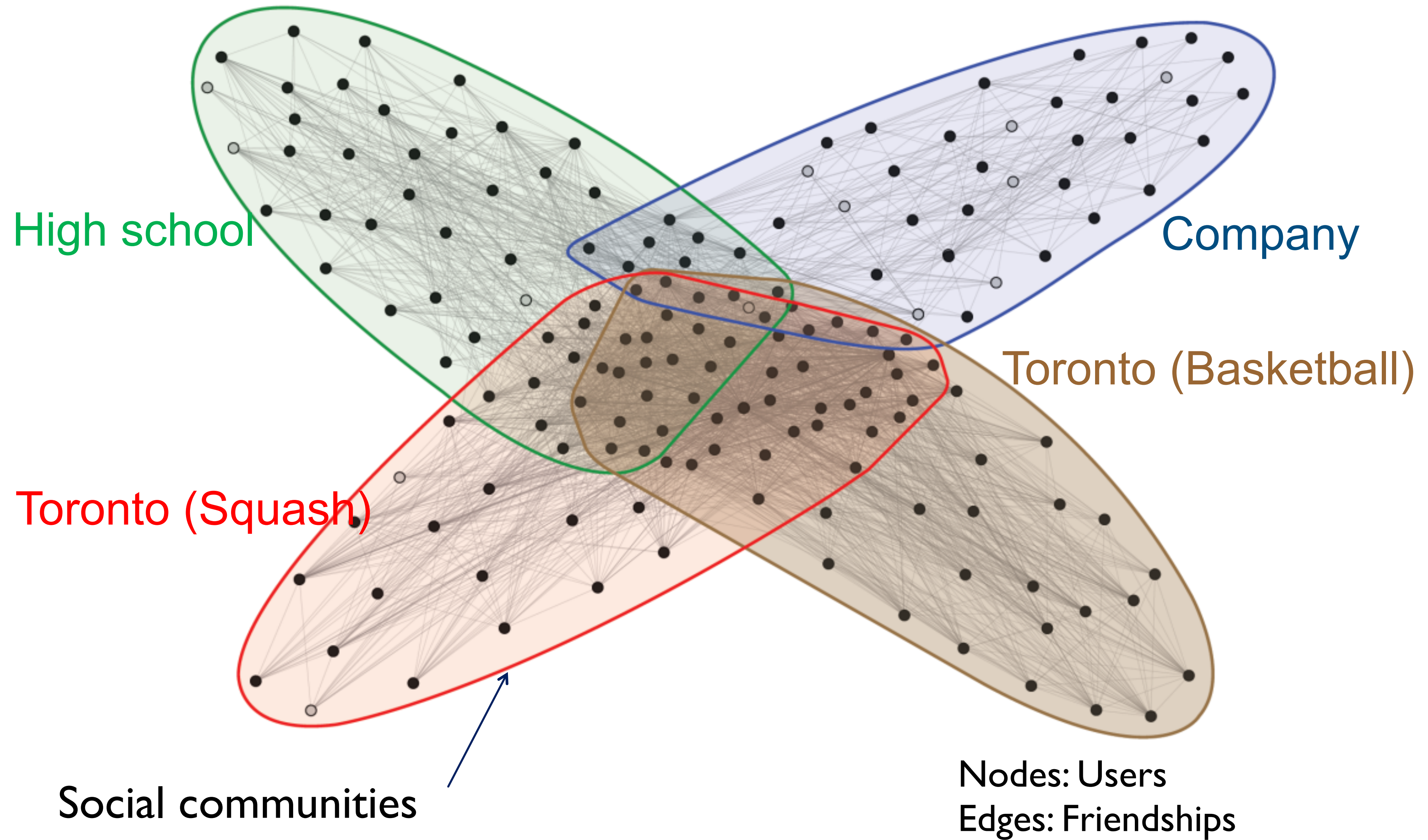


# Facebook Ego-network





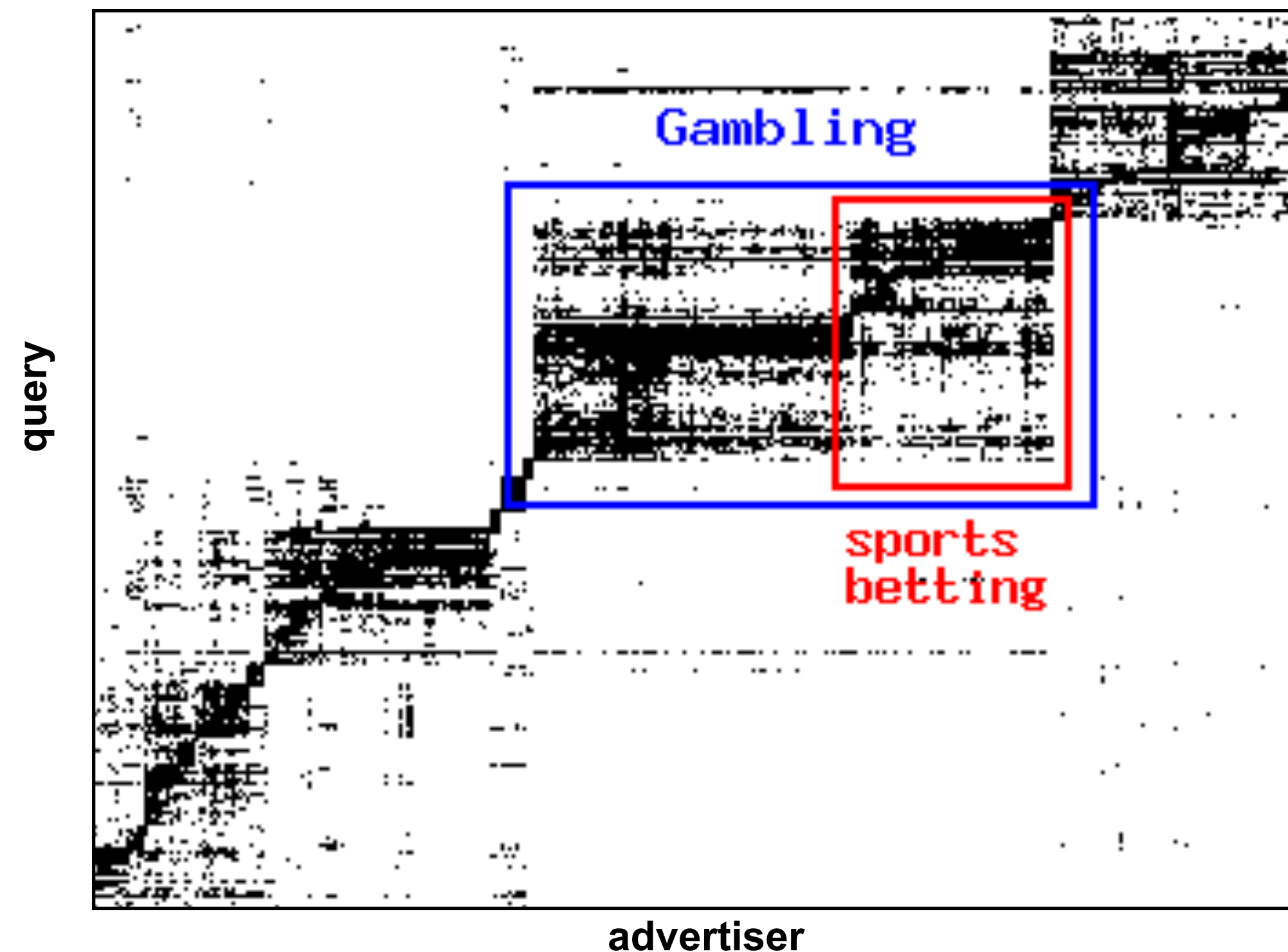
# Facebook Ego-network





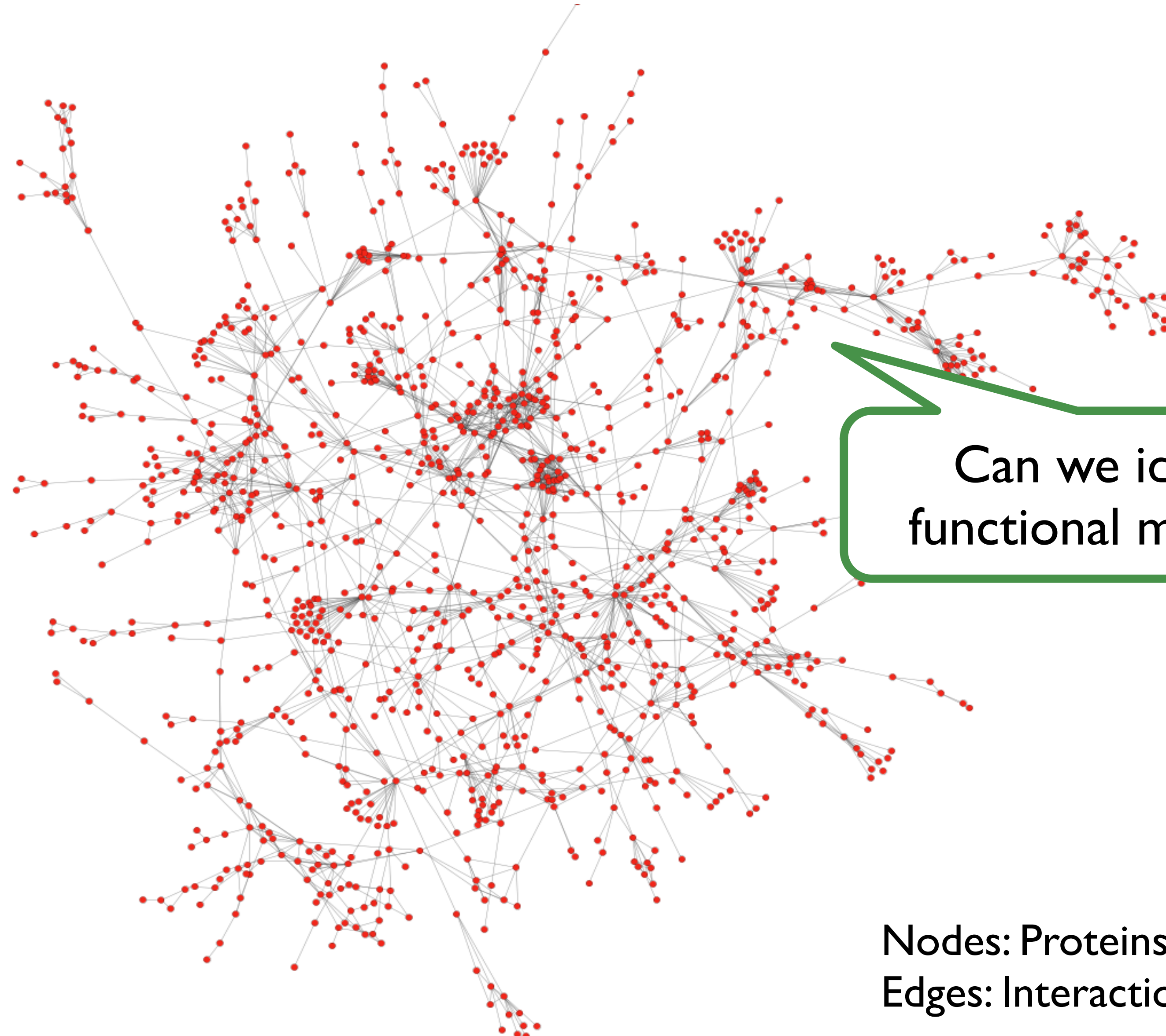
# Micro-Markets in Sponsored Search

Find micro-markets by partitioning the  
“query x advertiser” graph:



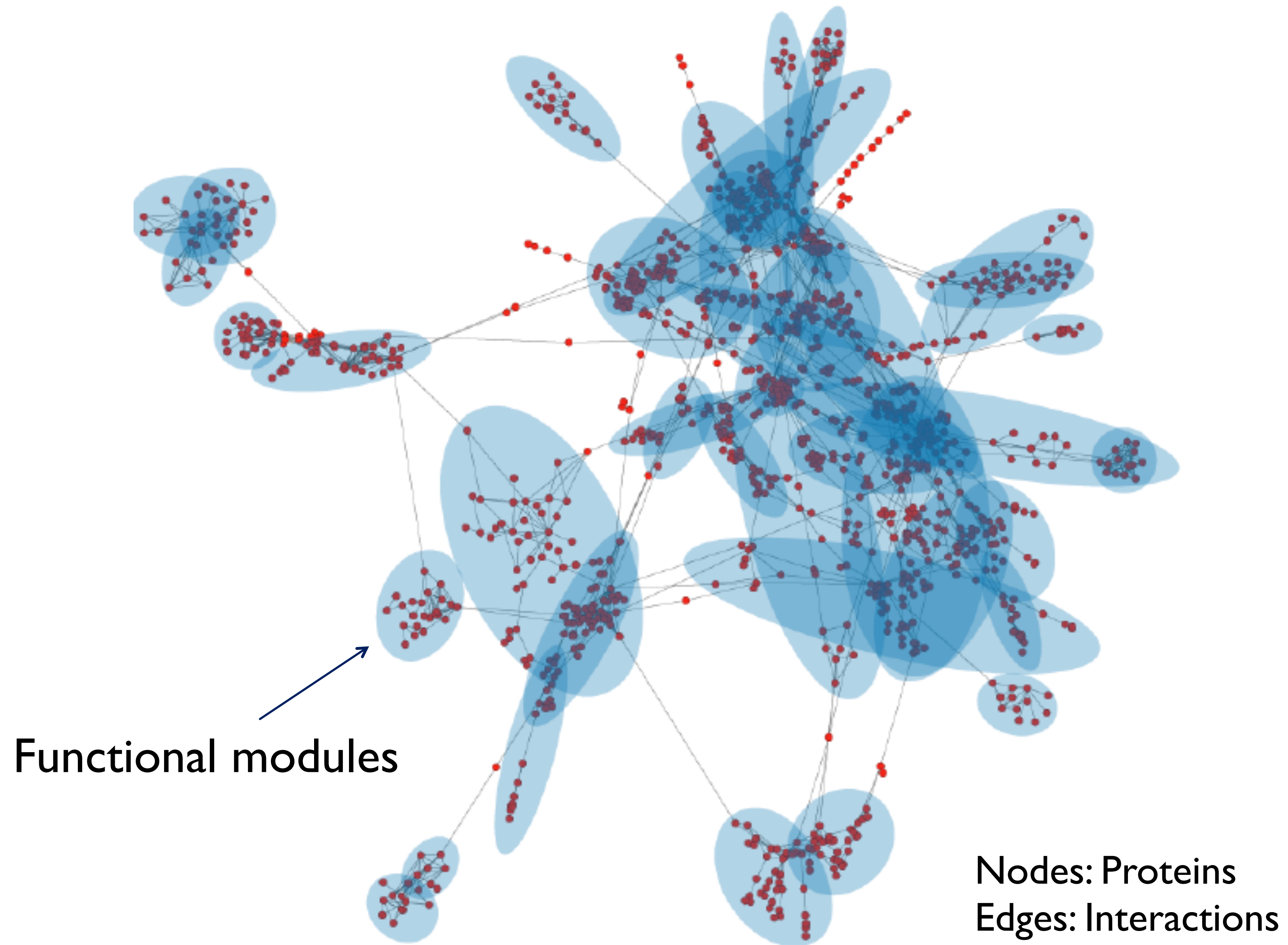


# Protein-Protein Interactions



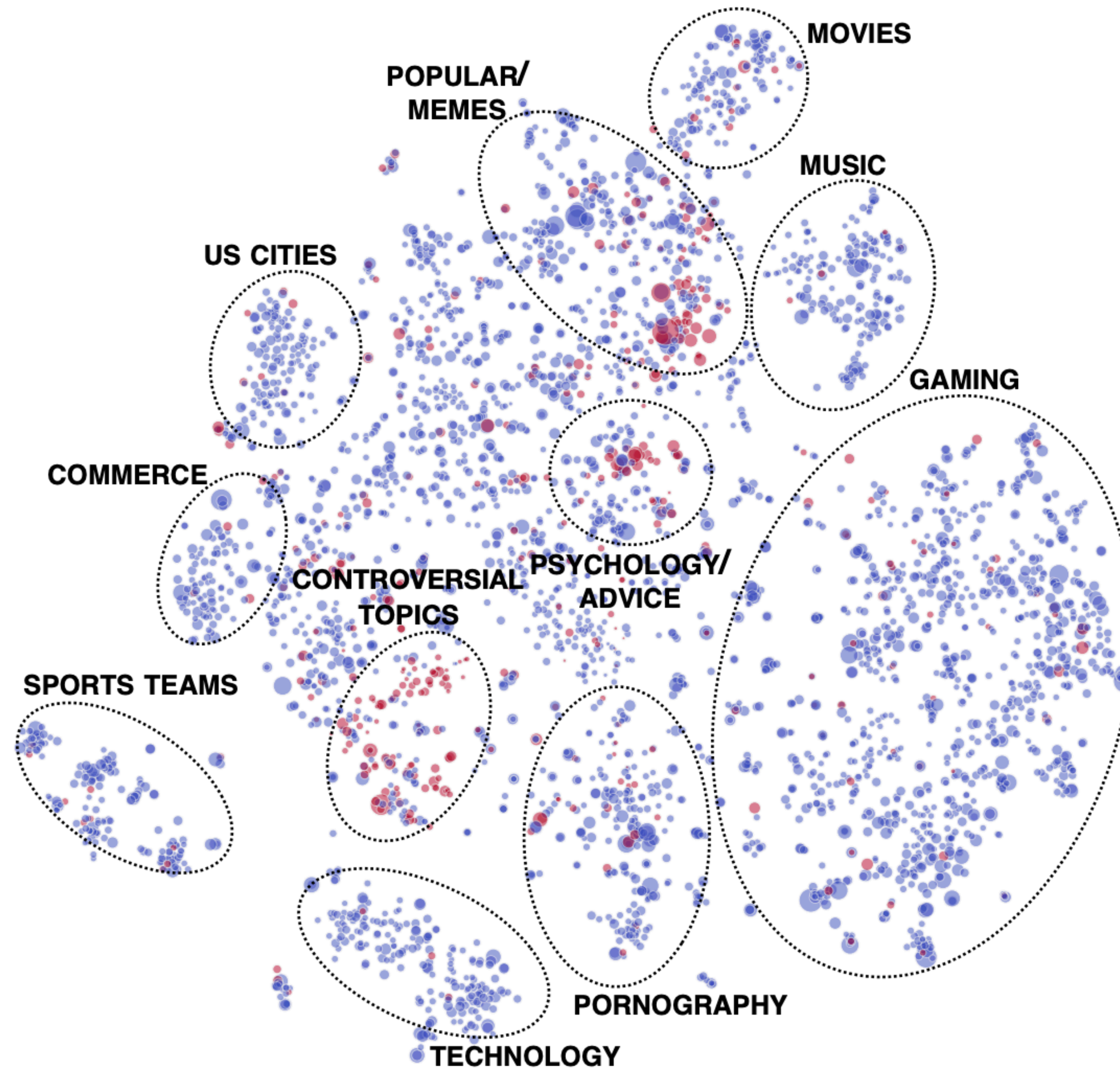
Nodes: Proteins  
Edges: Interactions

# Protein-Protein Interactions





# Community Structure on Reddit





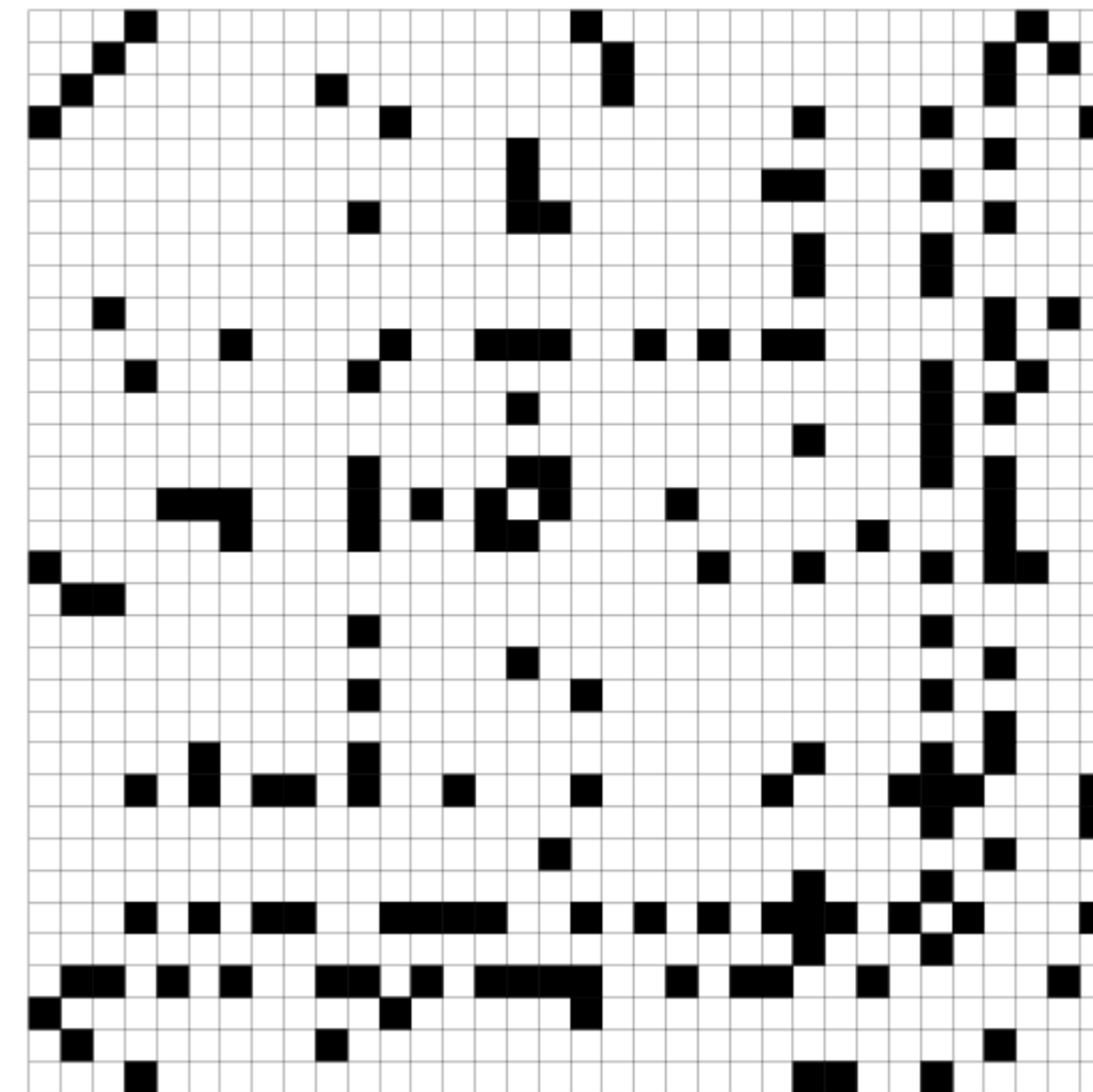
# Community Structure

Many real-world networks exhibit community structure that is “obvious” to the naked eye

**But what about finding communities from data?**

```
There is an edge between 0 and 1.  
There is an edge between 0 and 1.  
There is an edge between 0 and 0.  
There is an edge between 1 and 1.  
There is an edge between 1 and 1.  
There is an edge between 1 and 0.  
There is an edge between 2 and 3.  
There is an edge between 2 and 3.  
There is an edge between 2 and 2.  
There is an edge between 3 and 3.  
There is an edge between 3 and 3.  
There is an edge between 3 and 2.  
There is an edge between 4 and 4.
```

What are the communities now?



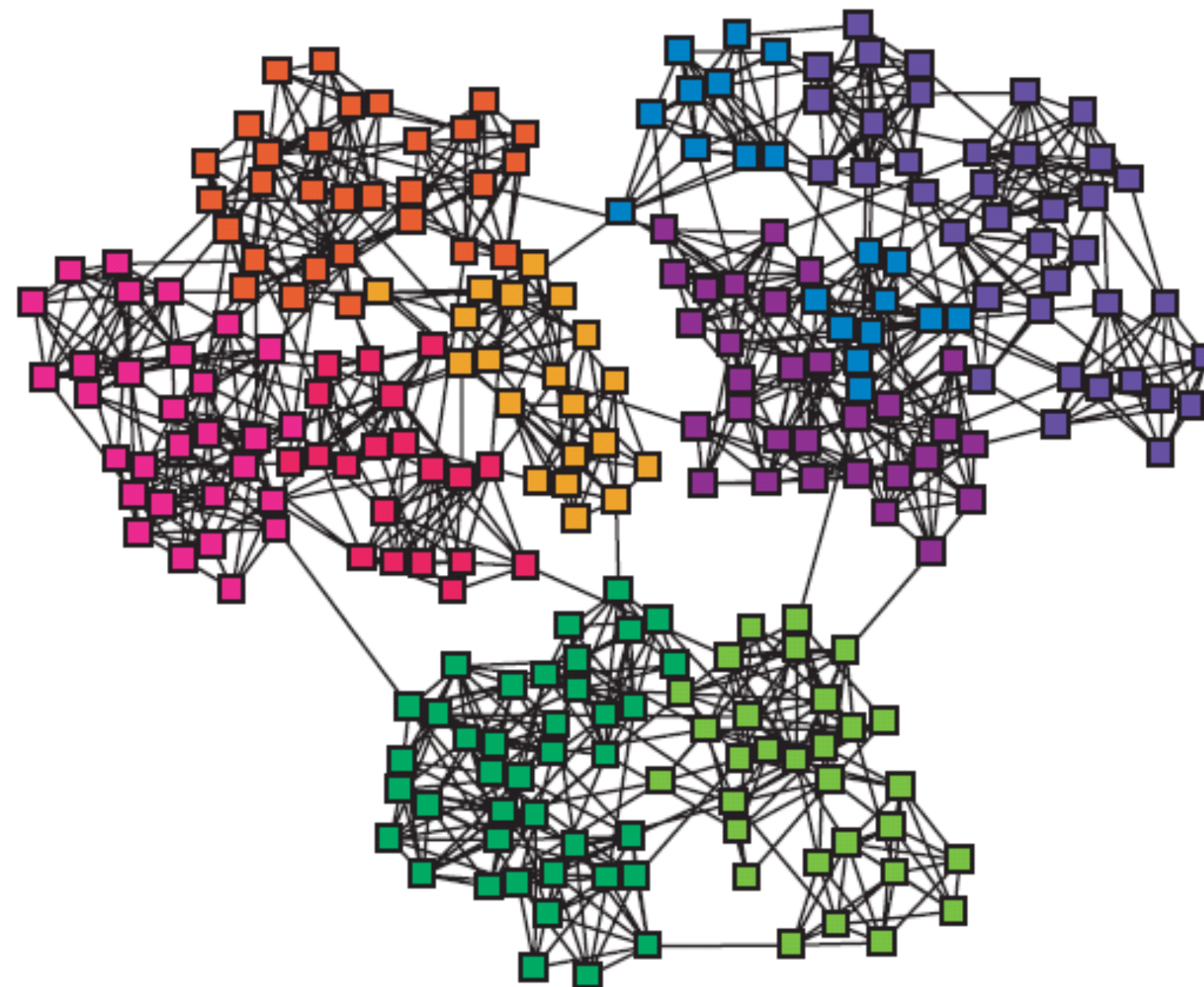
Do this with IB edges...

# Finding Network Communities

How to automatically find such densely connected groups of nodes?

Ideally such automatically detected clusters would then correspond to real groups

For example:

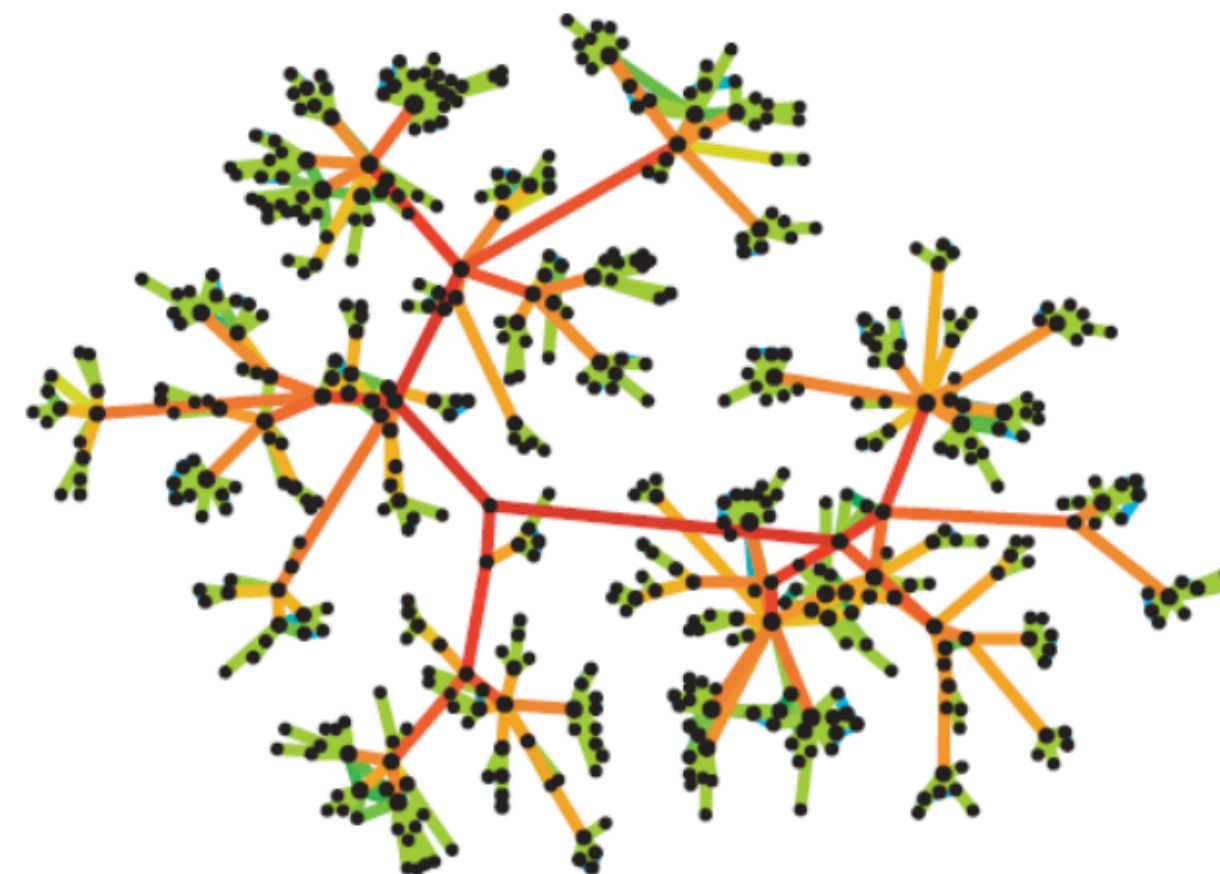
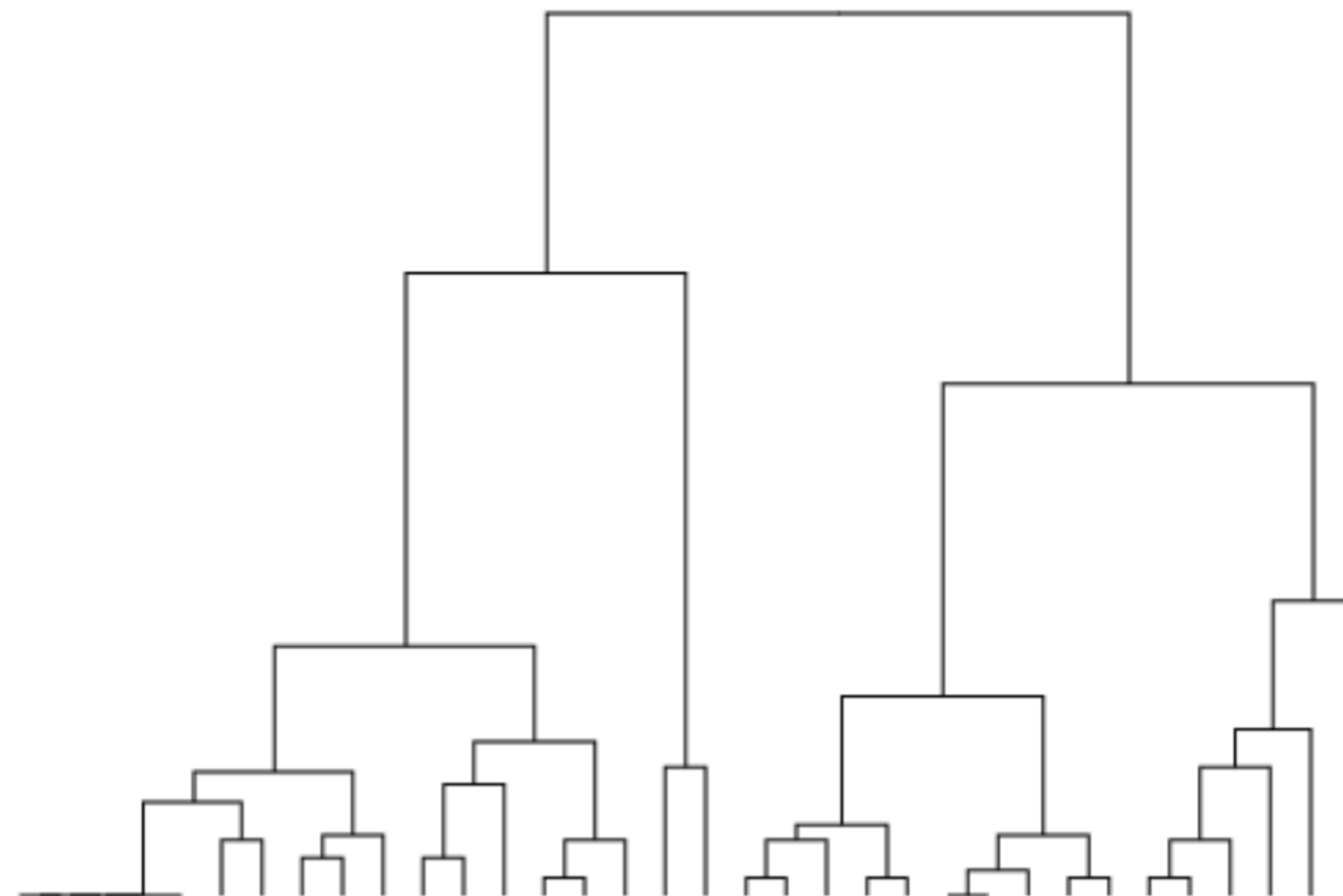


Note: We will work with undirected (unweighted) graphs

# Graph Partitioning

## Two general approaches:

1. Start with every node in the same cluster and break apart at “weak links” (“**divisive clustering**”)
2. Start with every node in its own “community” and join communities that are close together (“**agglomerative clustering**”)

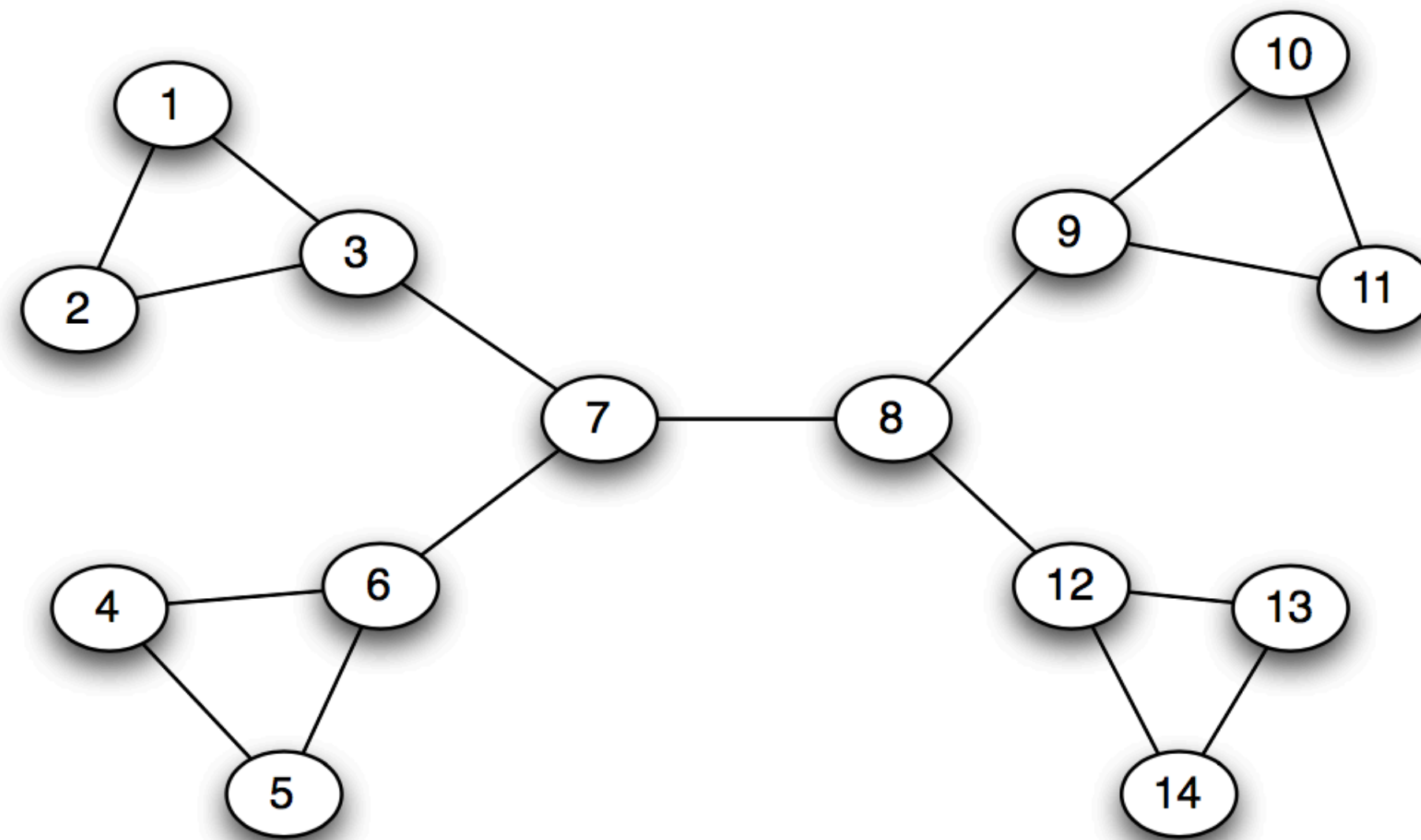




# Graph Partitioning

We'll do the first: **start with the whole graph as a community and recursively split it up into smaller communities**

Consider the following graph:

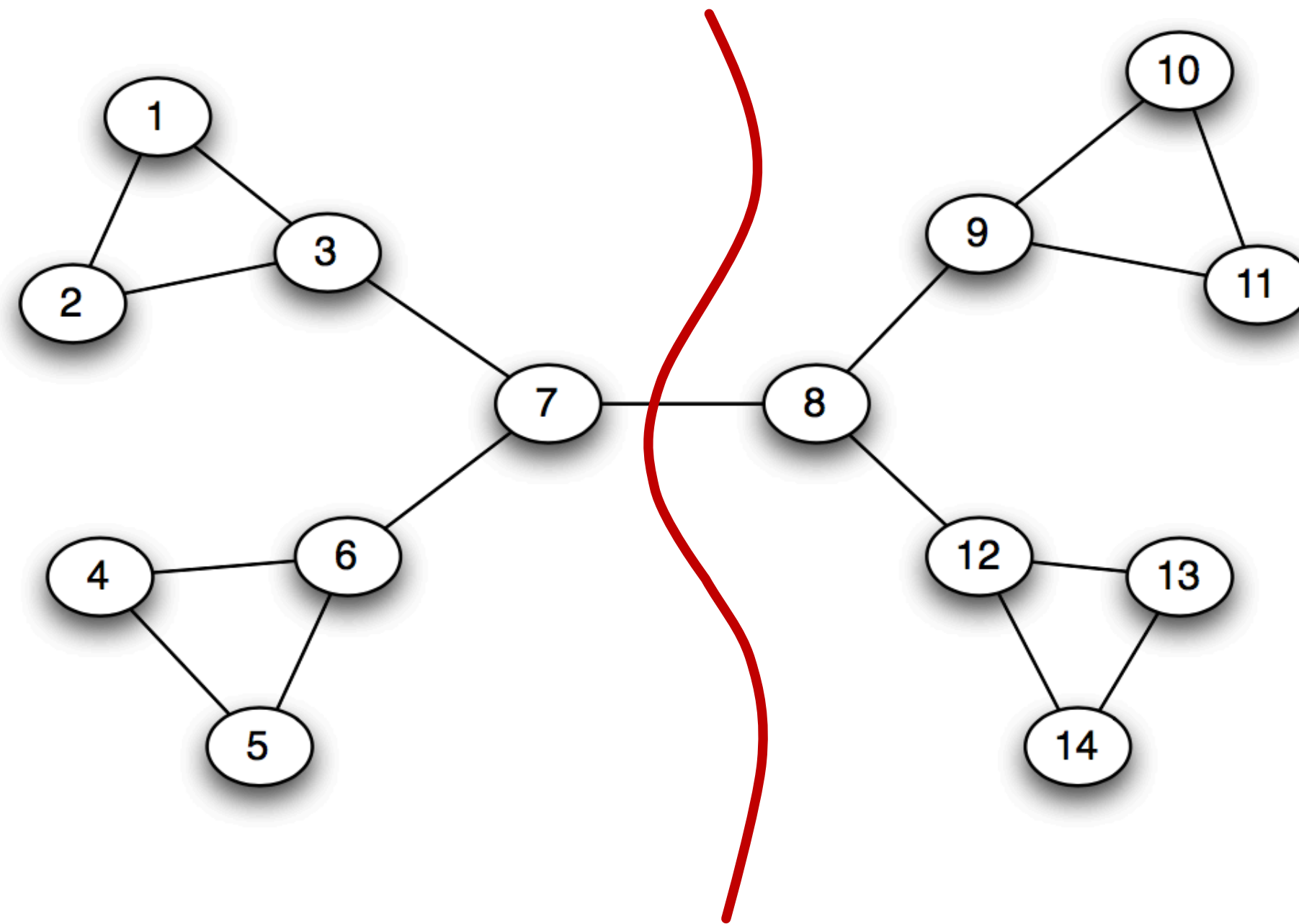


Where would you make the first cut?

# Graph Partitioning

We'll do the first: start with the whole graph as a community and recursively split it up

Consider the following graph:

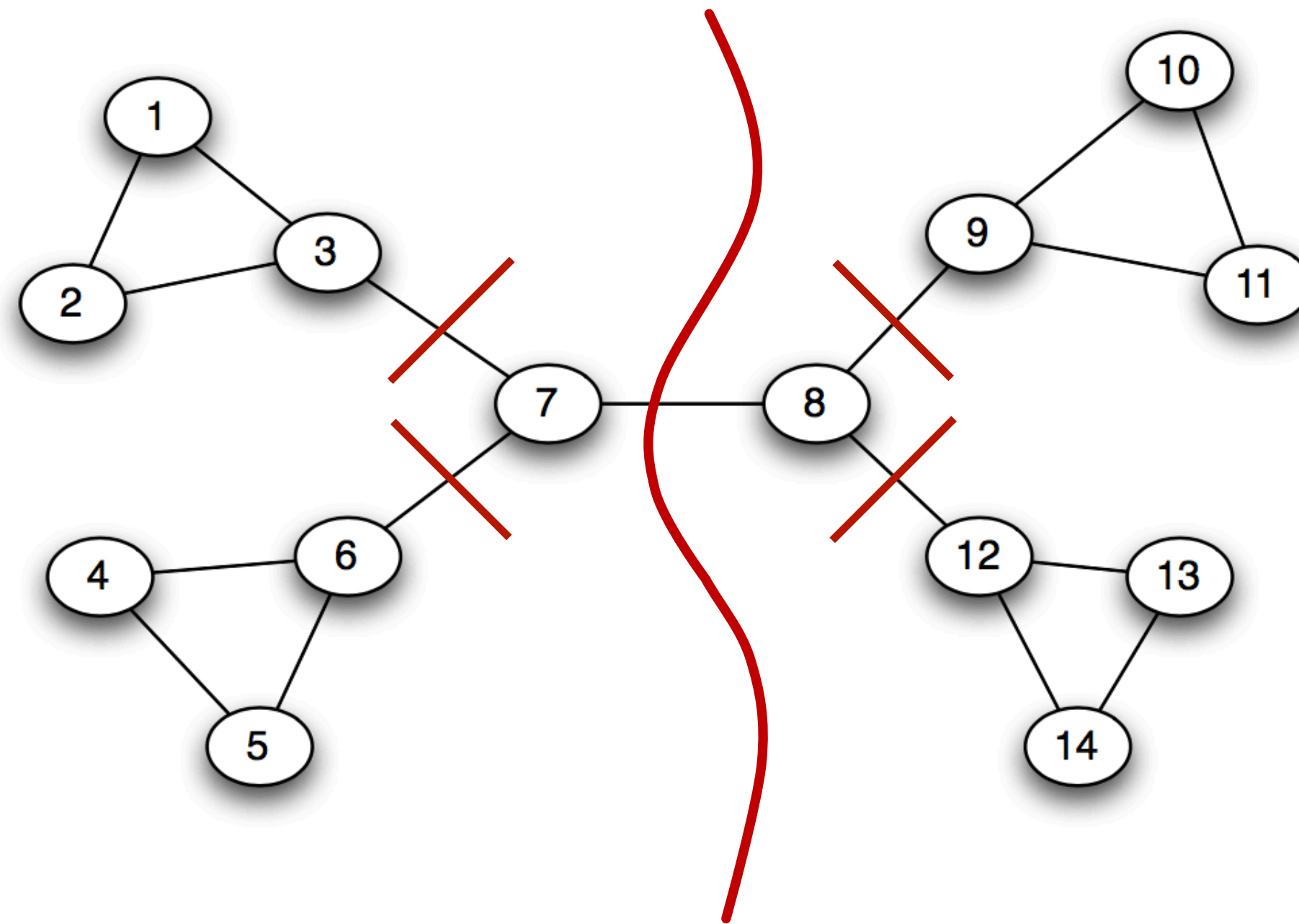


And now?

# Graph Partitioning

We'll do the first: start with the whole graph as a community and recursively split it up

Consider the following graph:

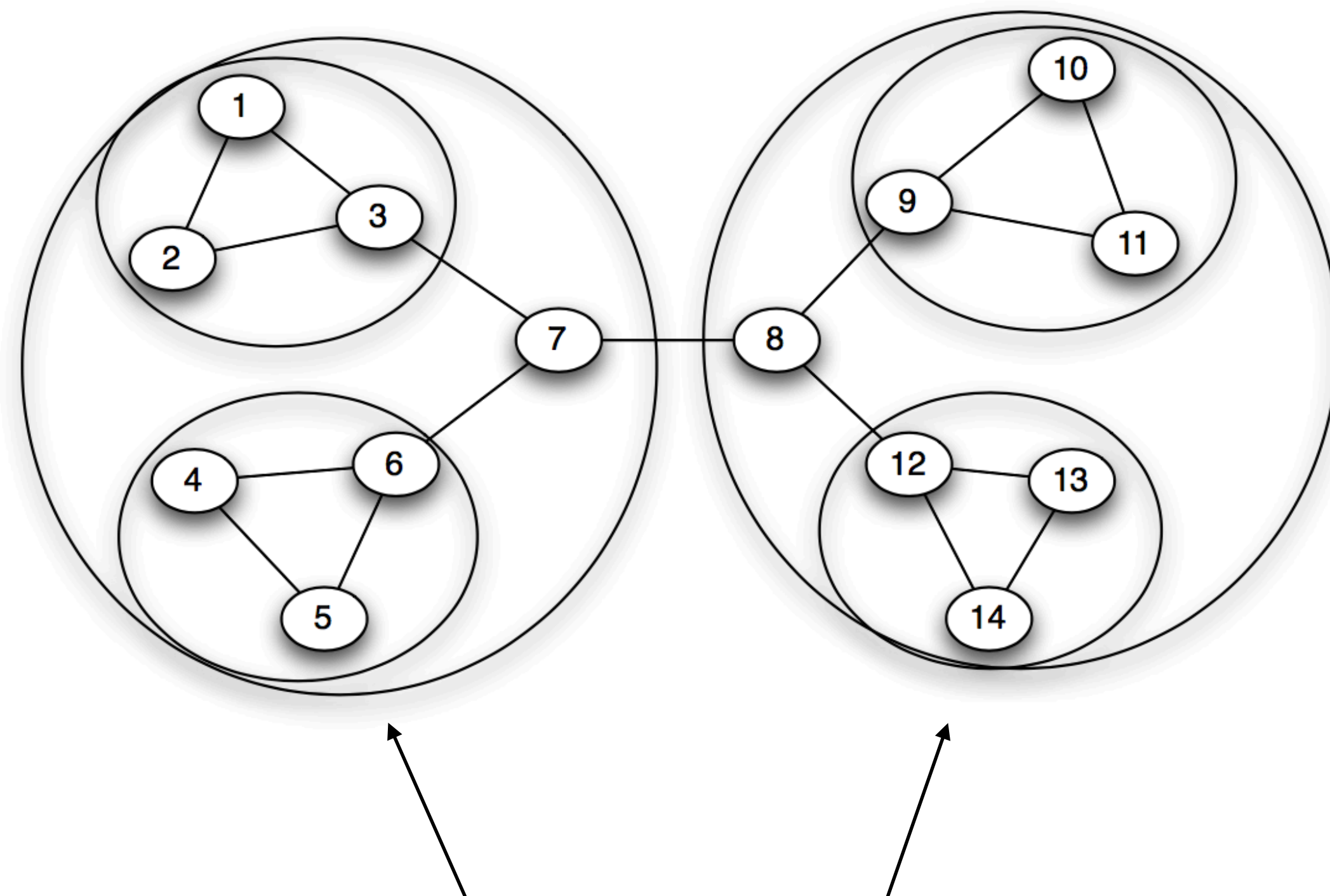




# Graph Partitioning

We'll do the first: start with the whole graph as a community and recursively split it up

Consider the following graph:



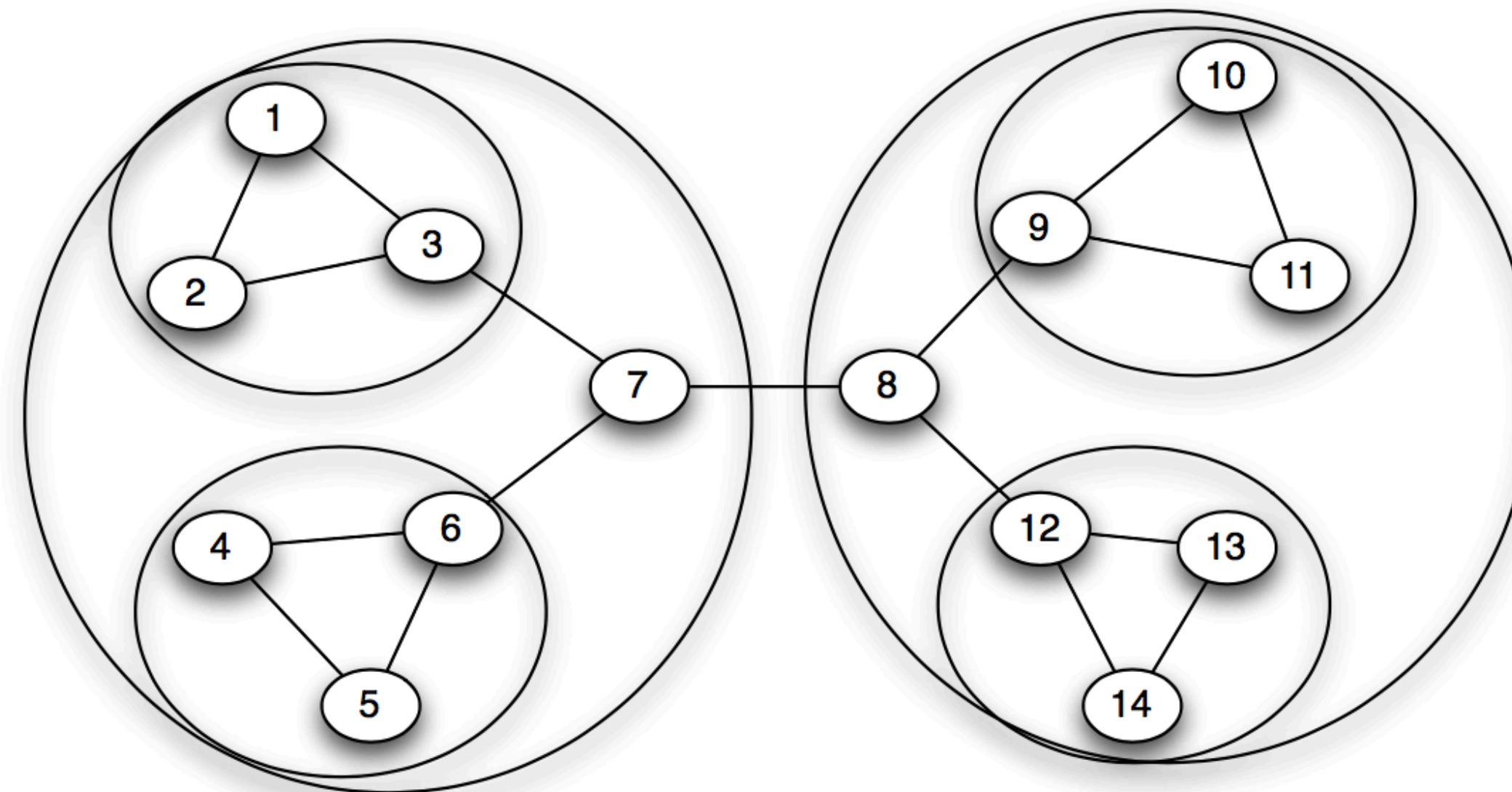
**Tightly-knit regions**

# Graph Partitioning

This naturally produces **nested communities**

This is familiar from everyday life:

- Countries, provinces, cities...
- Sports, Arts, Business then teams, art forms, sectors

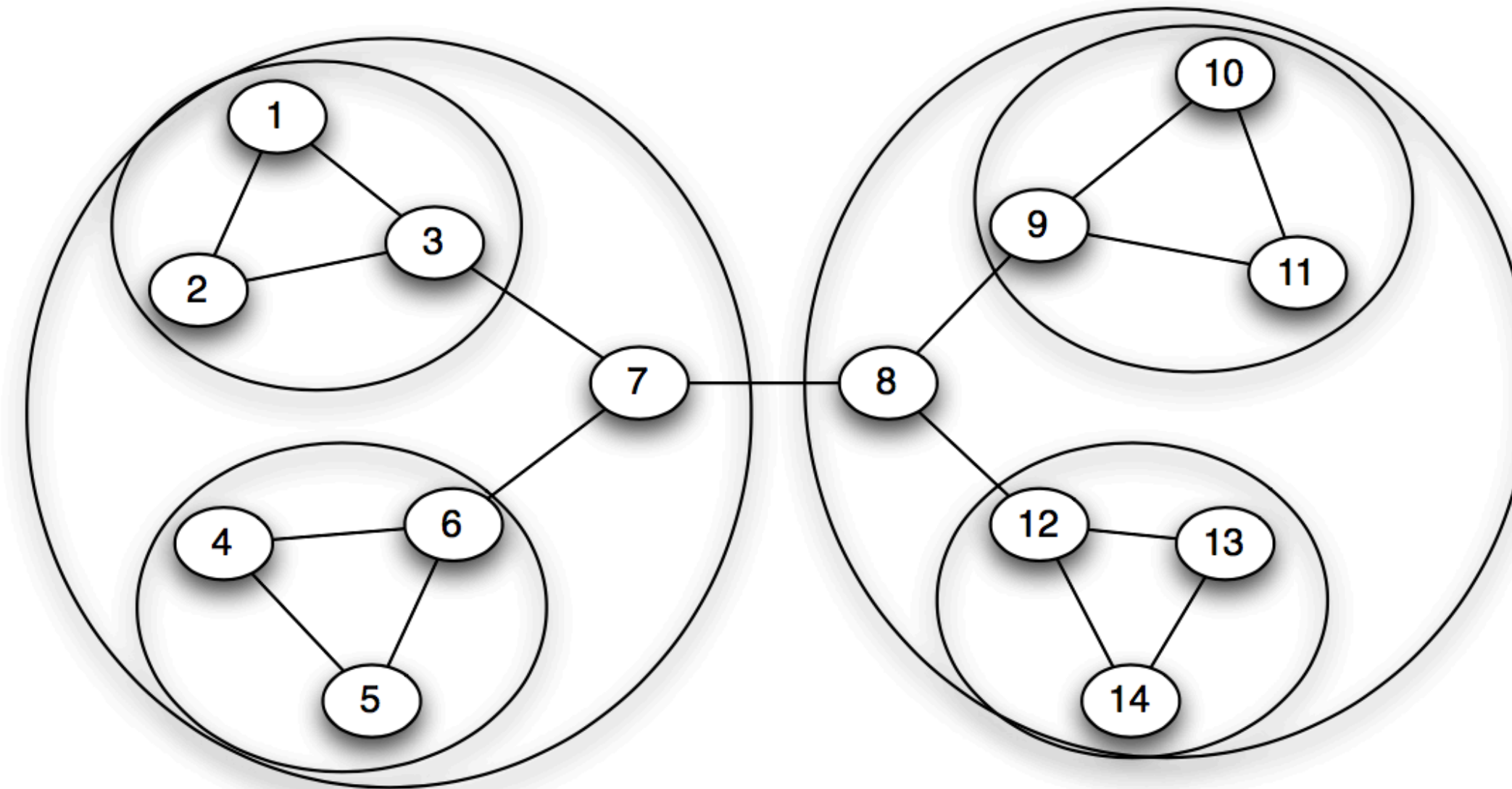


**Nested  
structure!**

# Graph Partitioning

A number of **both** agglomerative and divisive clustering methods will **find this partitioning**

- Divisive will **delete 7-8 first**, etc.
- Agglomerative would **add 7-8 last**, etc.



**Nested  
structure!**

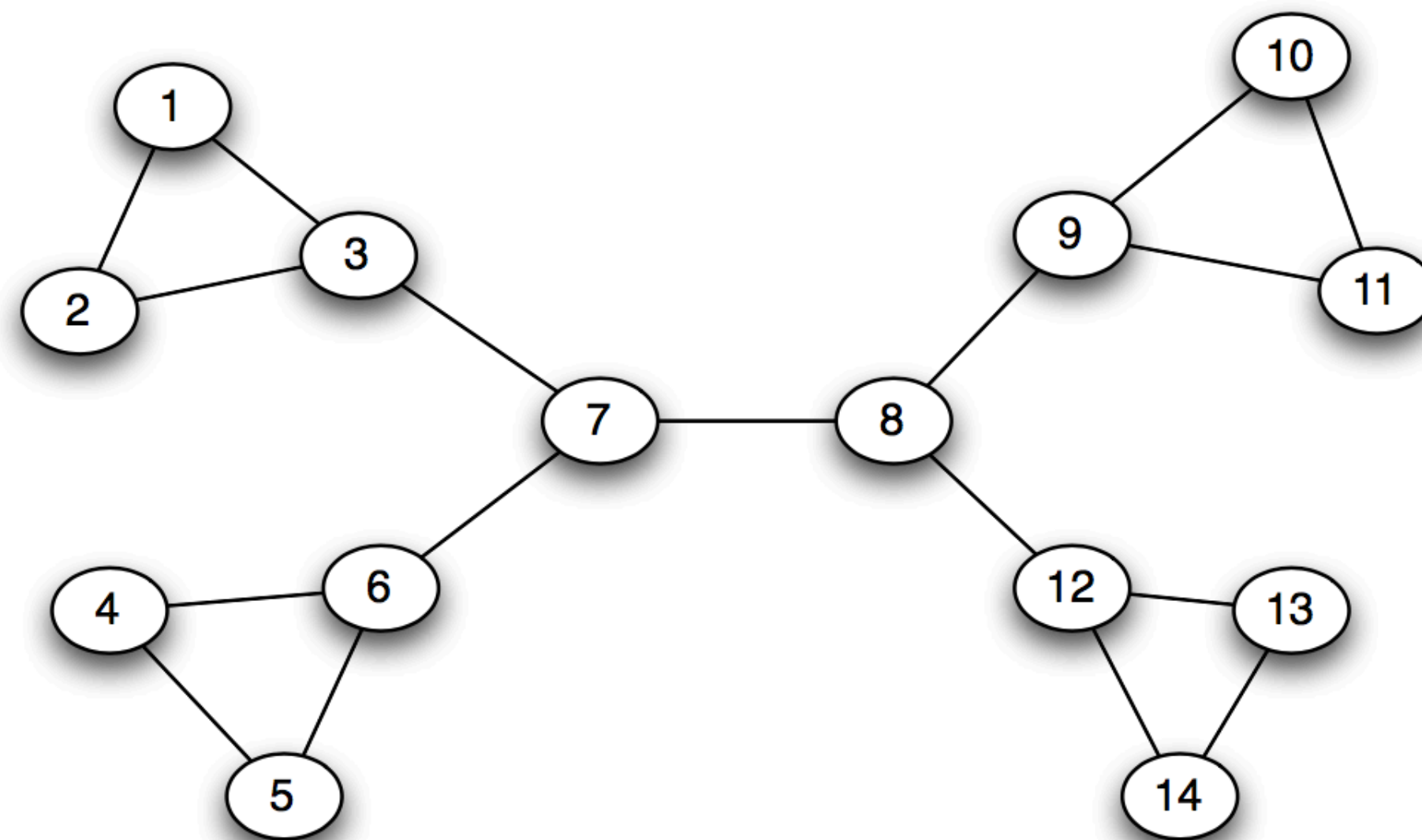


# Graph Partitioning

Back to divisive clustering: Why is 7-8 a good candidate for the first cut?

It is a **bridge**

Recall that a weak tie is defined as an edge that separates weakly-conne

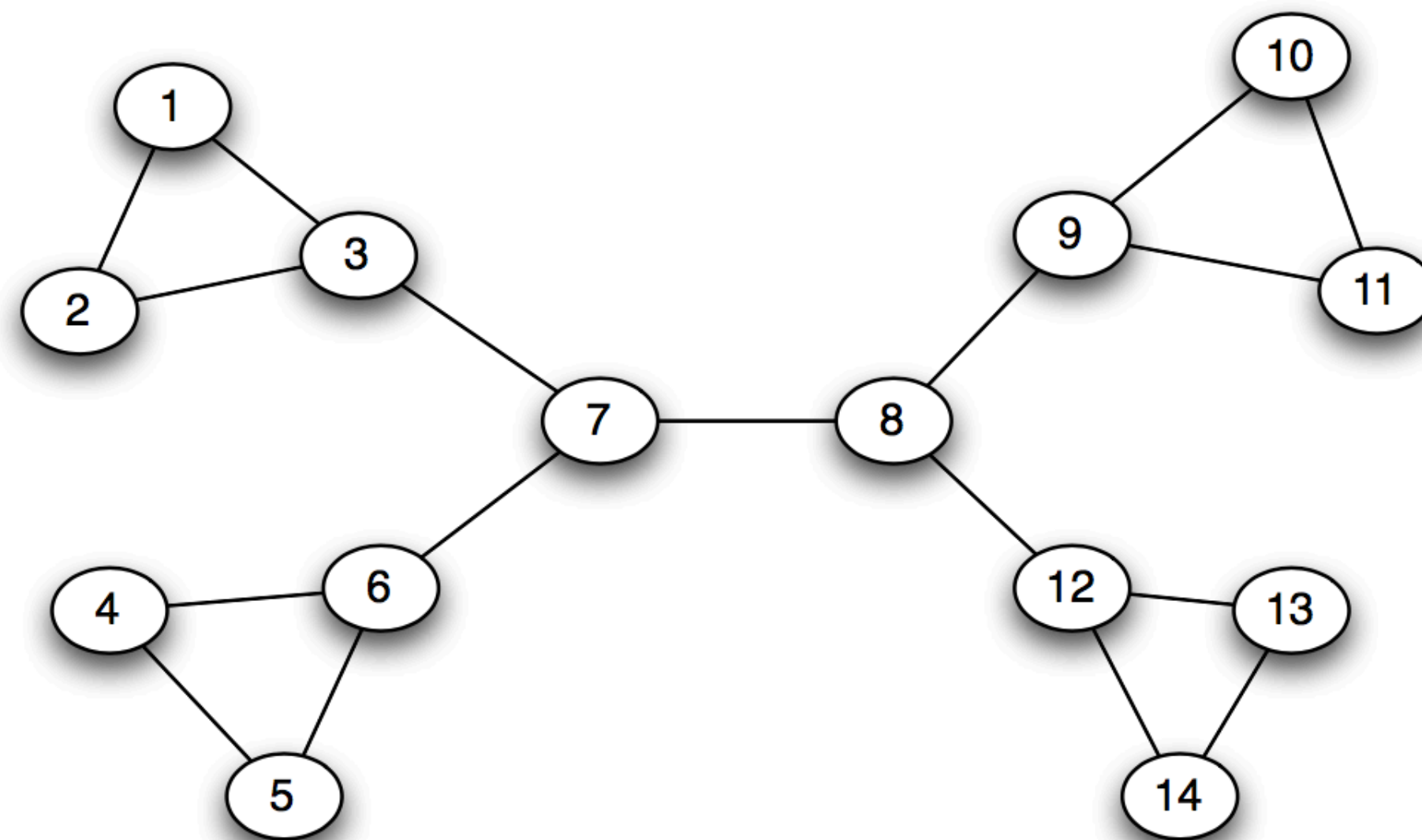


# Graph Partitioning

Divisive clustering algorithm: Recursively remove bridges?

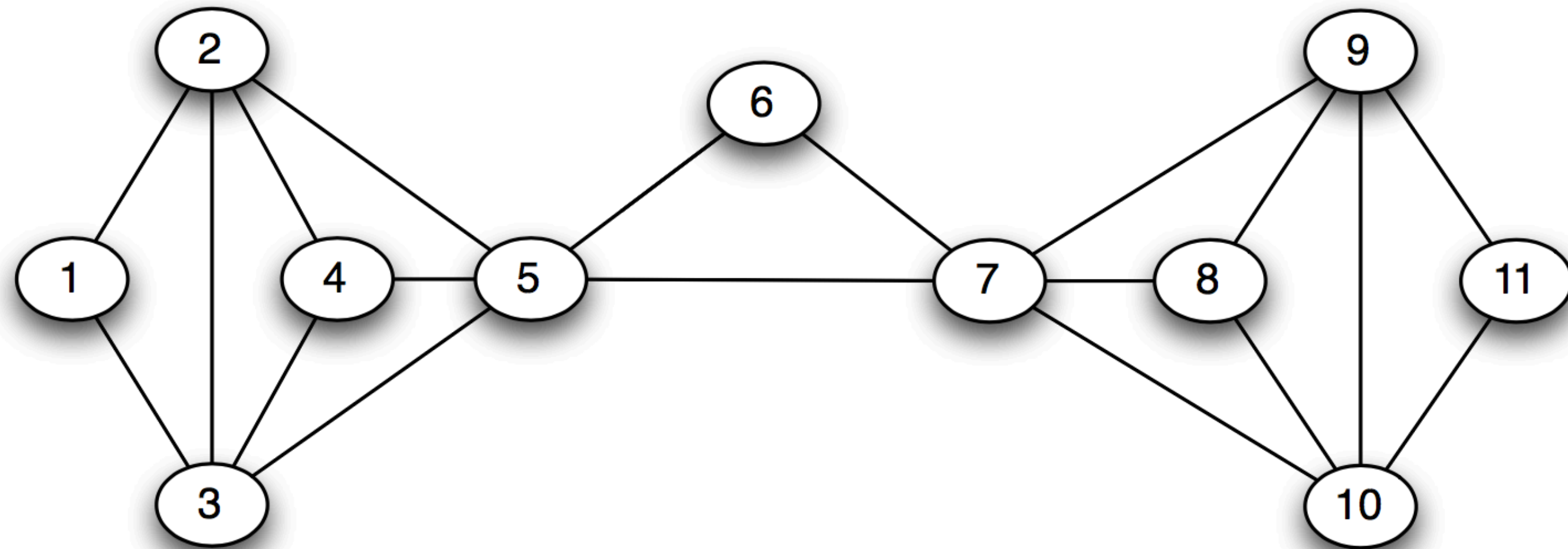
Right idea, but **not strong enough**: There are other bridges too (which ones?)

3-7, 6-7, 8-9, 8-12 are also bridges!



# Graph Partitioning

Also, sometimes there are **no bridges** (or even no local bridges) **but “natural” communities still exist**



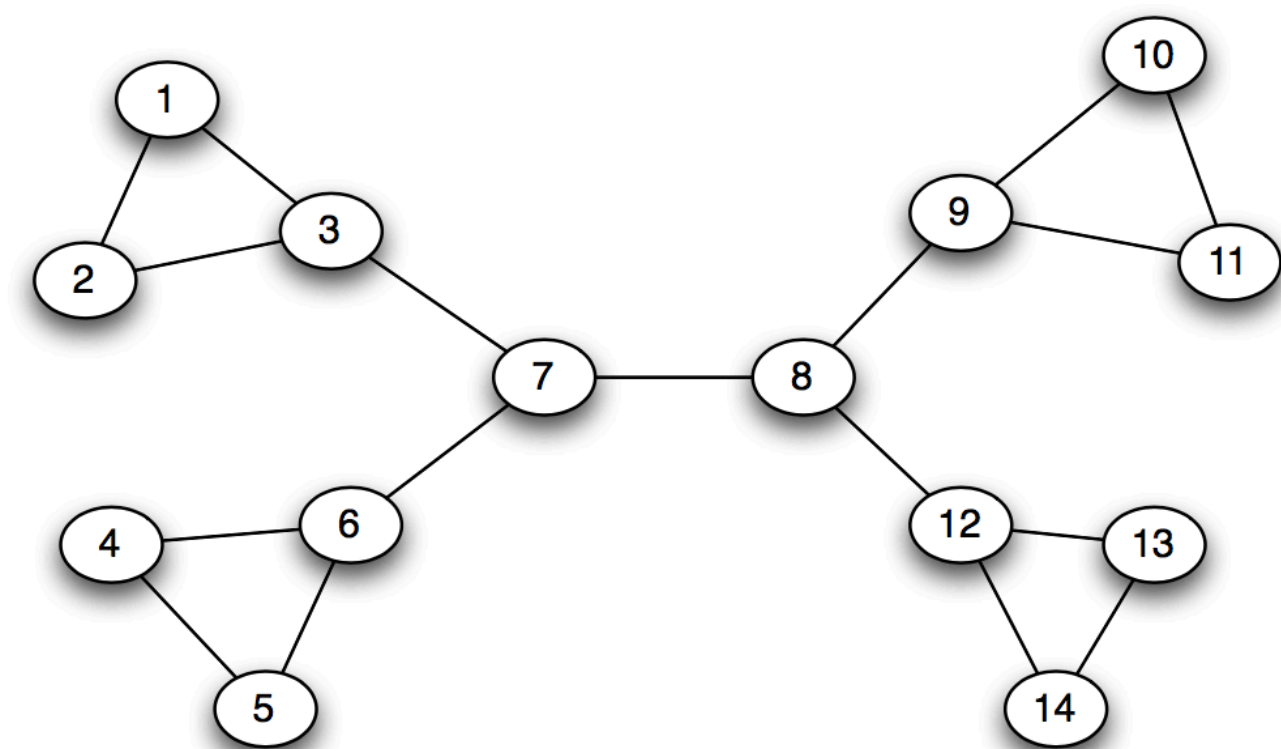
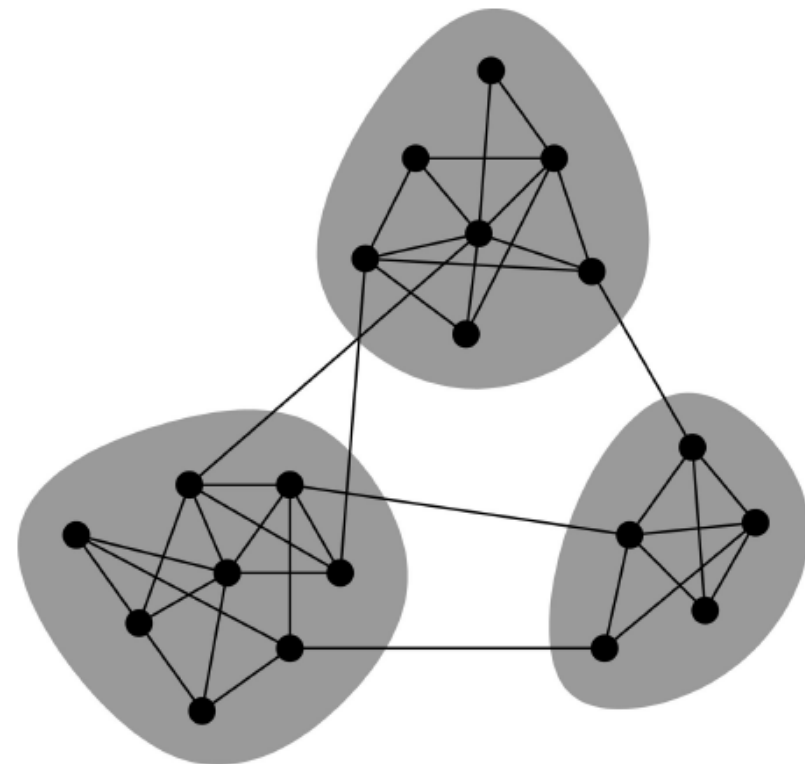


# Graph Partitioning

Recall definition of a **bridge**: an edge that, if you remove it, disconnects its endpoints

Thus it is **an edge that carries a shortest path**  
(obviously the shortest, since it's also the only path)

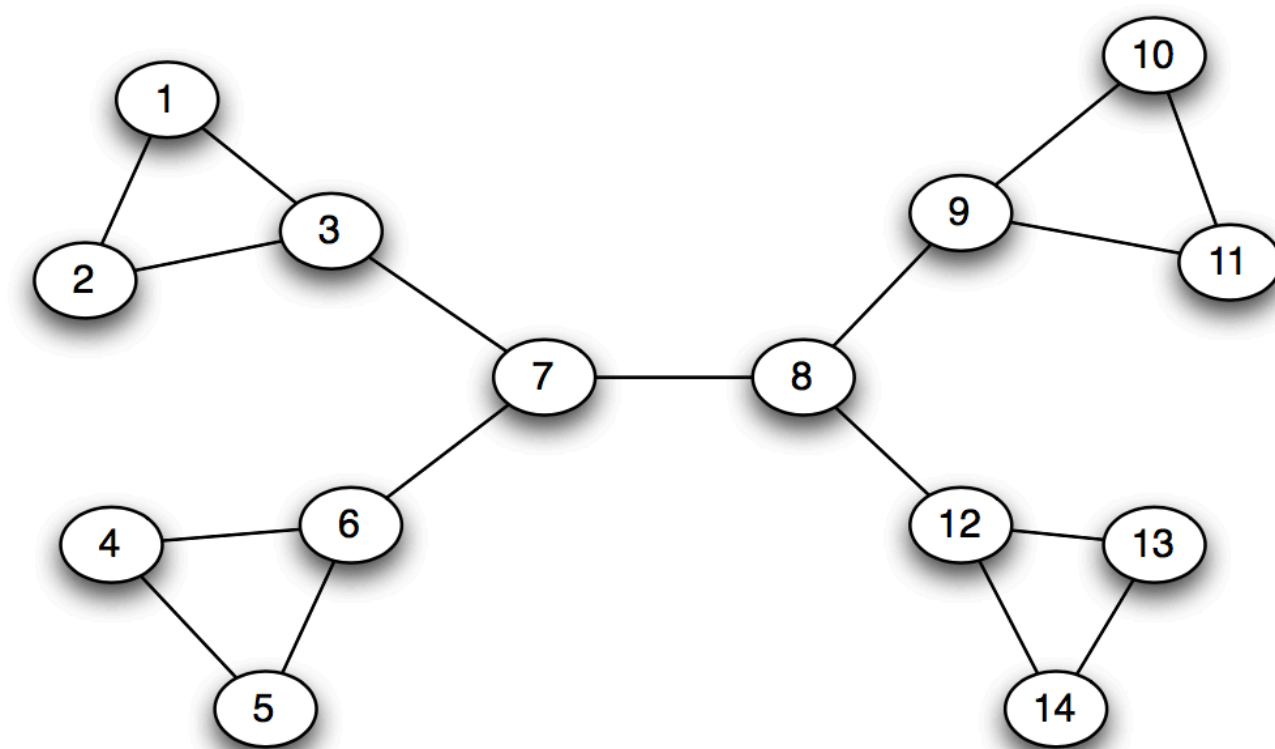
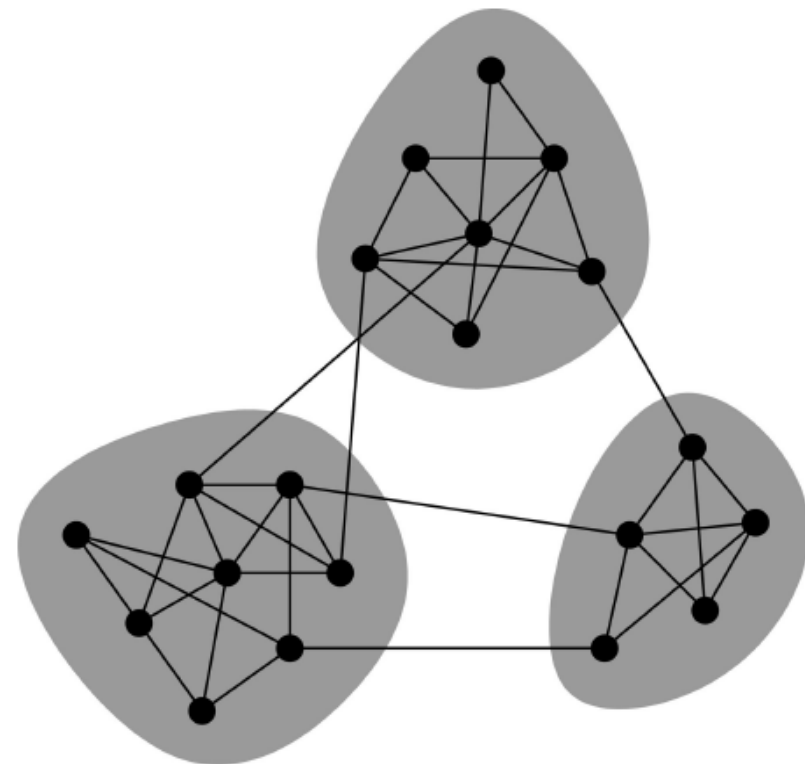
Need a **more nuanced definition** to distinguish bridges and  
“bridge-like” edges from highly embedded edges



# Graph Partitioning

**Definition:** the **betweenness** of an edge is how many (fractional) shortest paths travel through it

- For every pair of nodes A,B say there is one unit of “flow” along the edges from A to B
- Flow between A to B divides evenly among all shortest paths from A to B
- If k shortest paths,  $1/k$  flow on each path



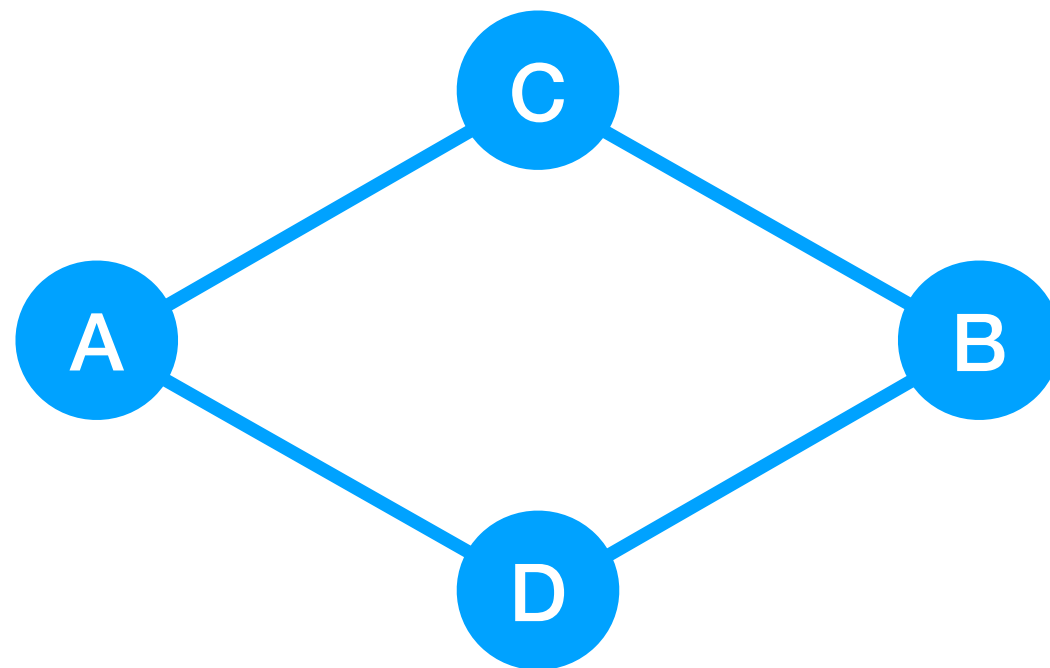
# Graph Partitioning



One unit of flow from A to B  
 $\text{Betweenness}(A-B) = 1$



# Graph Partitioning



One unit of flow from A to B

Two shortest paths from A to B, split evenly among them

So edges a–c, c–b, a–d, d–b get 1/2 flow each from the (A,B) pair

...and repeat for one unit of flow between every other pair of nodes:

(A,C), (A,D), (B,C), (B,D), (C,D)

# Girvan-Newman algorithm

Divisive hierarchical clustering based on the notion of edge **betweenness** (Number of shortest paths passing through an edge)

**Girvan-Newman Algorithm** (on undirected unweighted networks):

**Repeat until no edges are left:**

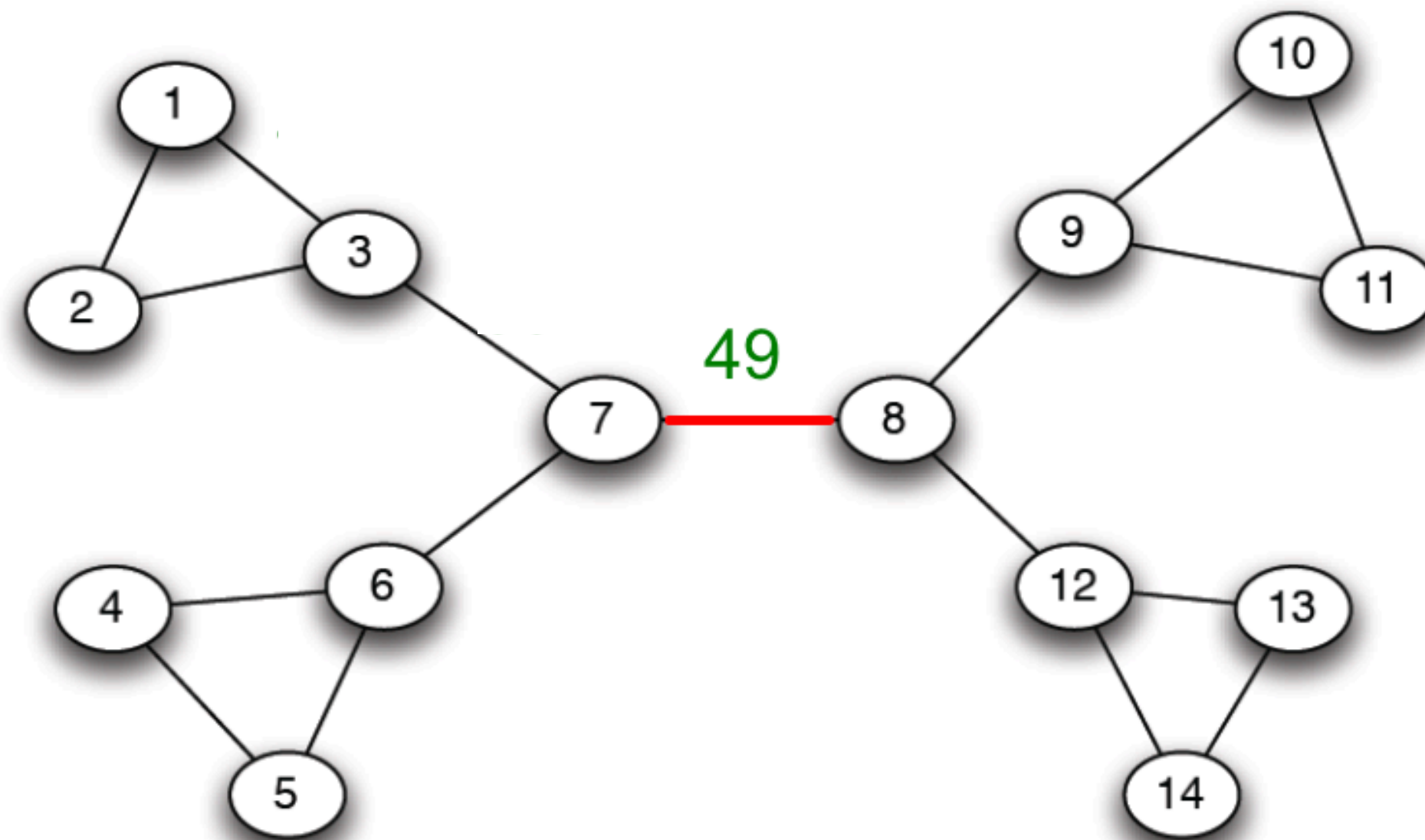
- (Re)calculate betweenness of every edge
- Remove edges with highest betweenness (if ties, remove all edges tied for highest)
- Connected components are communities

**Gives a hierarchical decomposition of the network**

# Girvan-Newman: Example

Consider edge **7-8**:

- Each node A on left and node B on right has shortest path passing through **7-8**
- No flow passing between nodes on same side passes through **7-8**
- Betweenness(**7-8**) =  $7 \times 7 = 49$





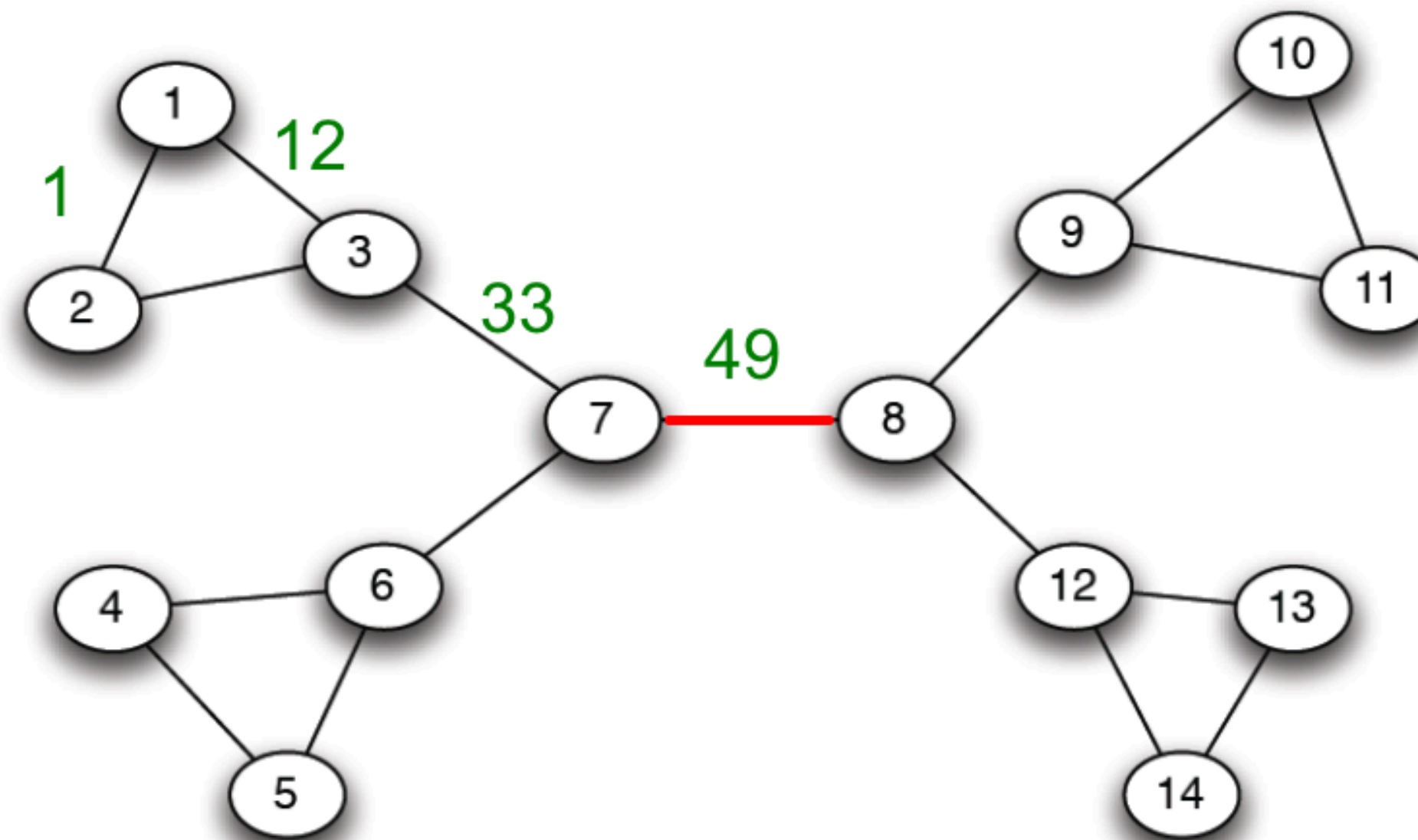
# Girvan-Newman: Example

Other edges:

3-7 carries full flow from 1,2,3 to 4-14:  $3 \times 11 = 33$

1-3 carries all flow from 1 to everyone else except 2:  $1 \times 12 = 12$

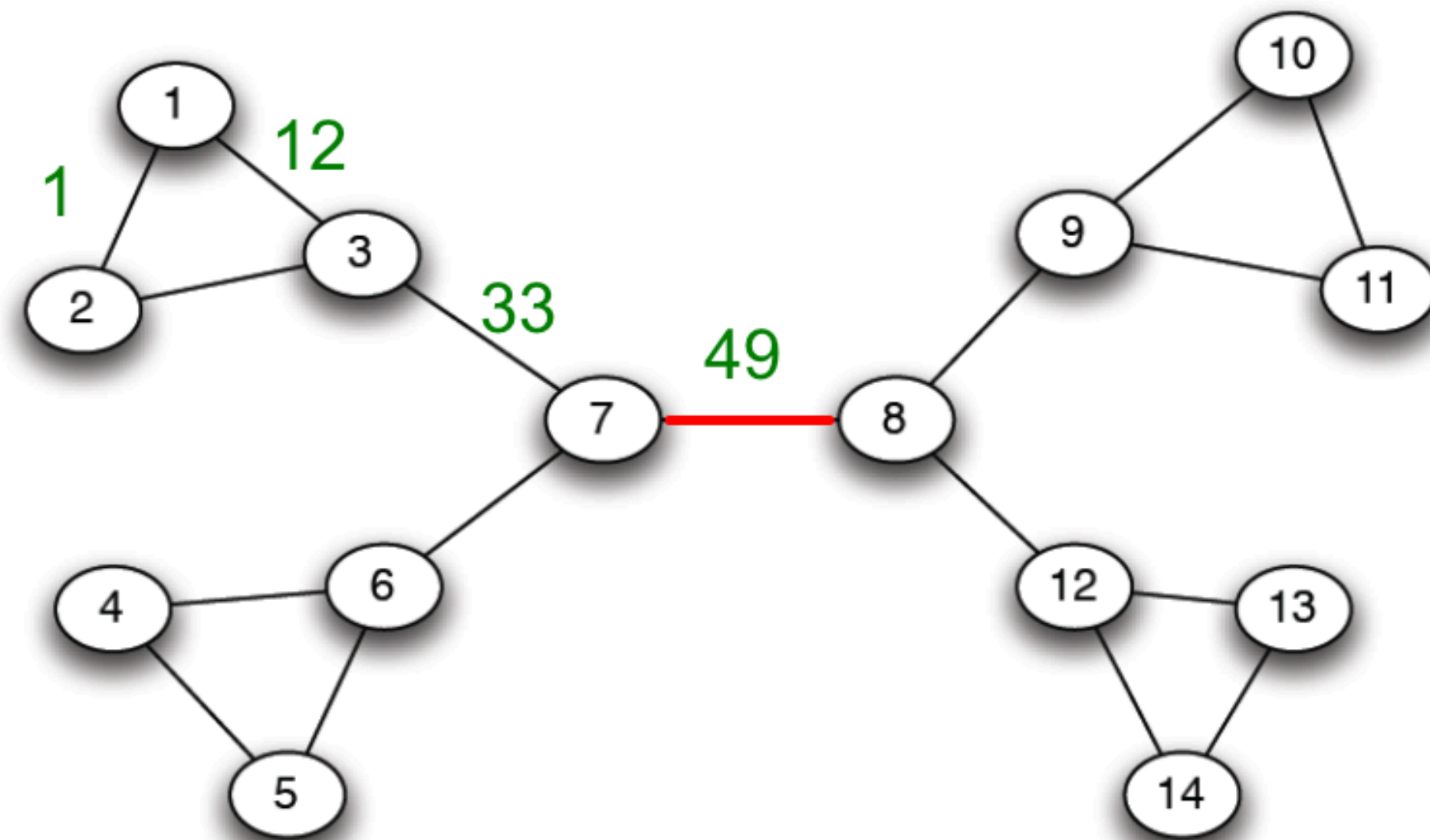
1-2 only carries flow from 1 to 2:  $1 \times 1 = 1$



By symmetry, we know  
betweenness for all other  
nodes as well in this graph

# Girvan-Newman: Example

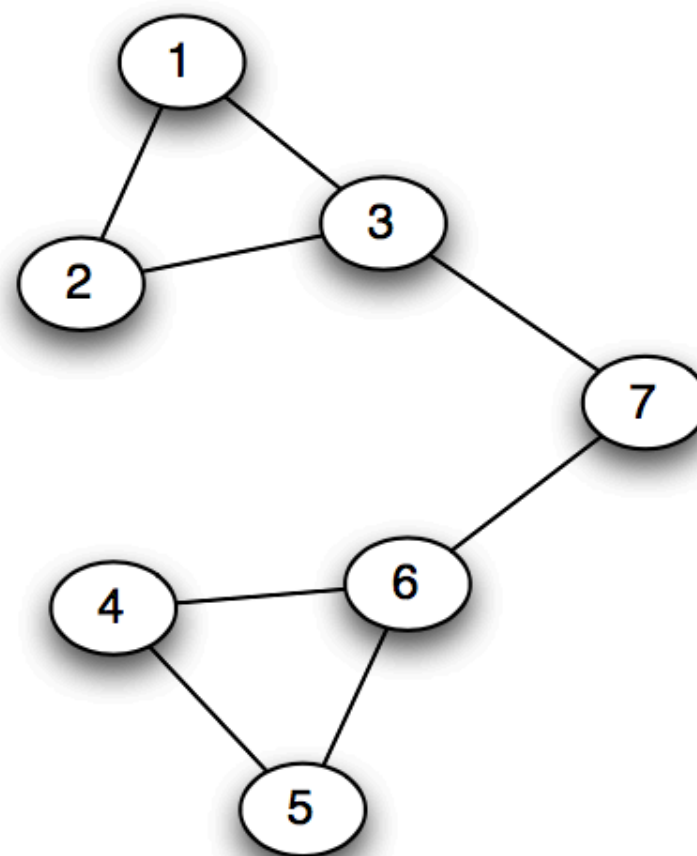
Girvan-Newman method: Remove edge of highest betweenness (or multiple if there is a tie)



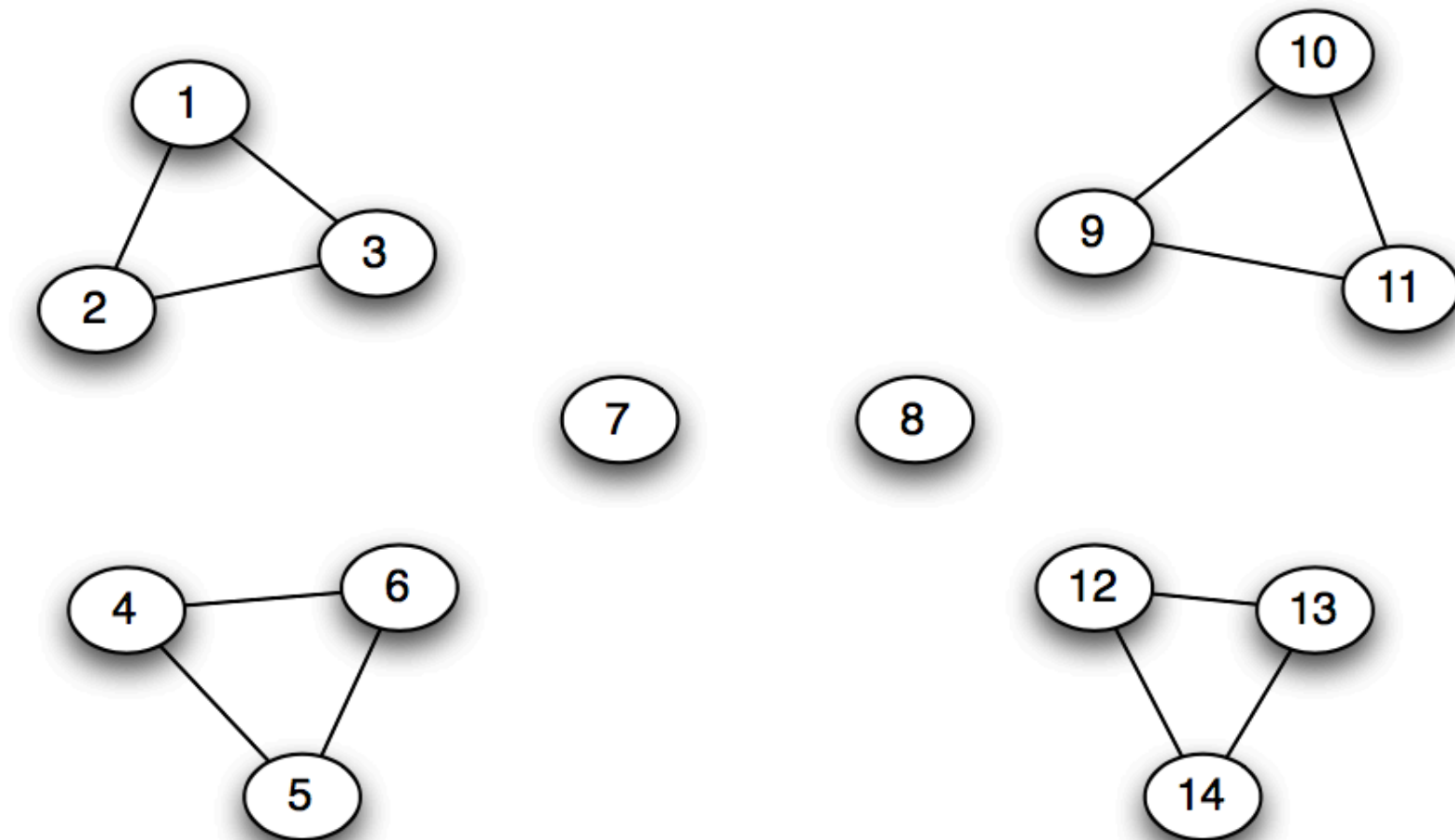
By symmetry, we know betweenness for all other nodes as well in this graph

# Girvan-Newman: Example

Step 1:

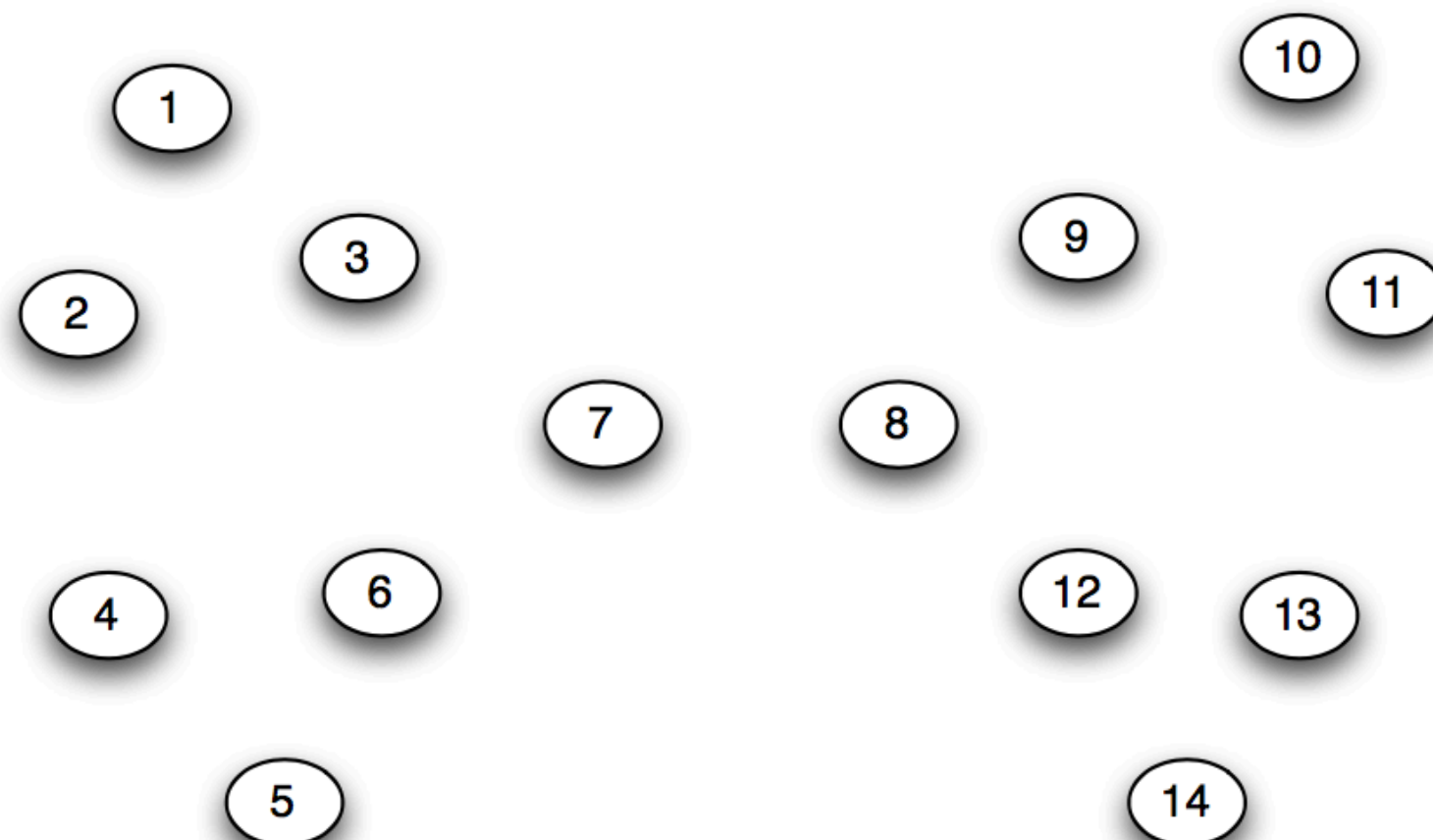


Step 2:

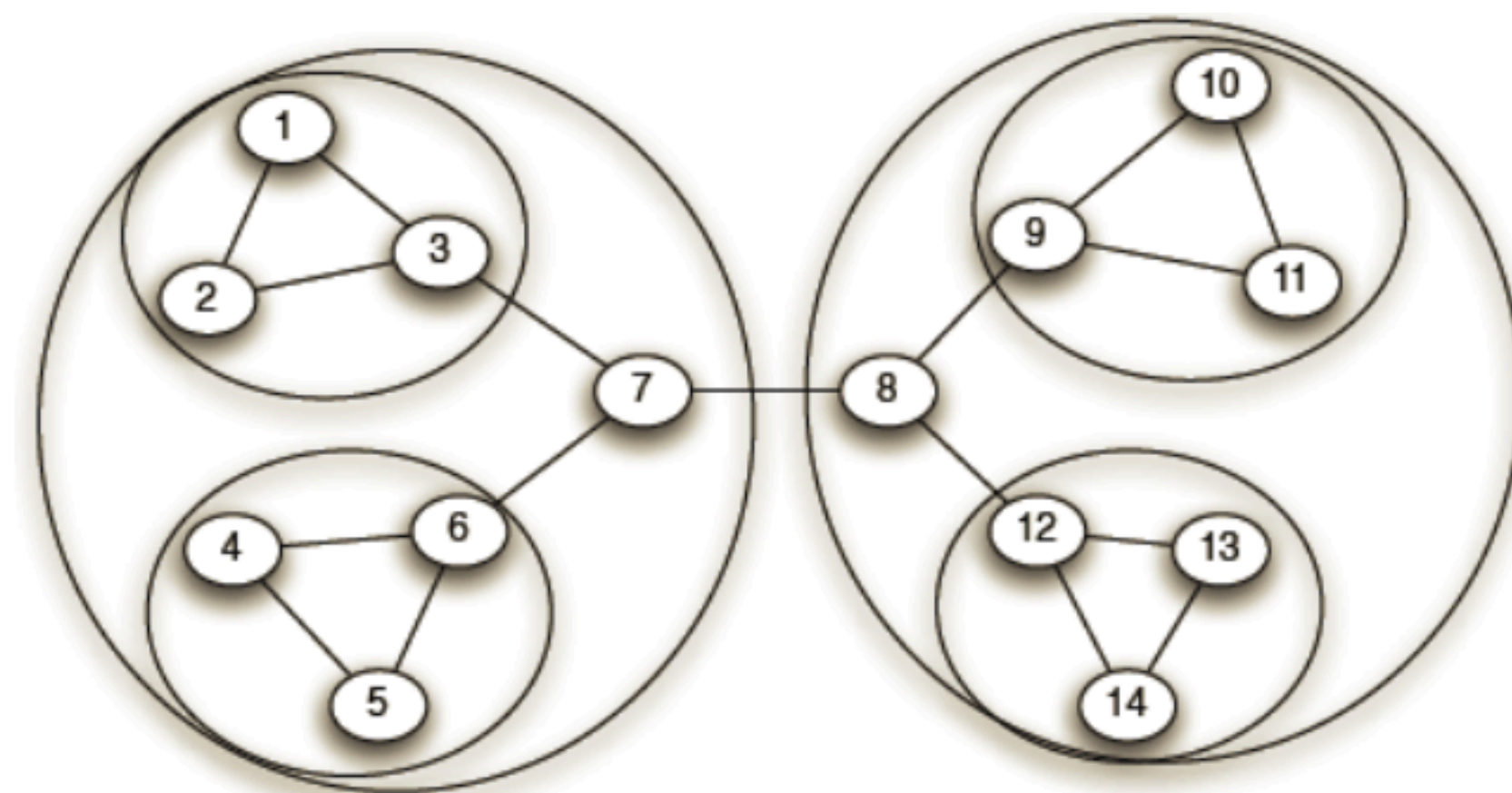


Need to re-compute  
betweenness at every step

Step 3:



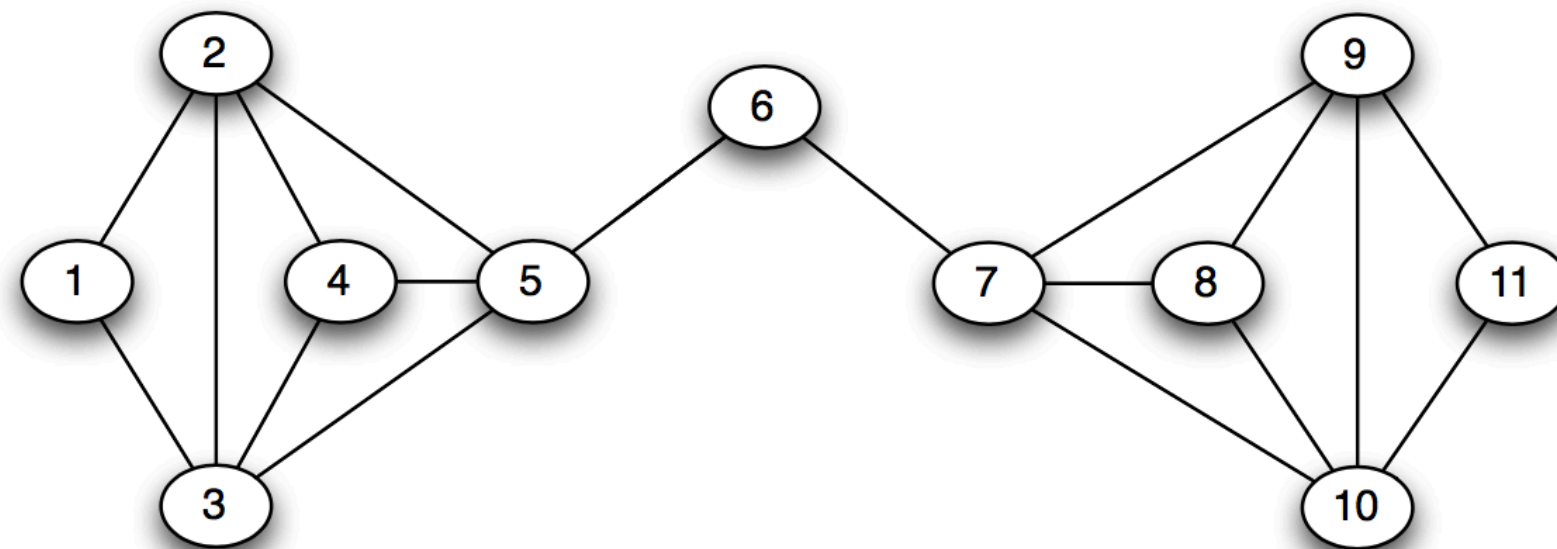
Hierarchical network decomposition:



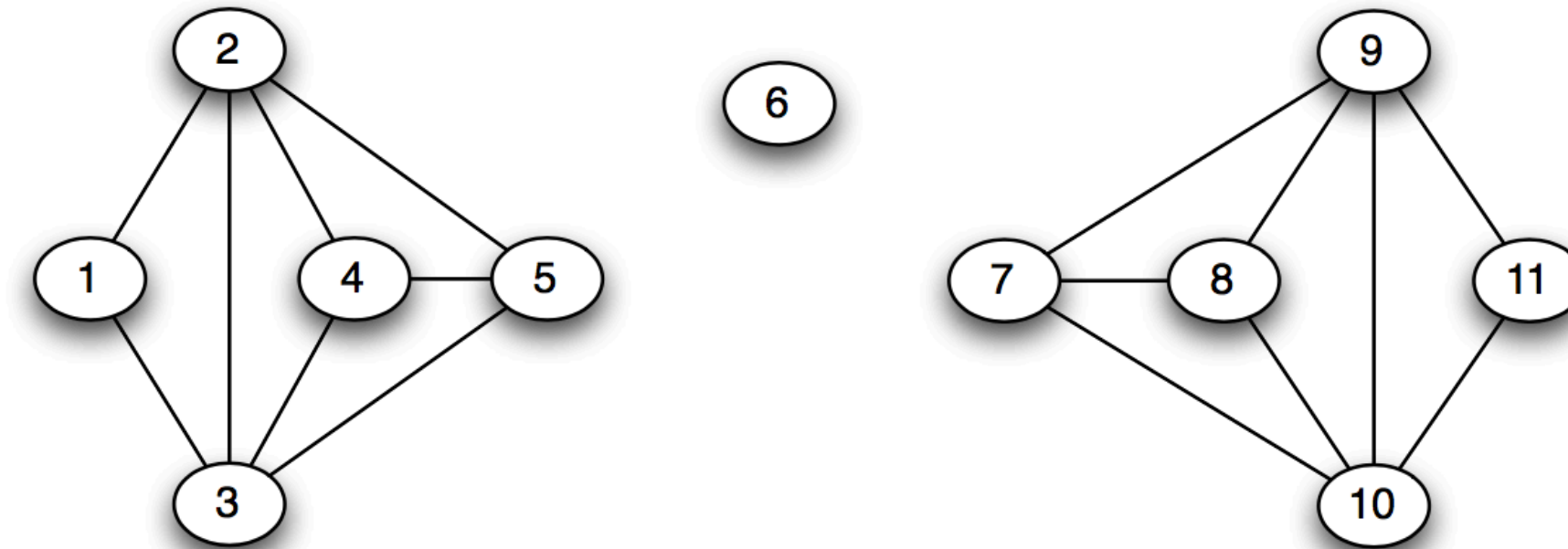


# Girvan-Newman: Example

Step 1:



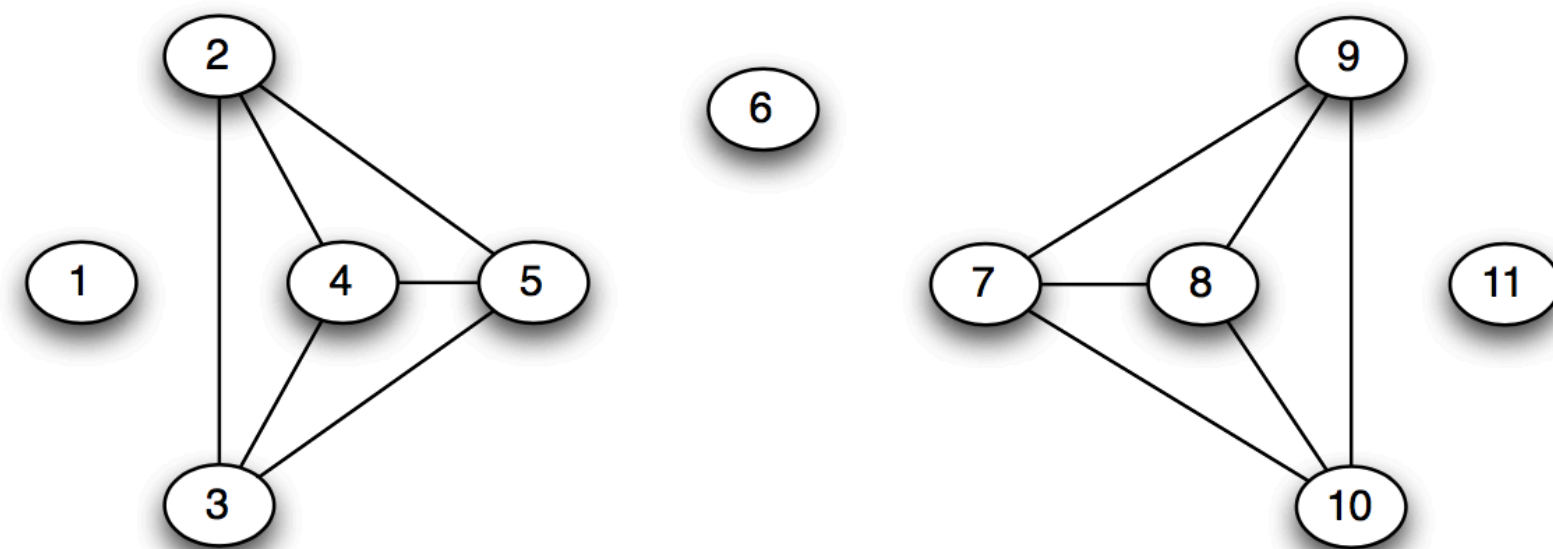
Step 2:



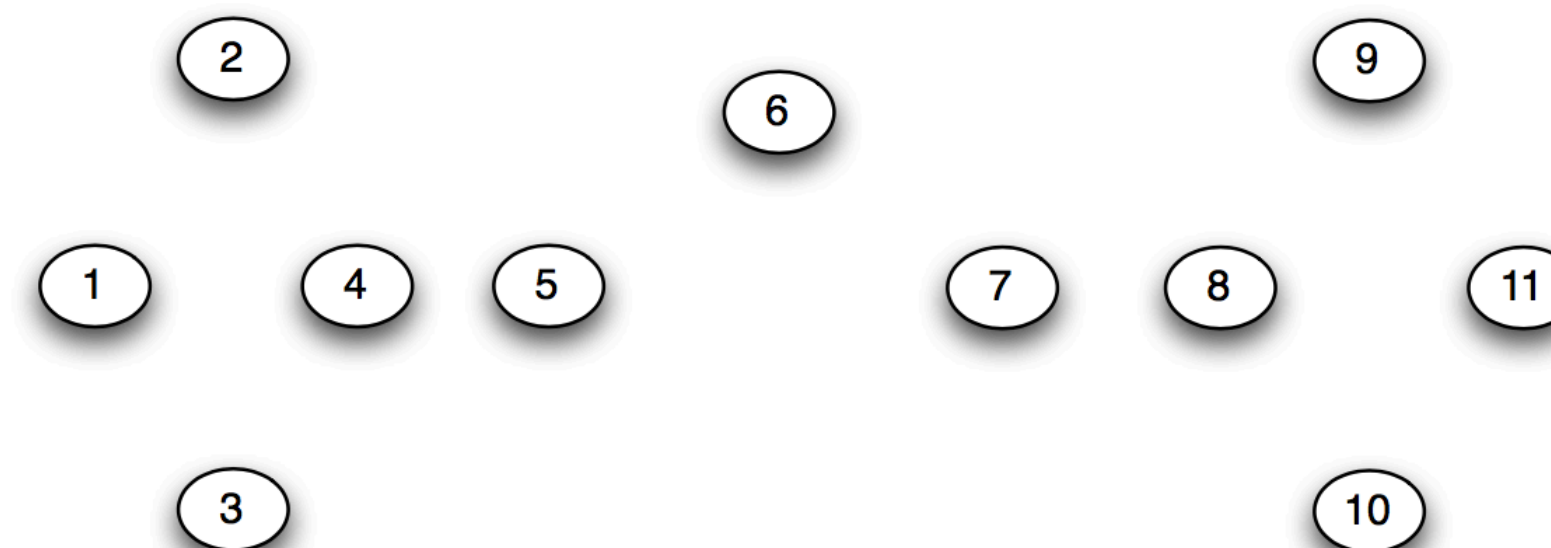
Need to re-compute  
betweenness at every step

25 units that used to be on 5-7  
get shifted to 5-6 and 6-7

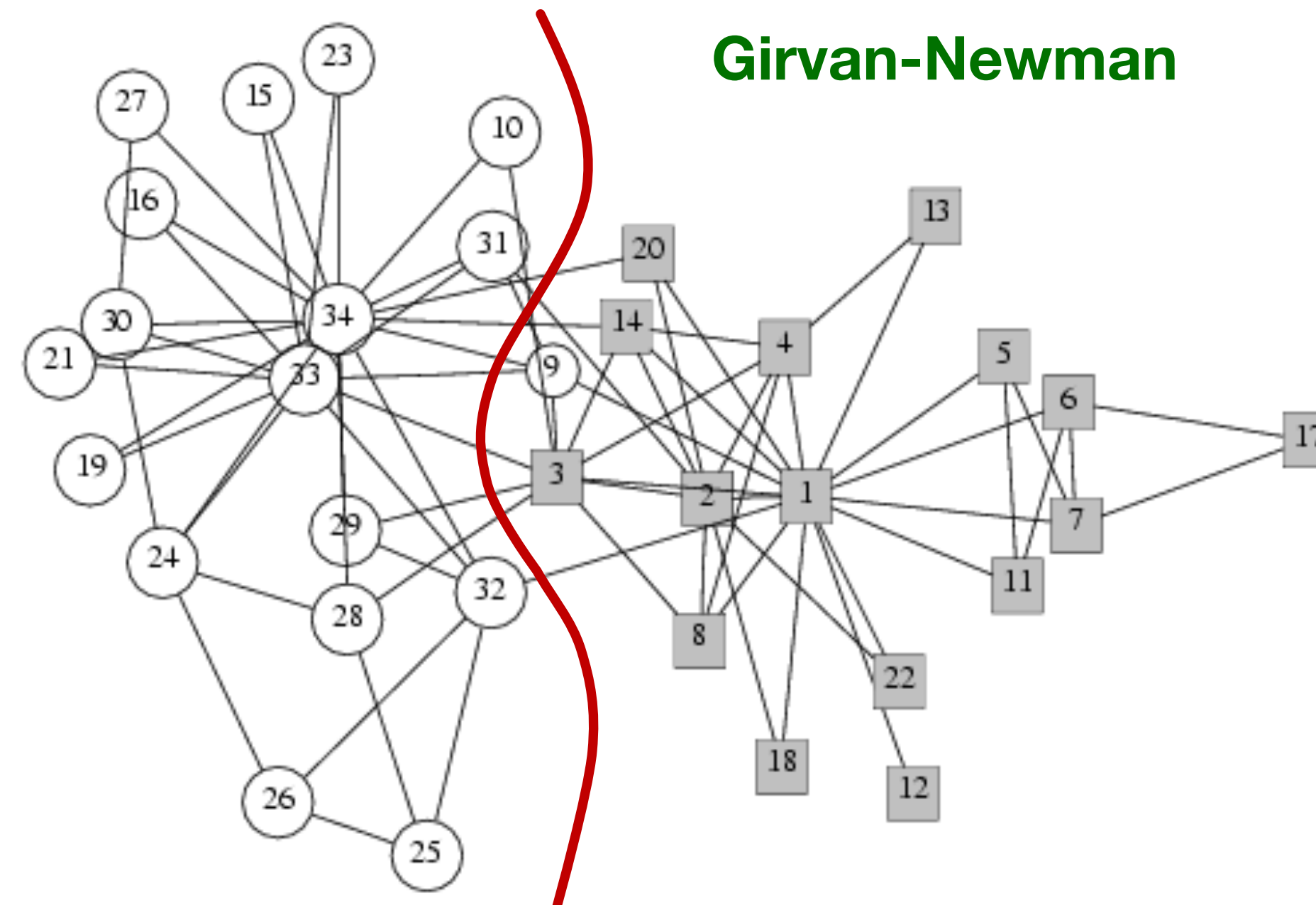
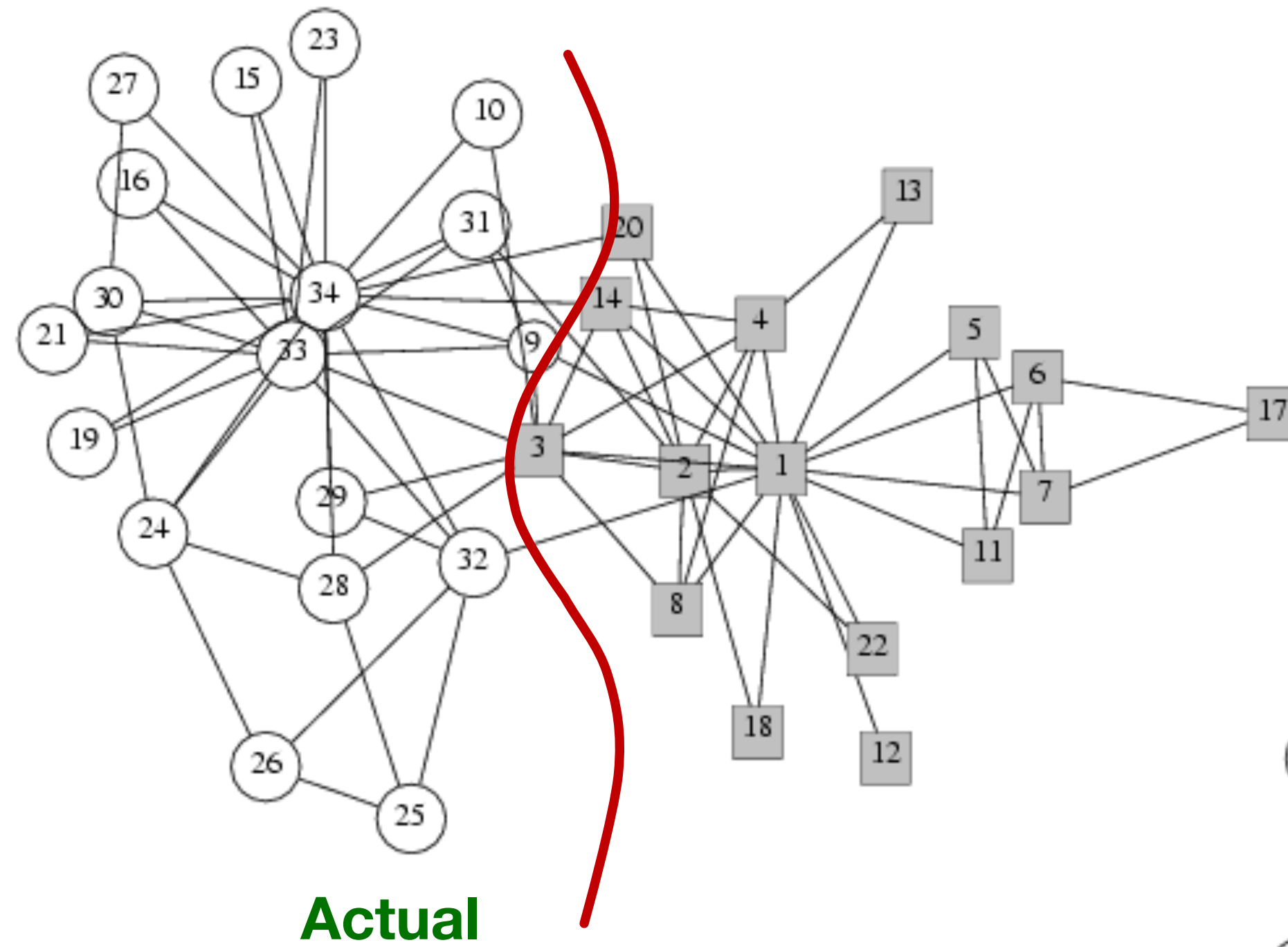
Step 3:



Step 4:



# Zachary Karate Club



# Visualizing Hierarchical Clusters

## Dendrogram

Graphical depiction of the hierarchical clustering splits done at every step



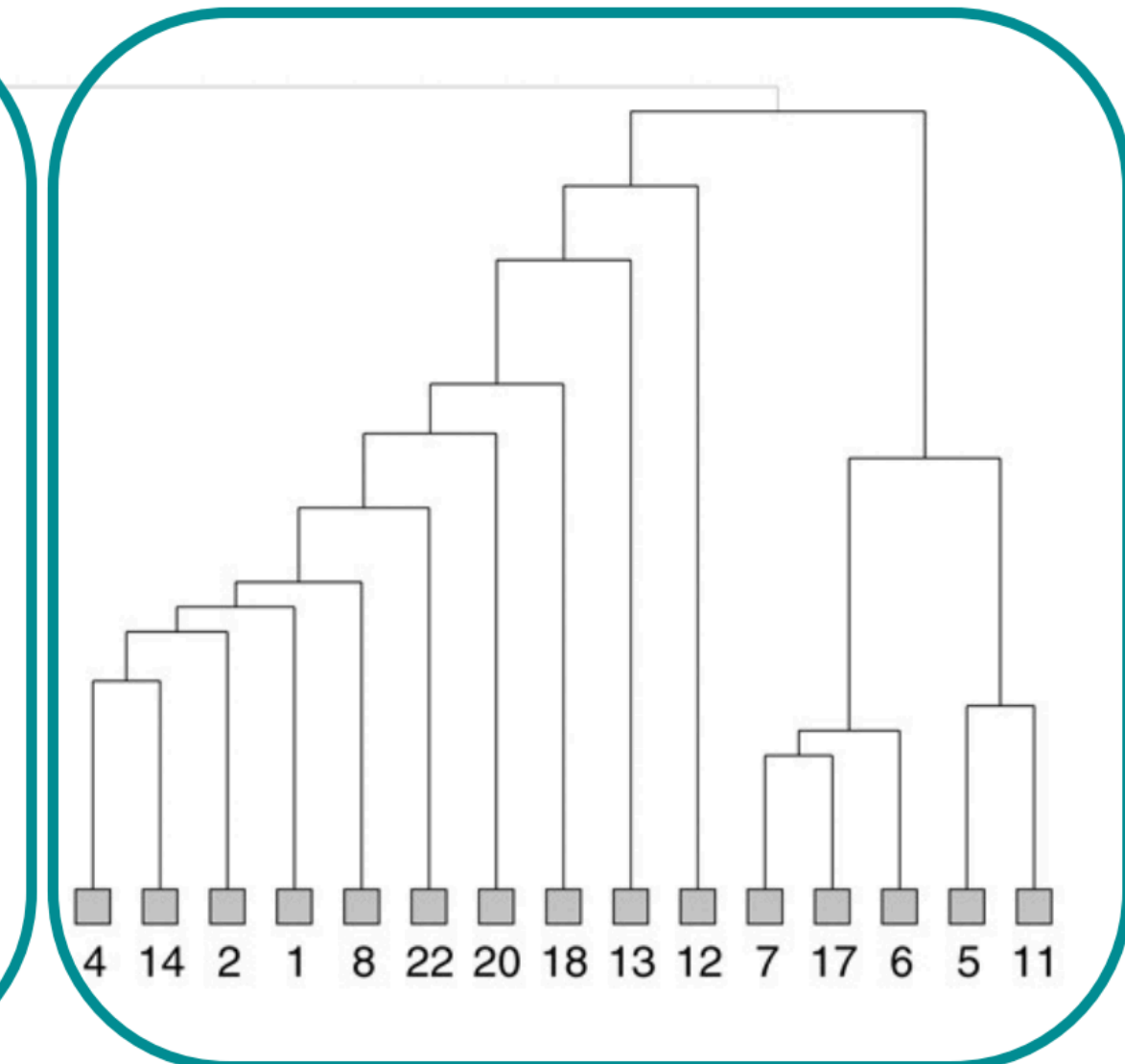
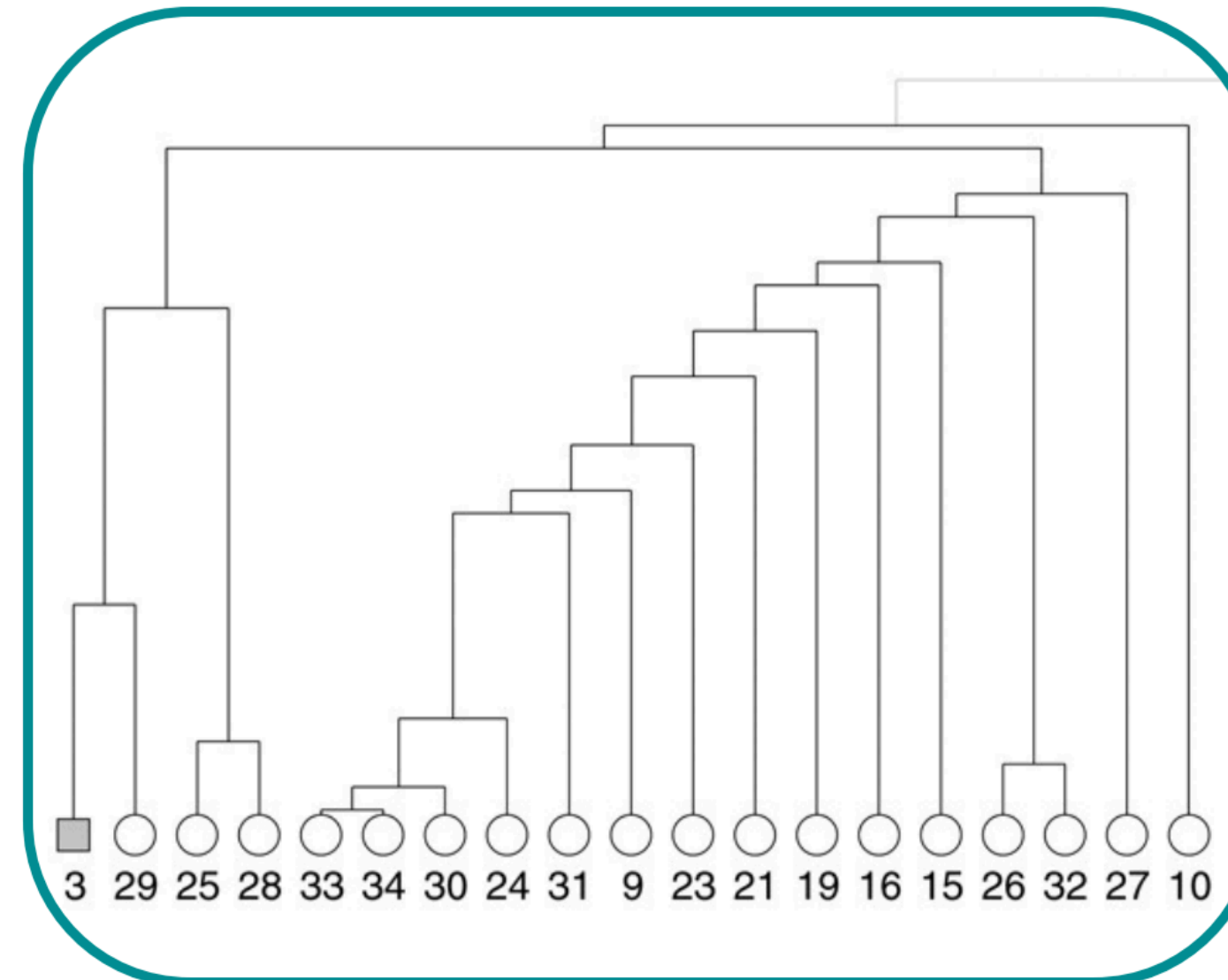
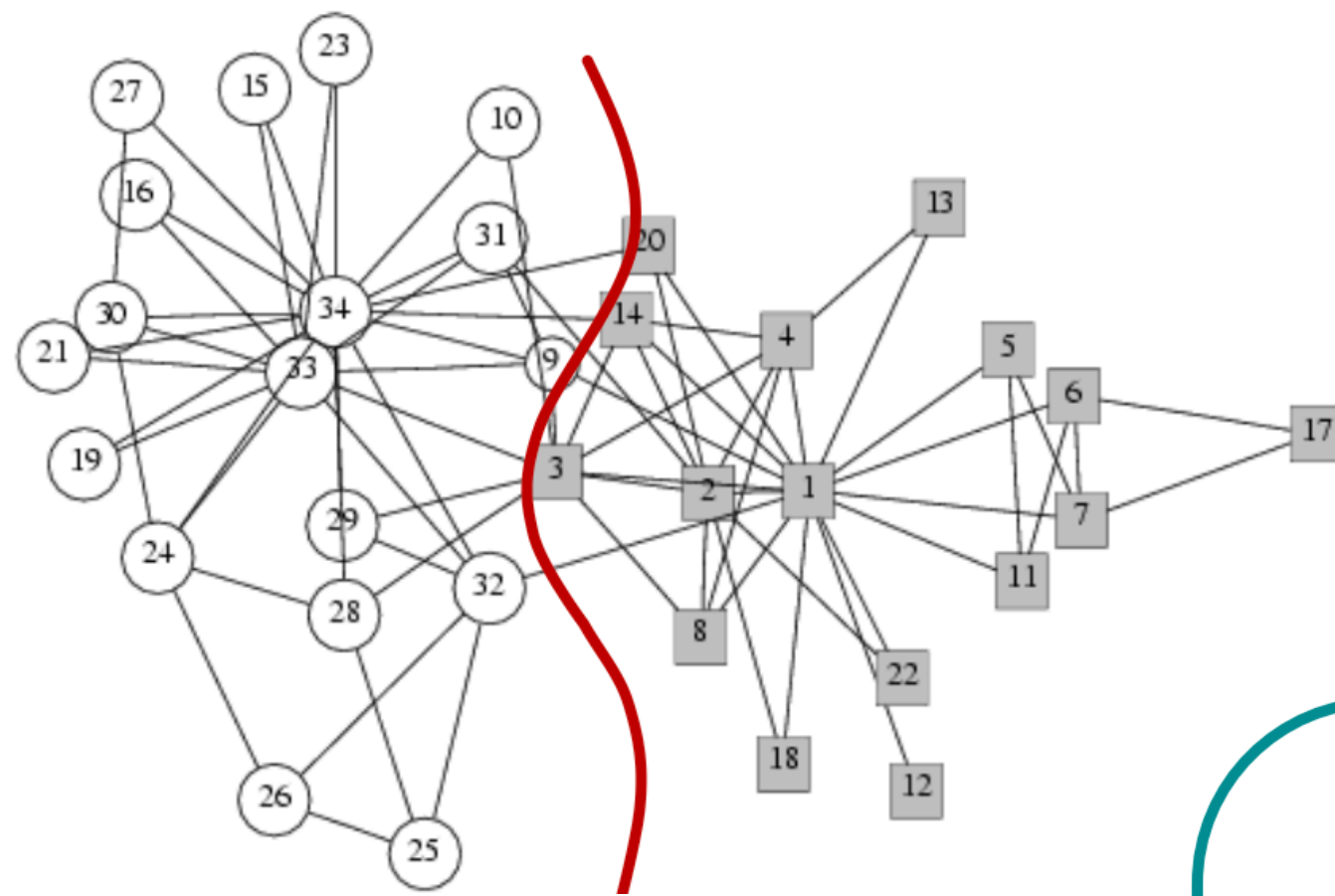
“First AB/CDEF, then C/DEF, then D/EF, then A/B, then E/F”



# Zachary Karate Club

## Dendrogram

Graphical depiction of the hierarchical clustering splits done at every step



# Girvan-Newman: Results

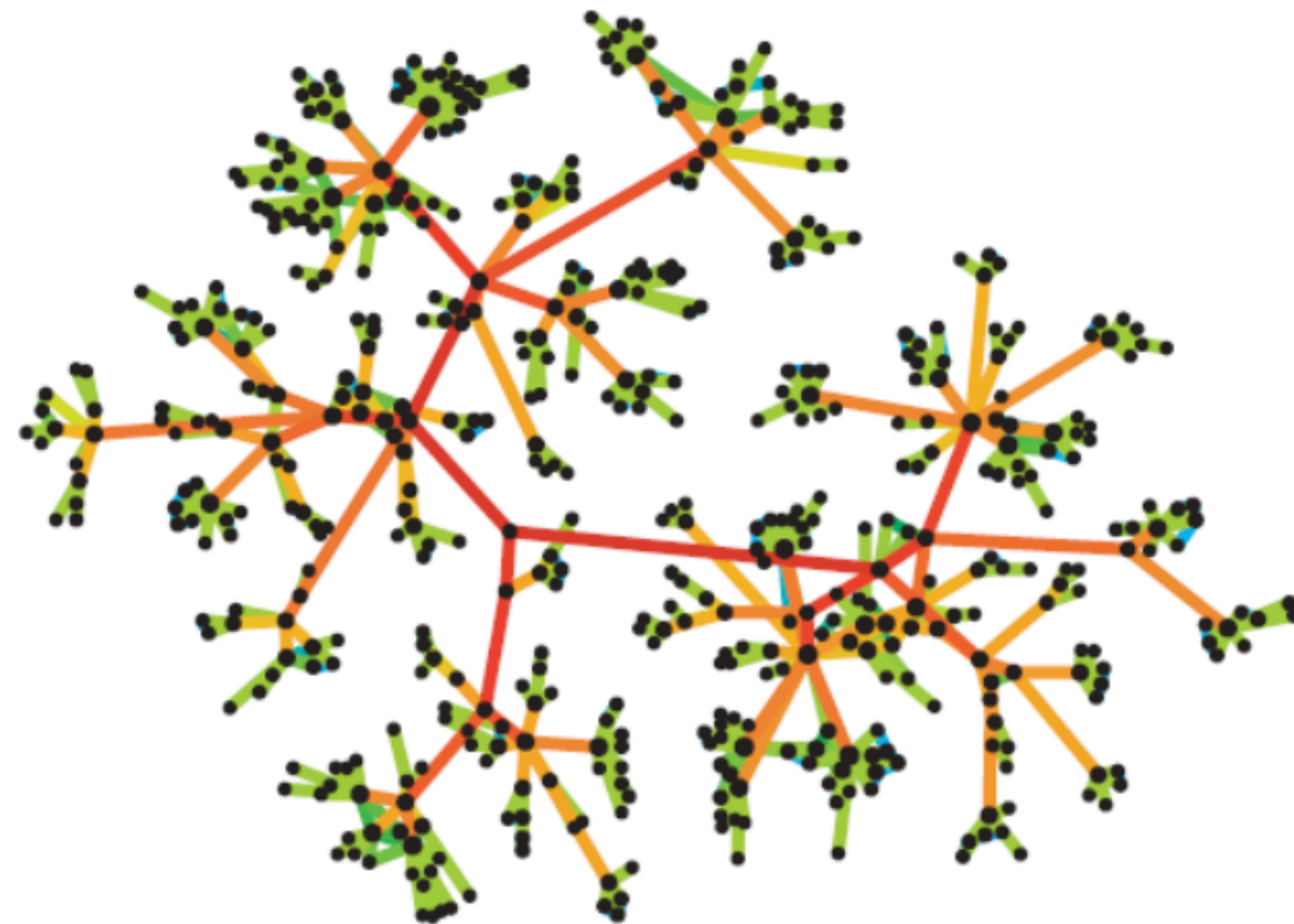


# Communities in physics collaborations

# We need to resolve a question

## How to compute betweenness?

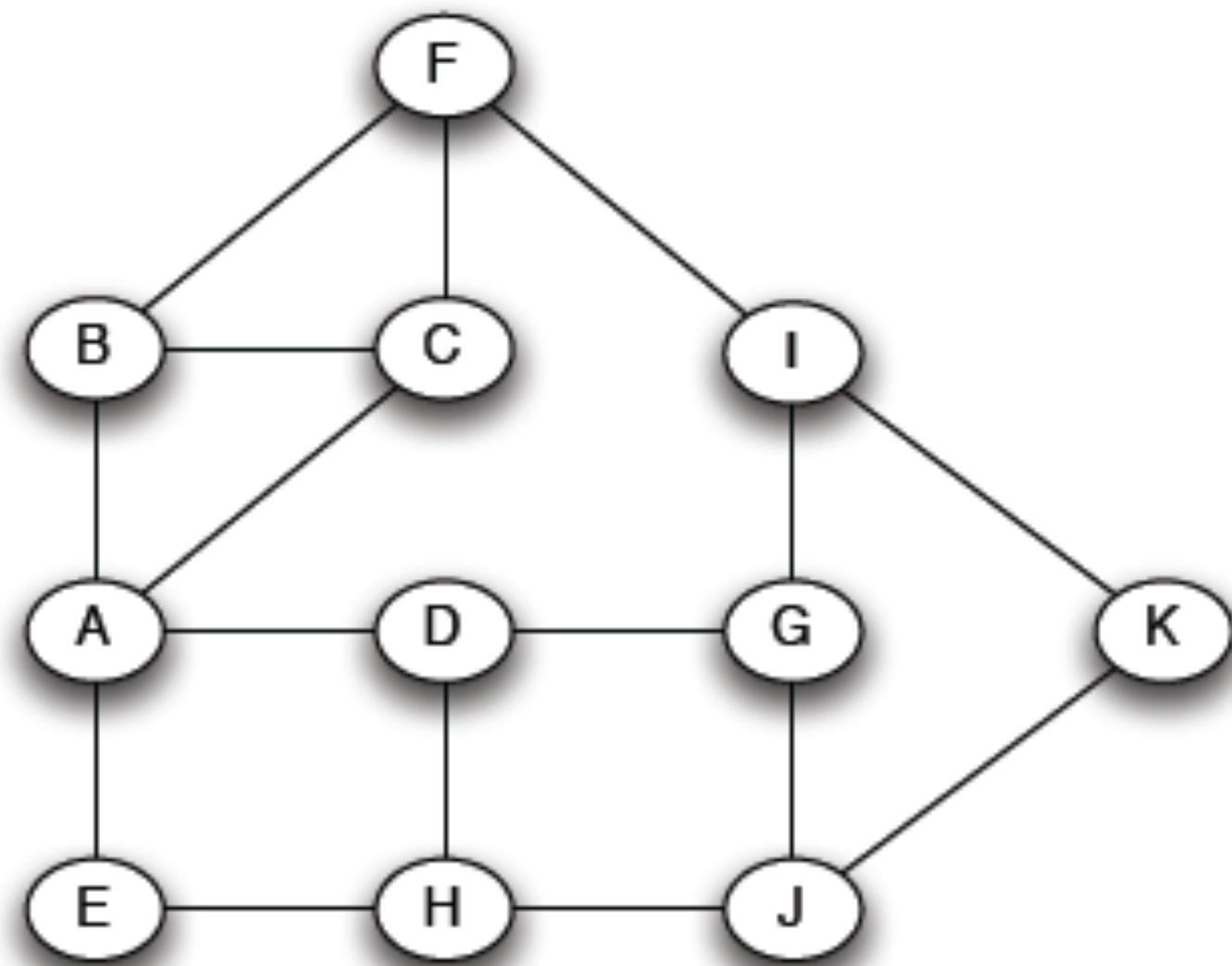
Counting all pairs of shortest paths for every edge is **computationally challenging!**



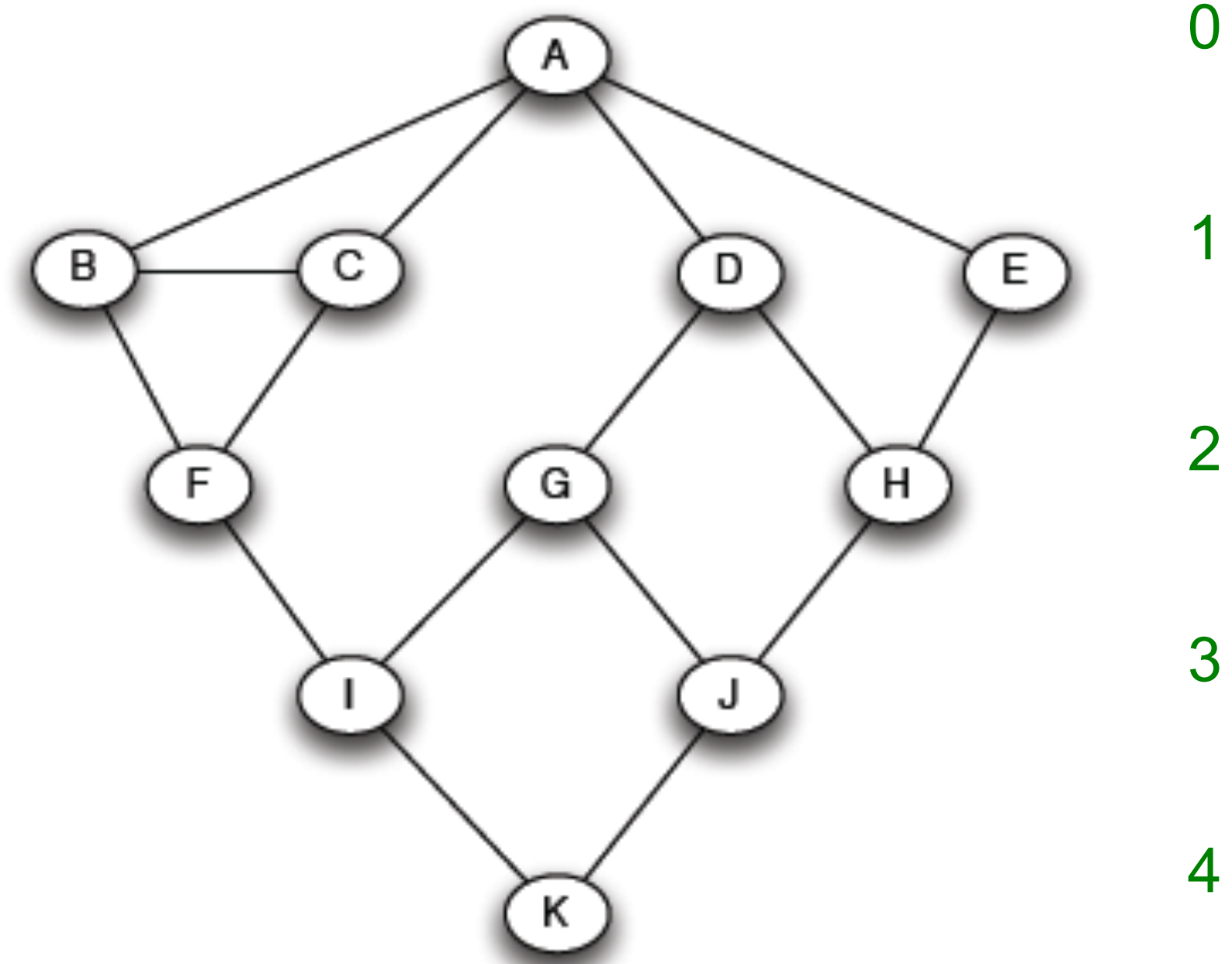


# How to Compute Betweenness?

Want to compute  
betweenness of paths  
starting at node A



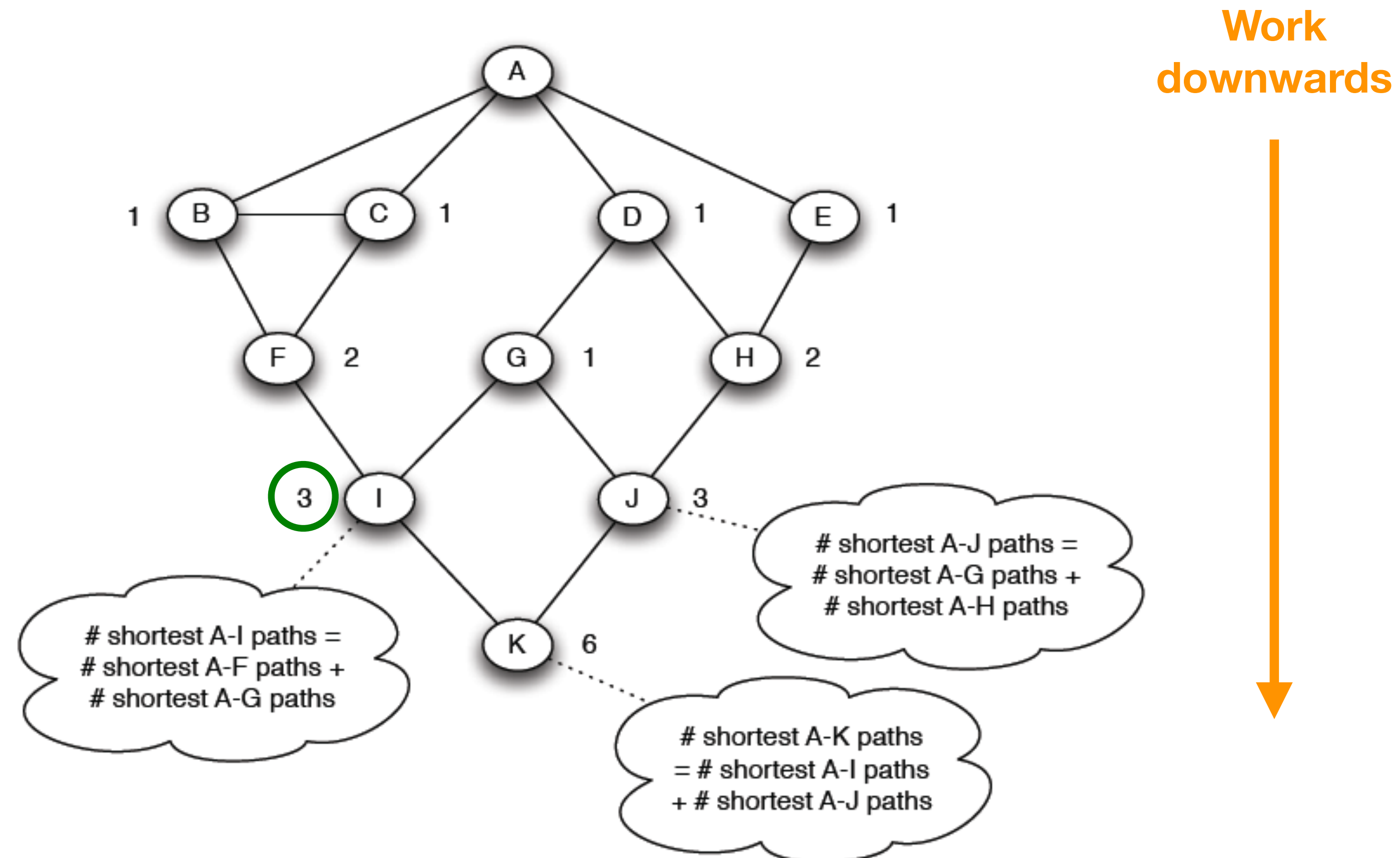
BFS starting from A:



Recall BFS goes layer-by-layer

# How to Compute Betweenness?

Count the number of shortest paths from A to all other nodes in the graph:



# How to Compute Betweenness?

How much flow goes from A to other nodes?

**Compute betweenness by working up the tree:** If there are multiple paths count them fractionally

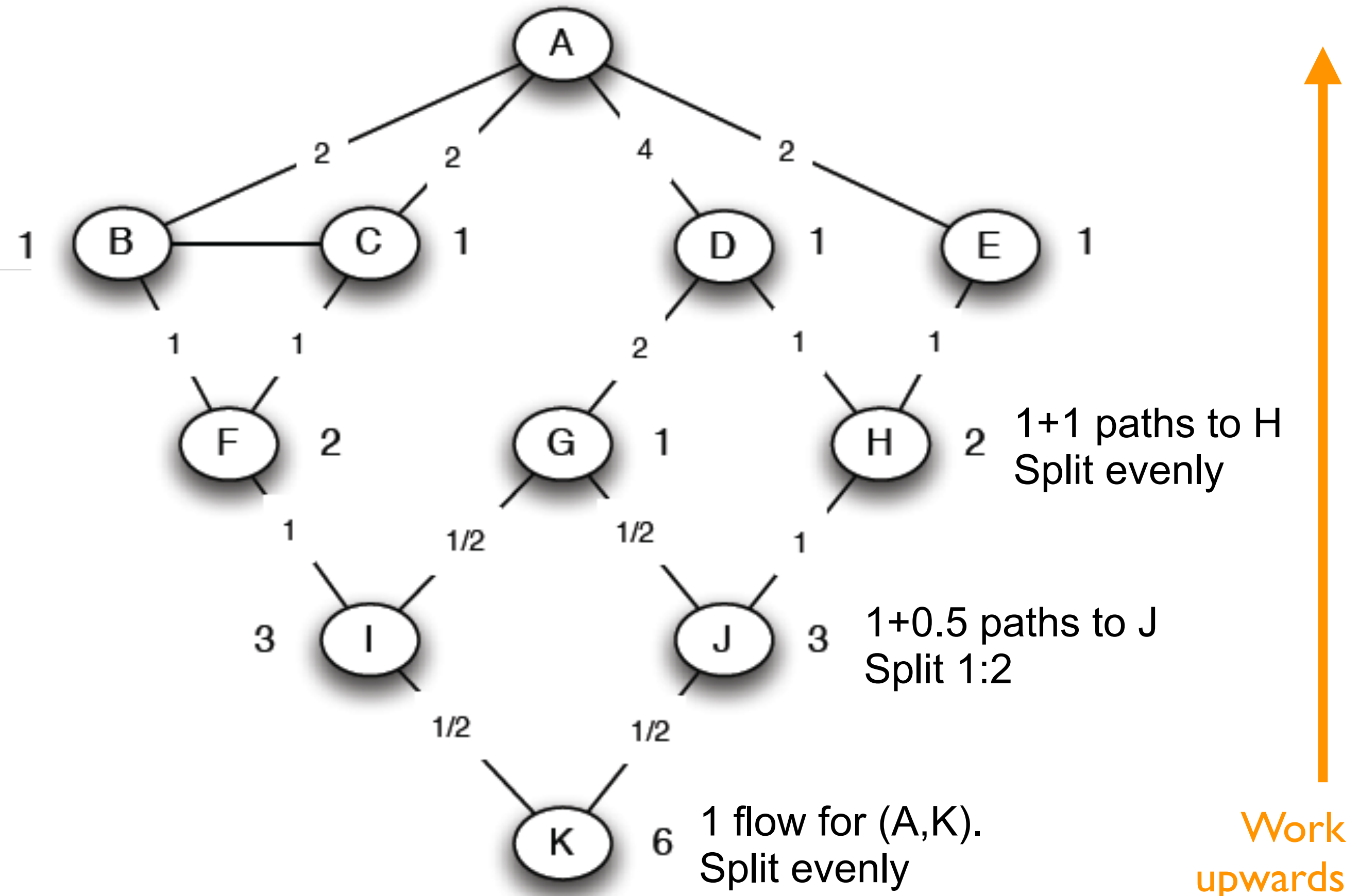
## The algorithm:

### • Add edge flows:

-- node flow =  
 $1 + \sum \text{child edges}$

-- split the flow up  
based on the parent  
value

• Repeat the BFS  
procedure for each  
starting node  $U$





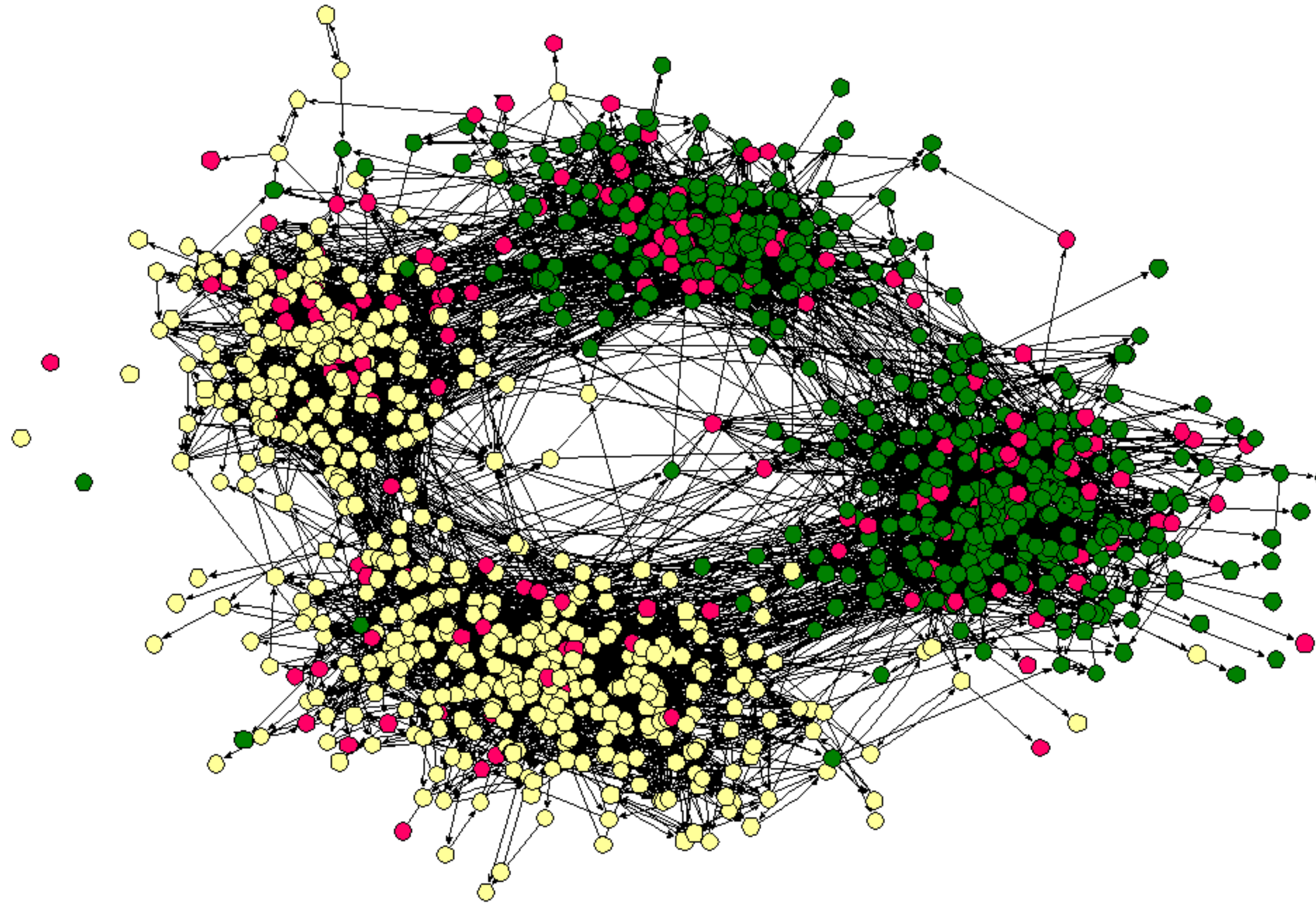
# Girvan-Newman

- Repeat for each node in the graph, add up the edge scores that edges receive in these computations
- For each edge  $(u,v)$ , must divide by 2 because we counted it once for  $u$  and once for  $v$
- Works on moderately-sized graphs
- To scale to big data, still expensive, and requires approximations or related more efficient methods



**Homophily**  
**“Birds of a Feather Flock Together”**

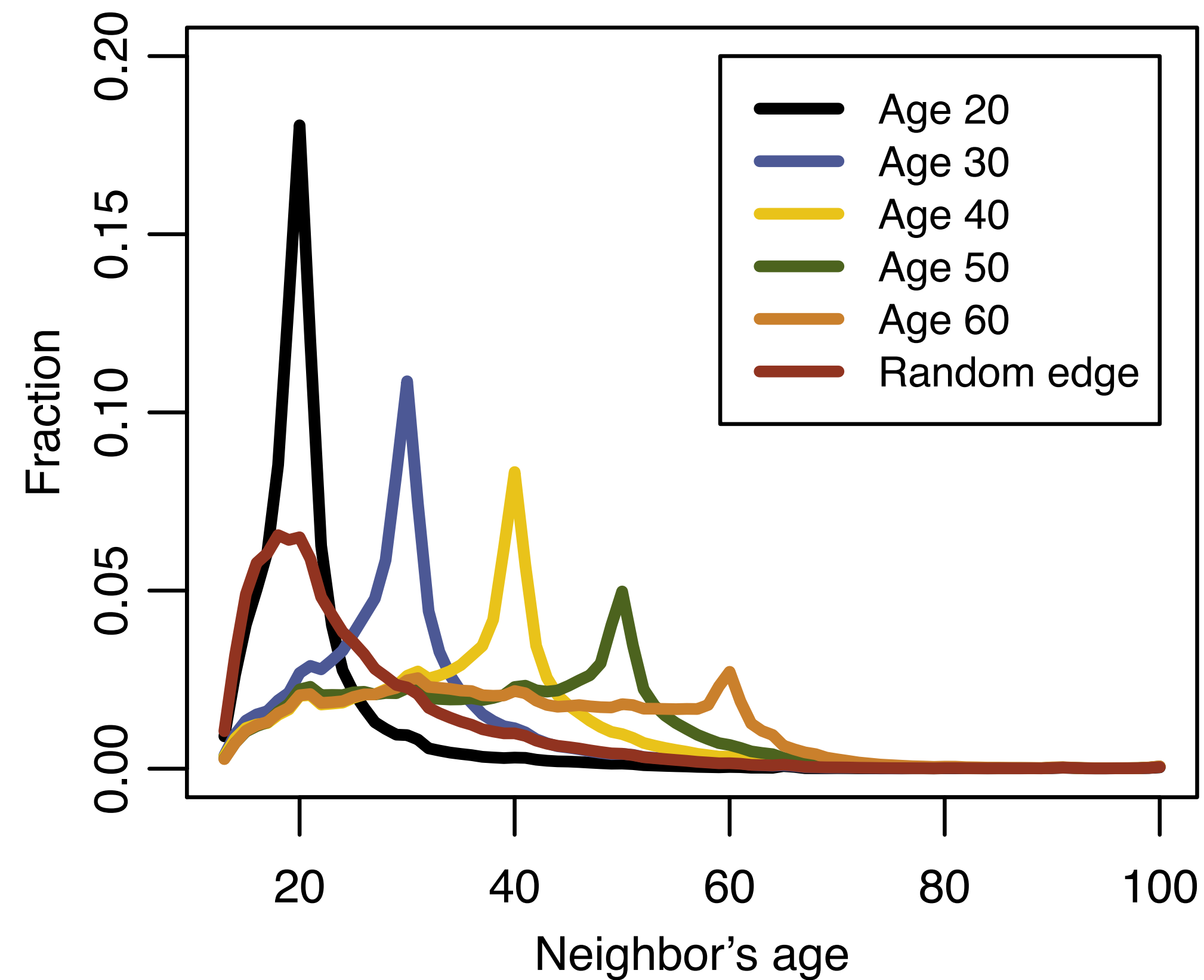
# Homophily



- US middle school + high school
- node color = self-identified race

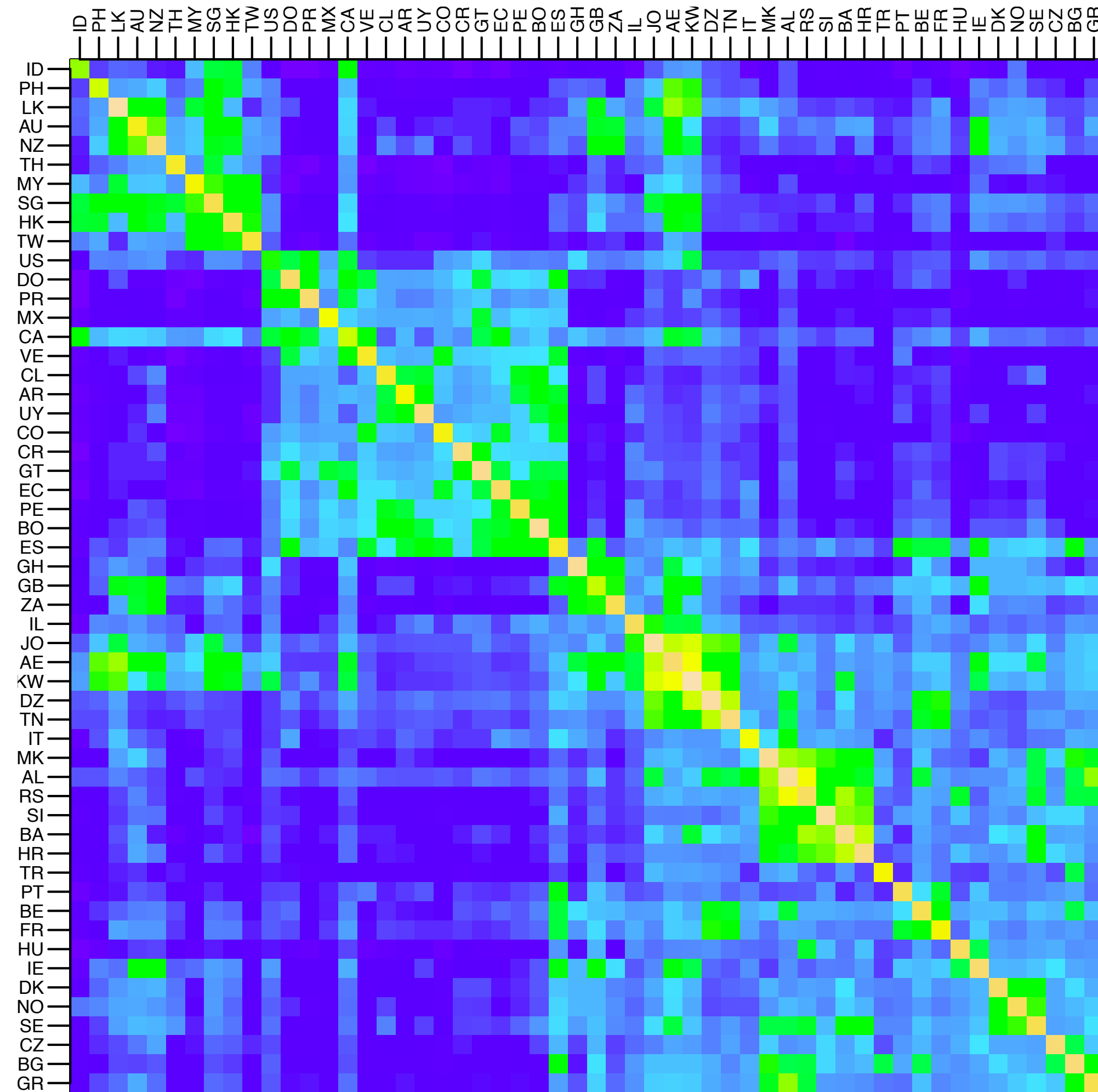


# Homophily: Age



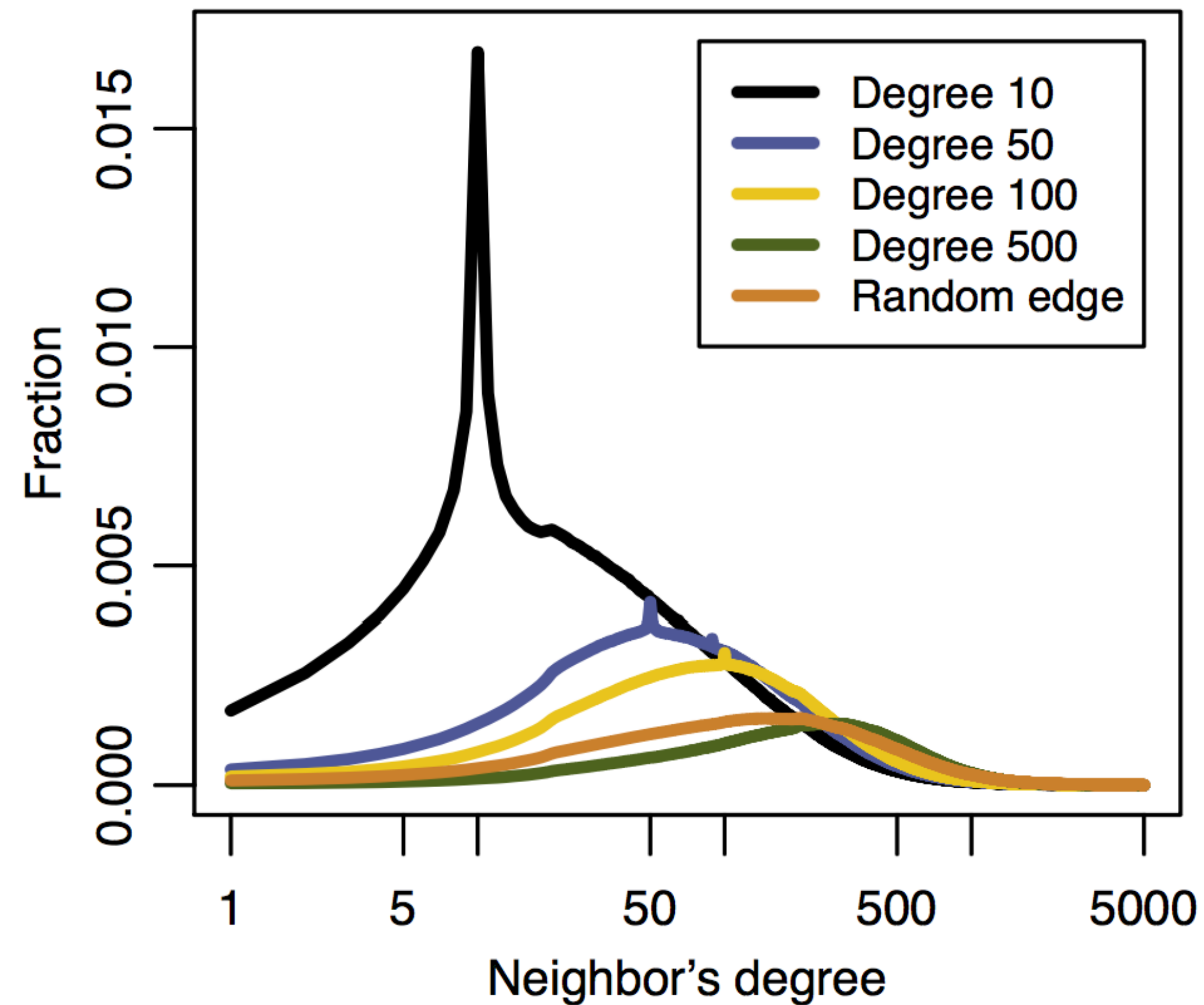
- Facebook friendship network, 2011

# Homophily: Nationality



- Facebook friendship network, 2011

# Homophily: Friend count

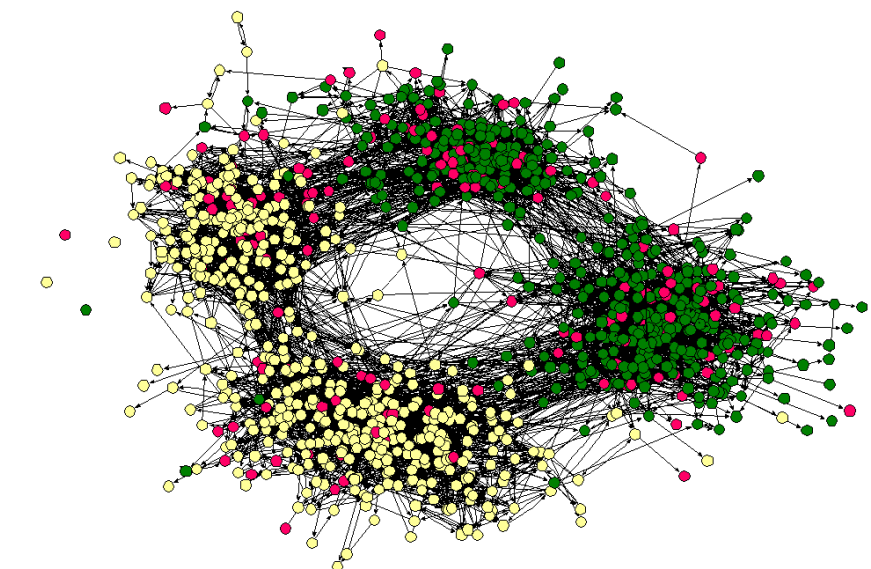


- Facebook friendship network, 2011



# Homophily

- Connections don't form uniformly at random
- **Null model**: what if they were forming at random?
- **Measuring homophily**: are there fewer connections between nodes across traits than you'd expect at random?
- **Homophily test**: If the fraction of cross-gender edges is significantly less than at random, then there is evidence of homophily.



# Homophily

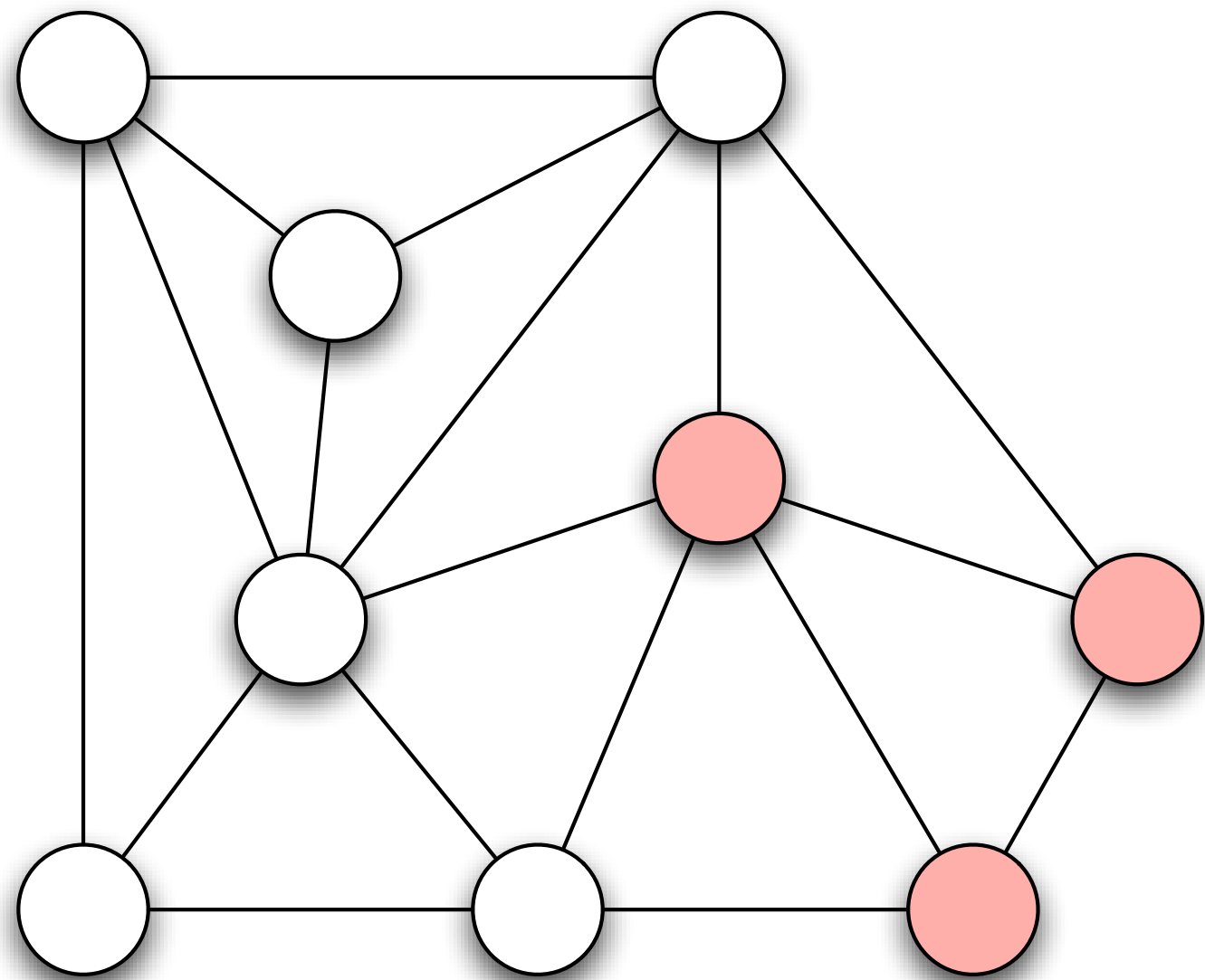
$p$  = Probability that a node is white

$q$  = Probability that a node is red

Prob an edge is between two white nodes?

Prob an edge is between two red nodes?

Prob an edge is between 1 red, 1 white?



Homophily test:

# Homophily

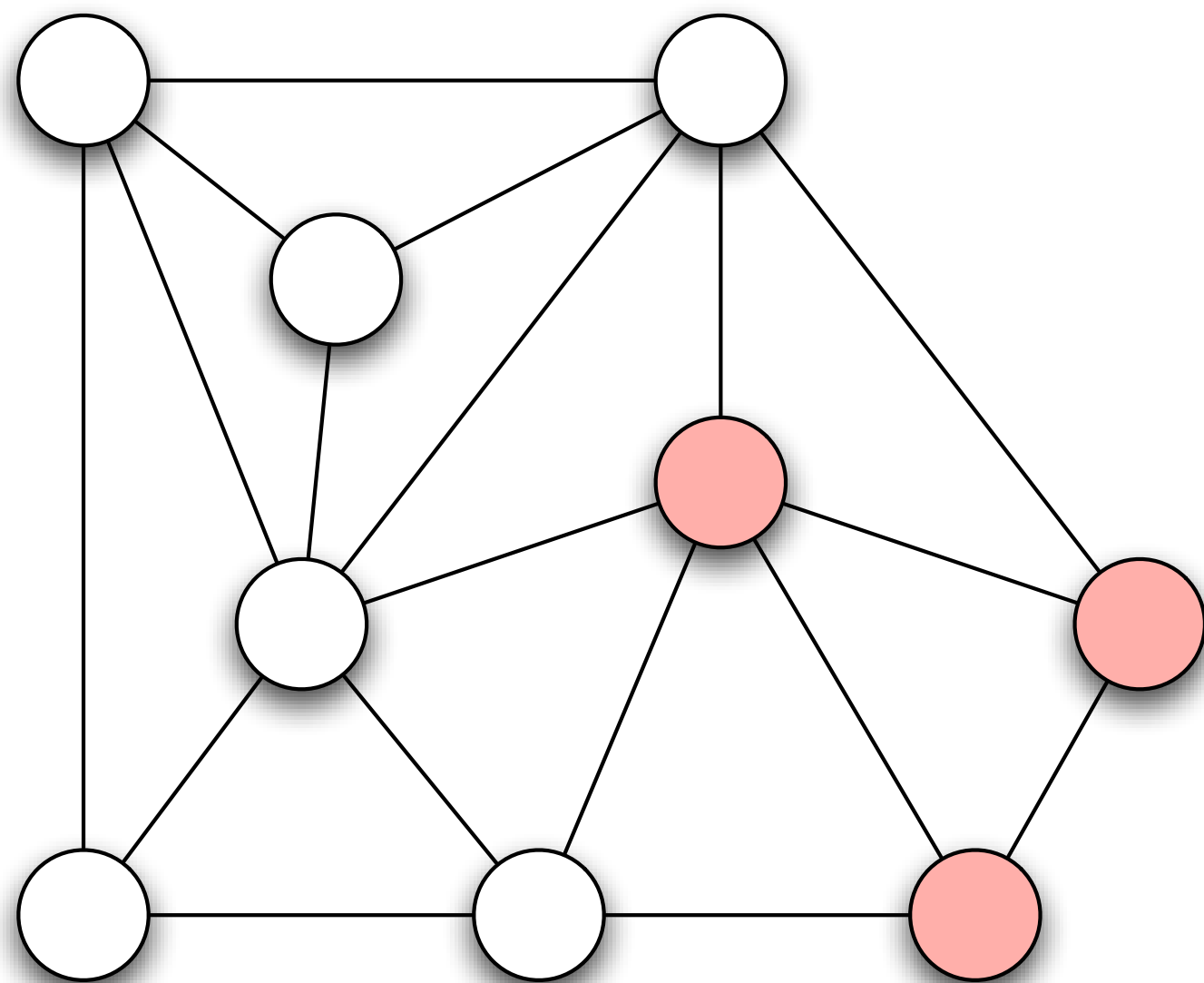
p = Probability that a node is white 6/9=2/3

q = Probability that a node is red 3/9=1/3

Prob an edge is between two white nodes?  $p^2$

Prob an edge is between two red nodes?  $q^2$

Prob an edge is between 1 red, 1 white?  $2pq$



Homophily test:

$$2pq = 4/9 = 8/18$$

Observed: 5/18



# The Friendship Paradox

# Friendship paradox

Your friends probably have **more friends** than you do

# Friendship paradox

Average degree  $\leq$  Average friend degree



# Friendship paradox

- **Facebook friend graph (2012):**
  - 720M people, 70B edges
  - Average Facebook user number of friends: 190
  - Average friend's number of friends: 635
  - User's friend count was lower than the average of their friends' friend counts 93% of the time
  - ???

# Friendship paradox

- Consider an example:
  - Two buses to school
    - One **big** one with 90 students
    - One **small** one with 10 students
  - Average bus size = 50
  - This is misleading...

# Friendship paradox

- Consider an example:
  - Two buses to school
    - One **big** one with 90 students
    - One **small** one with 10 students
  - Average bus size = 50
  - What about average *bus-rider experience*?



# Friendship paradox

## ■ From students' point of view:

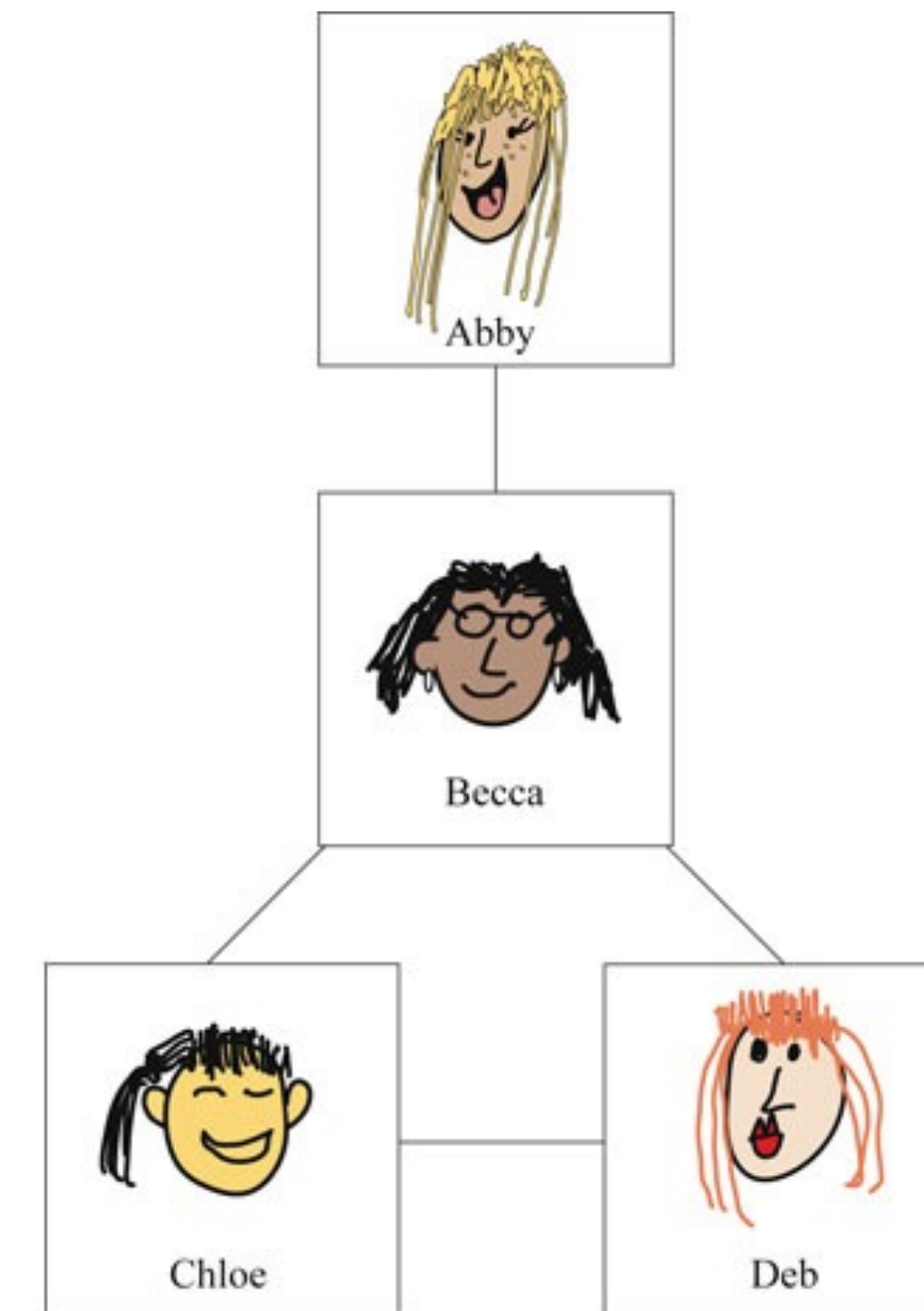
- How packed is your bus?
  - 90 students say 90
  - 10 students say 10

Average *bus-rider experience* =

$$[(90*90)+(10*10)]/100 = 82$$

# Friendship paradox

- Friend counts: 1, 3, 2, 2.
- Average friend count:
- Average friend count of a friend:



# Friendship paradox

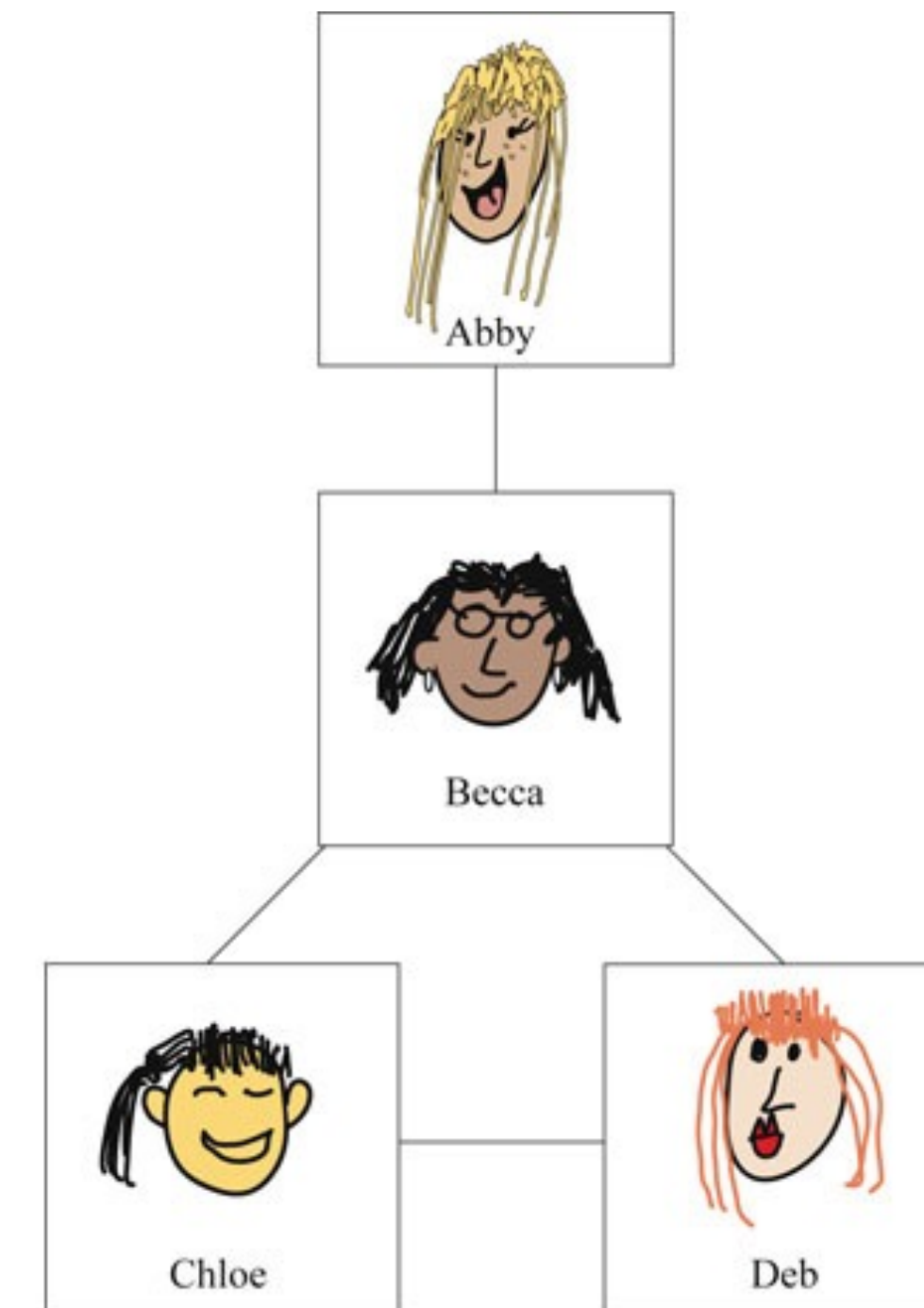
- Friend counts: 1, 3, 2, 2.
- Average friend count:  $8/4=2$
- Average friend count of a friend:

A: 3, avg = 3  
B: 1, 2, 2, avg =  $5/3$   
C: 3, 2, avg = 2.5  
D: 3, 2, avg = 2.5

Avg friend of friends =  $2.4166 > 2$

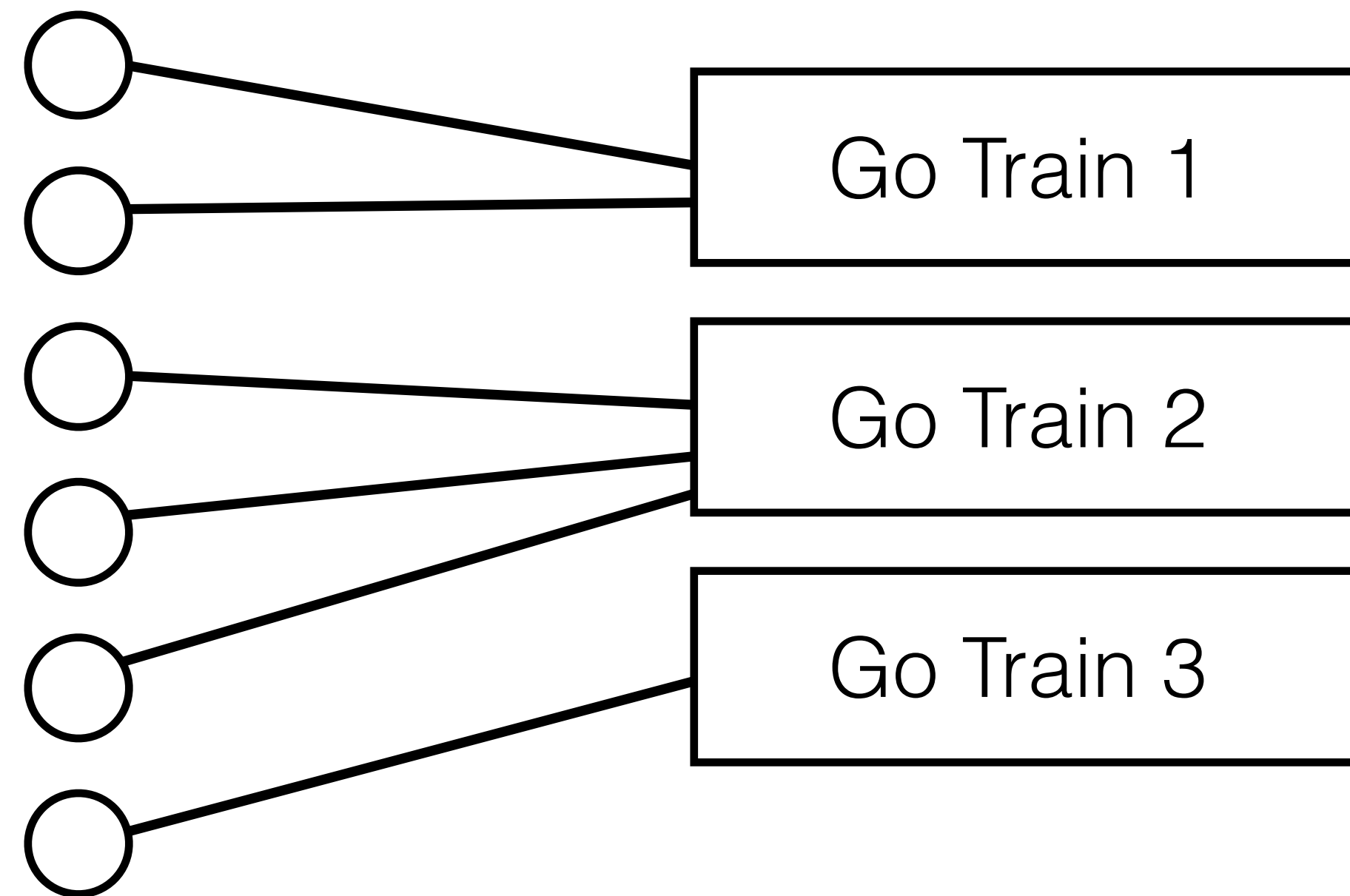
B mentioned 3 times, A only 1

“Average friend-experience” vs. average friends



# “Friendship paradox”

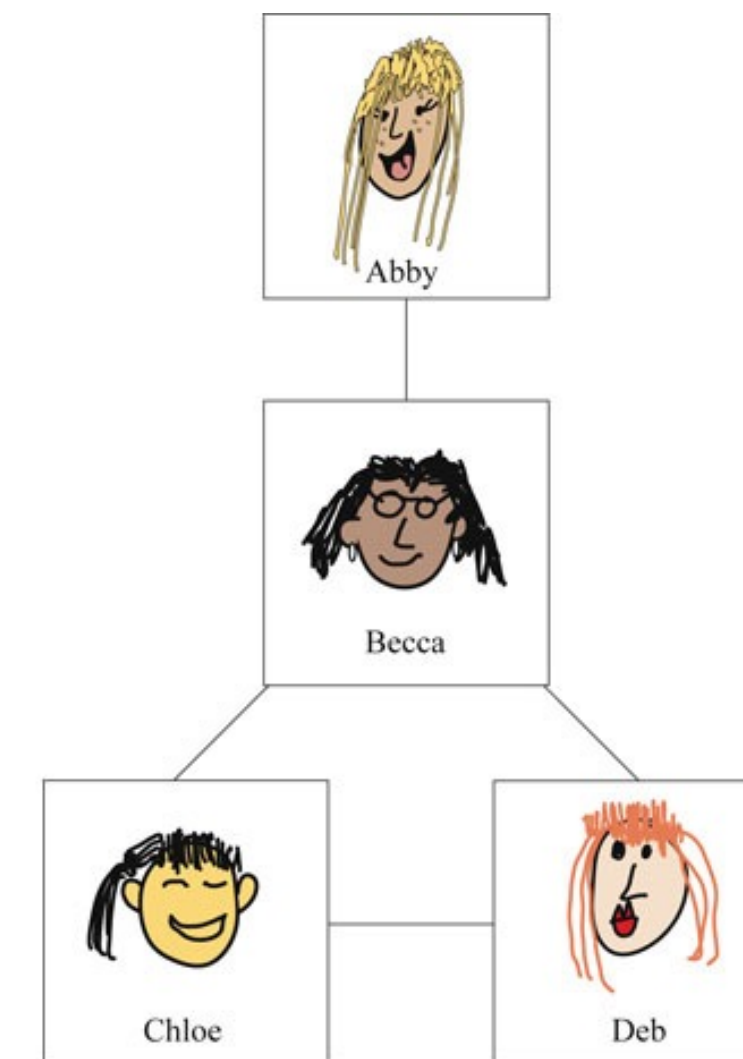
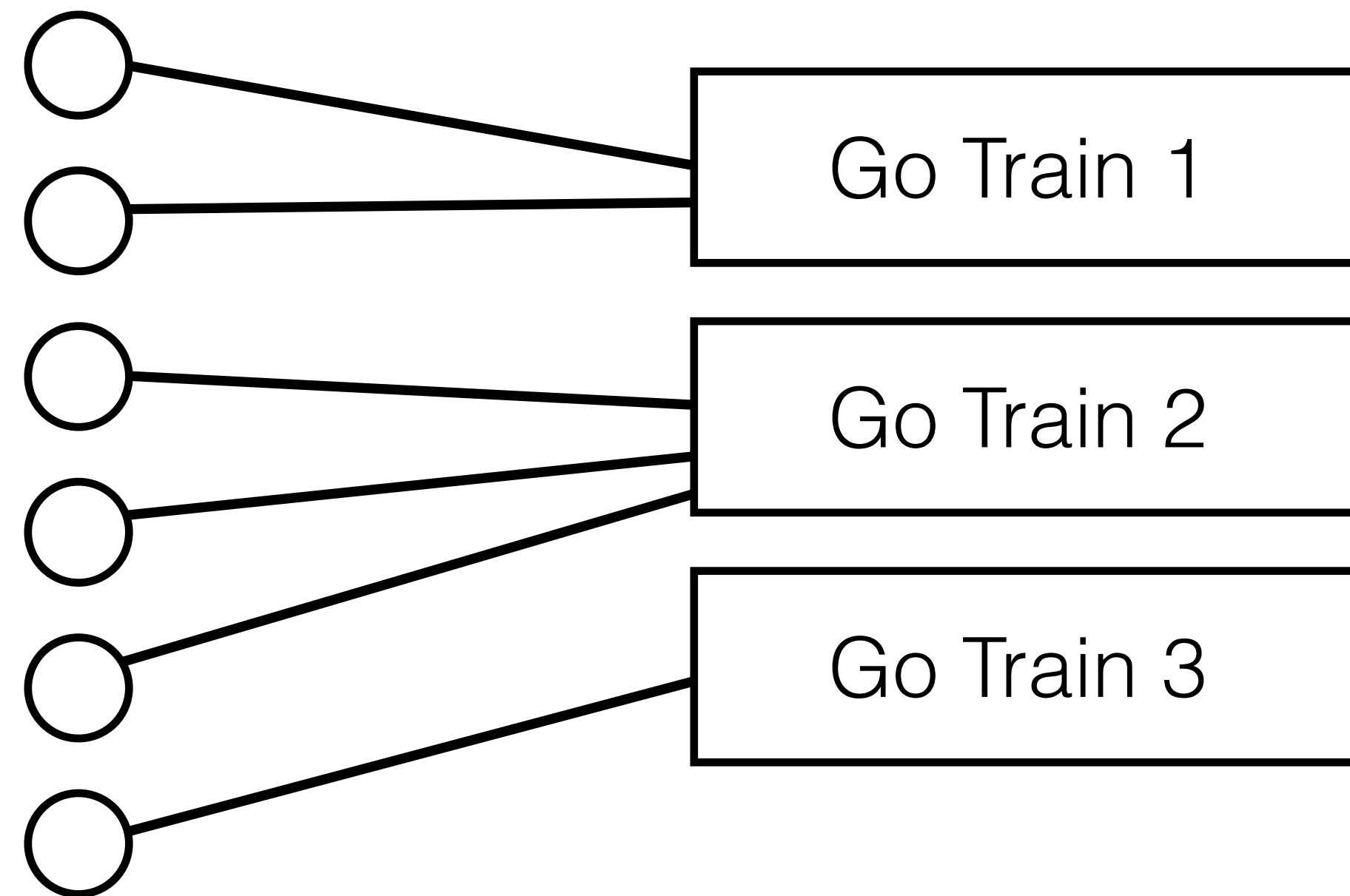
- Avg friend count person  $\leq$  Avg friend count of friend
- Avg # on a train  $\leq$  Avg # on “train experience”





# “Friendship paradox”

- Avg friend count person  $\leq$  Avg friend count of friend
- Avg # on a train  $\leq$  Avg # on “train experience”

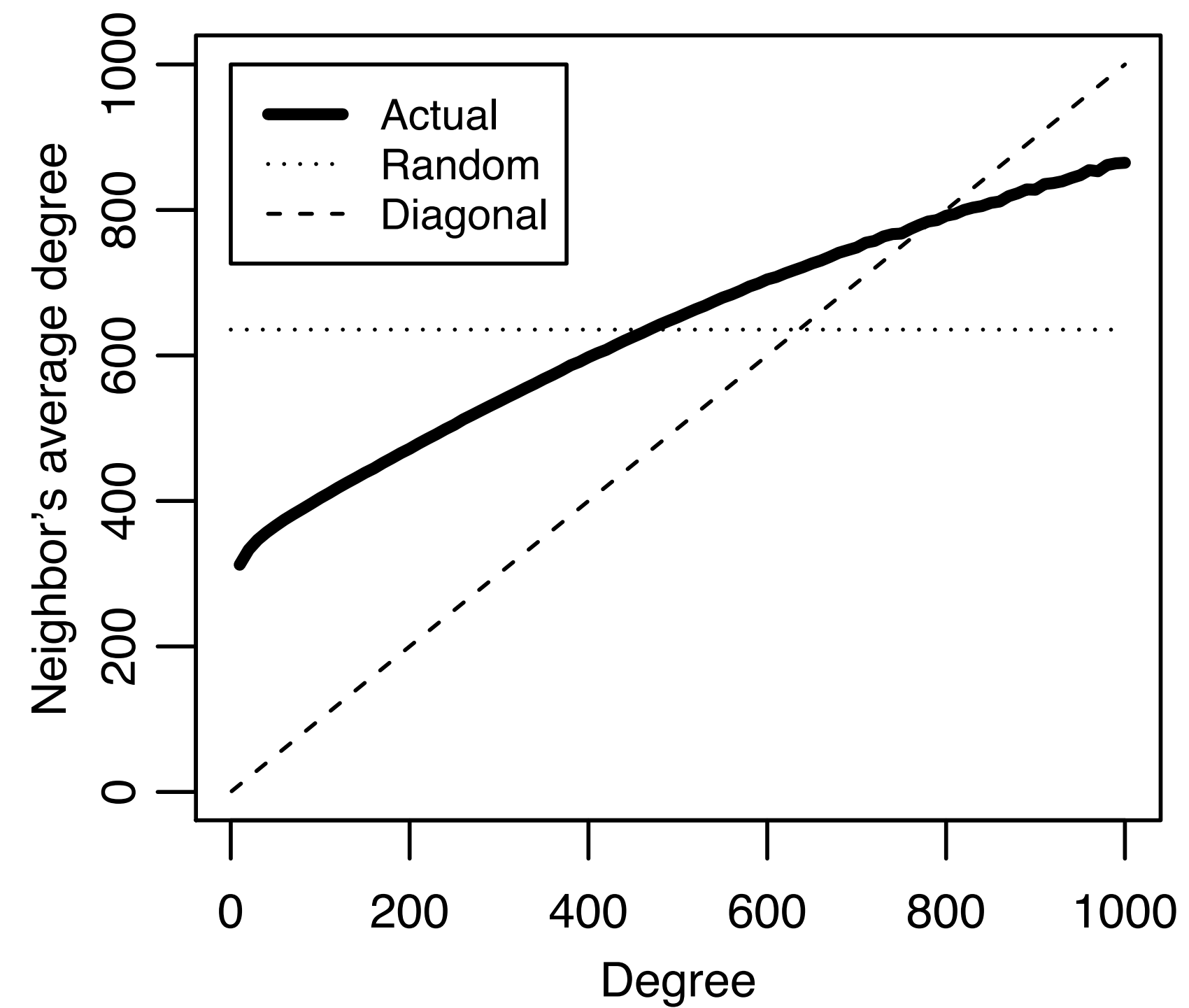


- Basic principle: weighted averages

# “Friendship paradox”

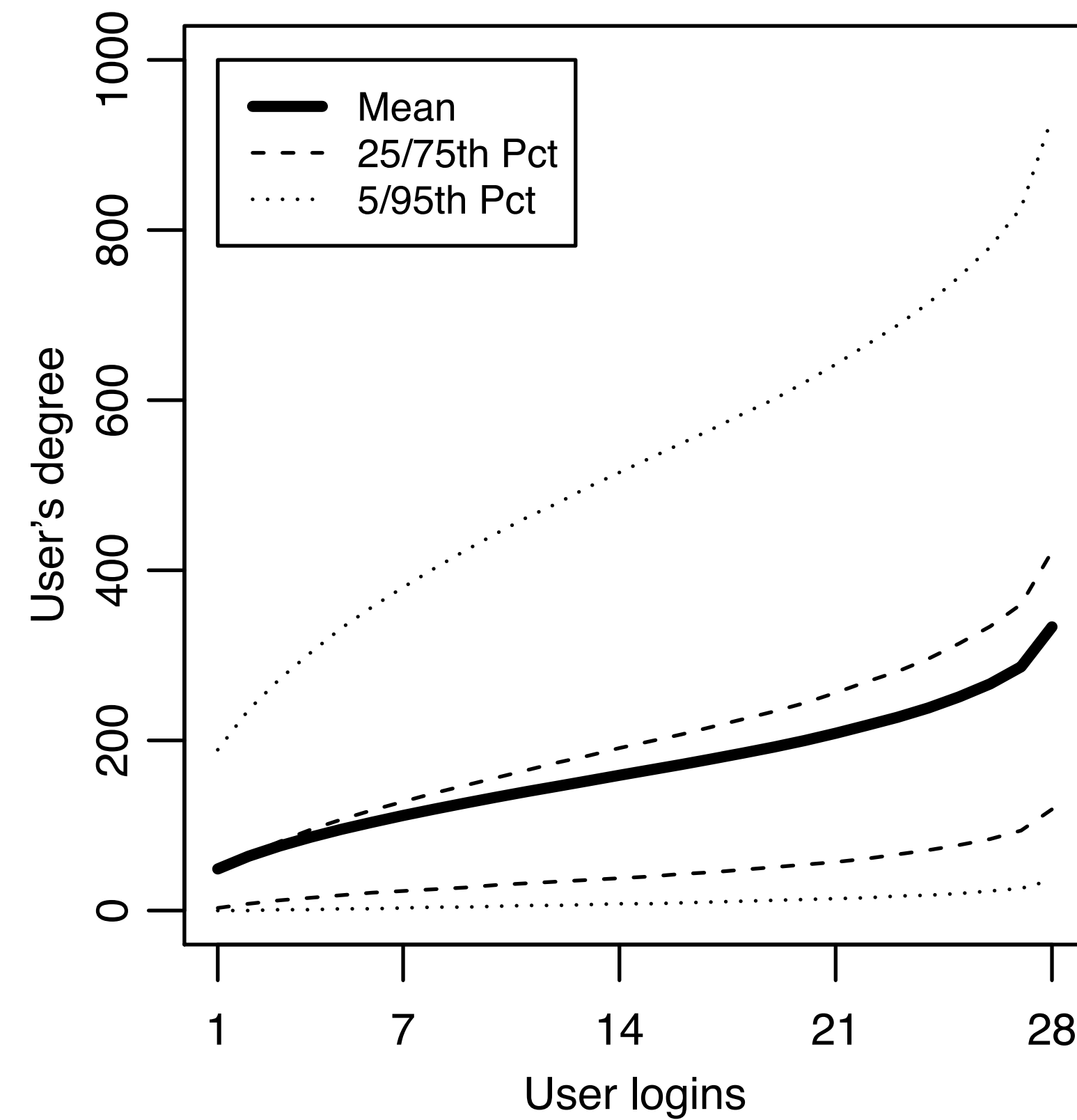
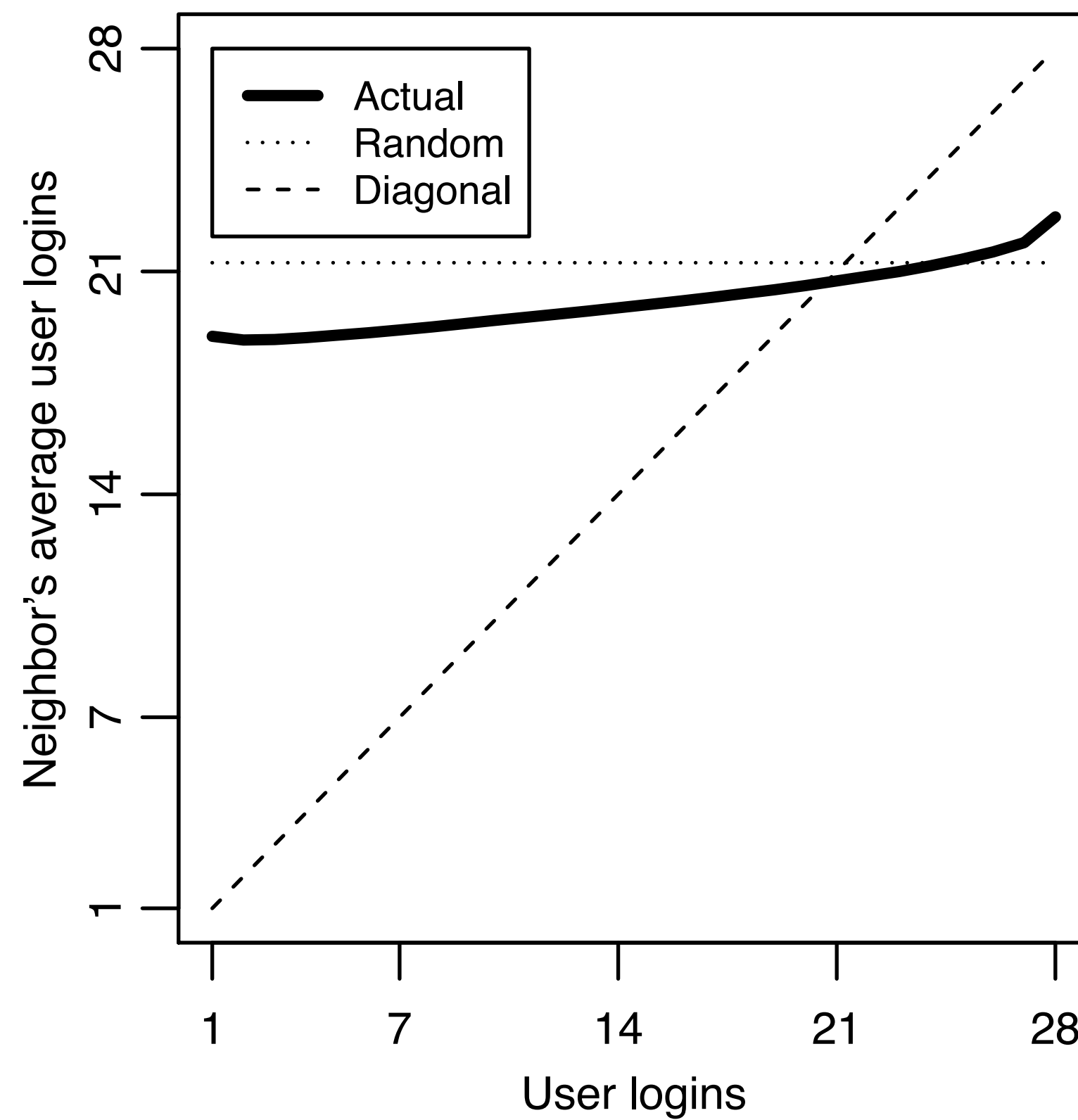
- Friend average =  $\frac{\text{Weighted average}}{\text{Average}}$
- Friend average = Average +  $\frac{\text{Variance}}{\text{Average}}$

# Friendship paradox on FB



# Corollary paradoxes

- “Your friends log in more than you” (and more)





# Friendship paradox

- **Not a social fact!**
  - It's a mathematical fact
  - Applies to virtually any network
  - But it has social implications...
    - Web pages you link to probably have more links
    - People you high-five probably high-five more people than you
    - Etc etc

# Friendship paradox

- **Application: Disease outbreak**
  - Many diseases spread via social networks
  - Model: immunize random friends of random people instead of random people
  - With random people: need to immunize 80-90% of population
  - With random friends of random people: only immunize 20-40% of population
  - We'll study contagion in later weeks