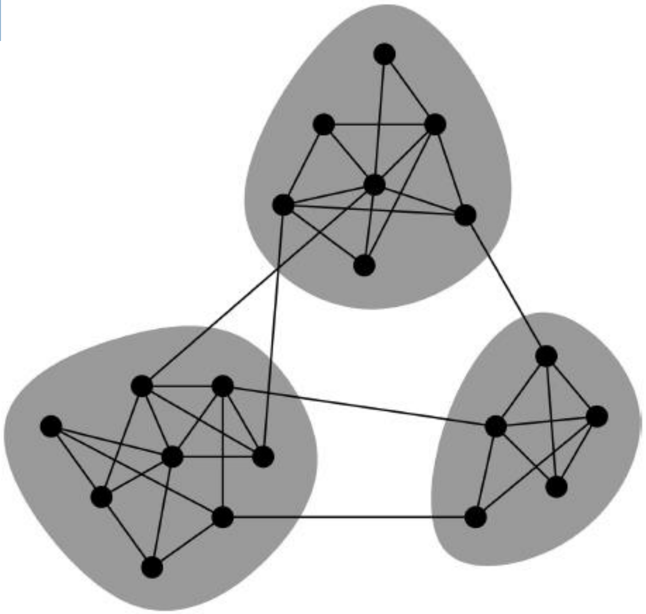
NETWORK COMMUNITIES

Network Communities

- networks are composed of **tightly connected sets of nodes**



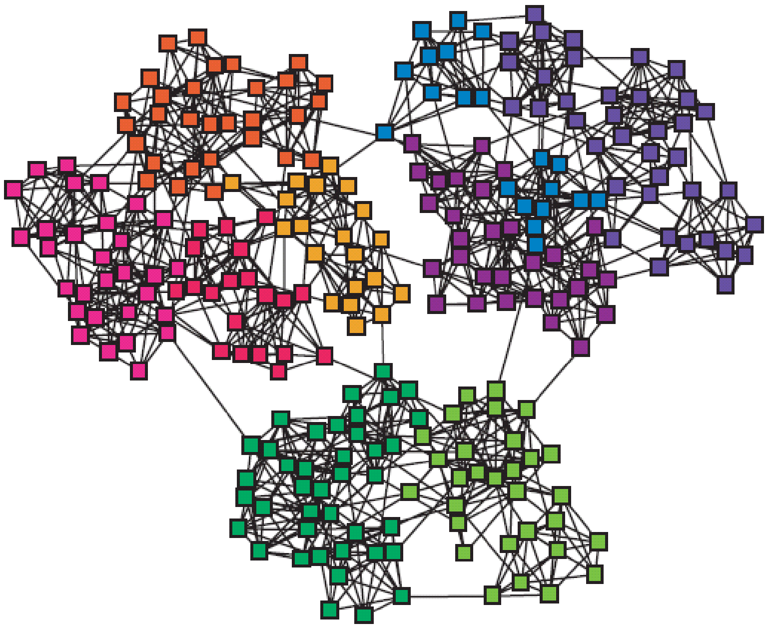
- **Network communities:**
  - Sets of nodes with **lots** of conn… **outside** (the rest of the network)

Communities, clusters, groups, modules

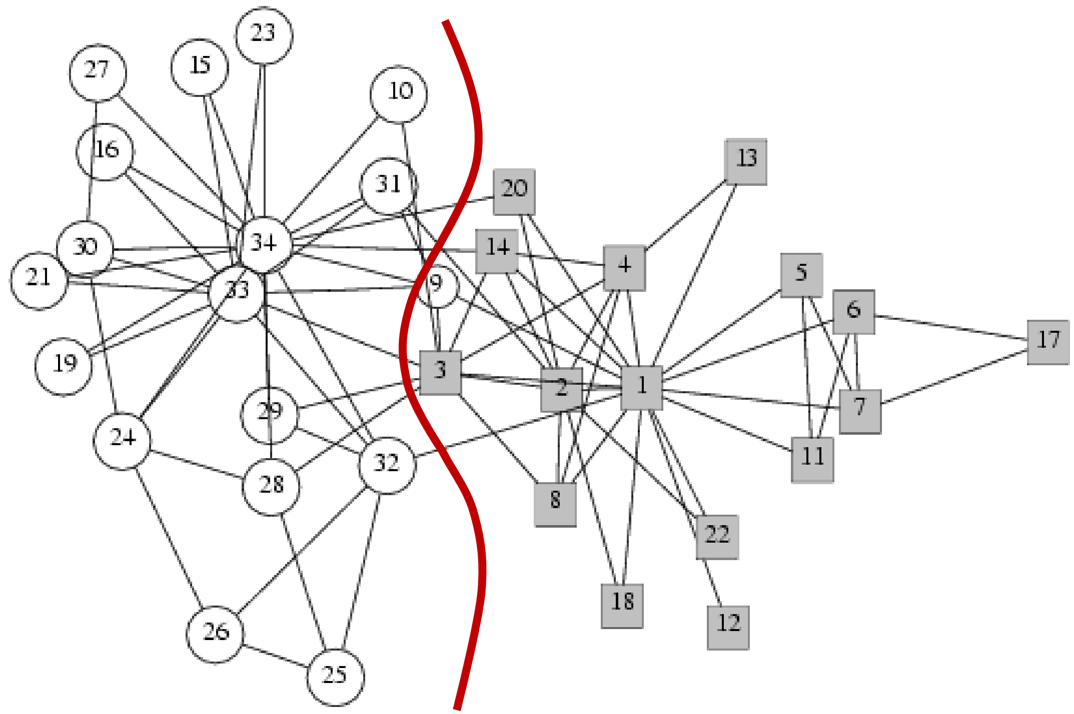
Finding Network Communities

- **How to automatically find such densely connected groups of nodes?**

- Ideally such automatically detected clusters would then correspond to real groups



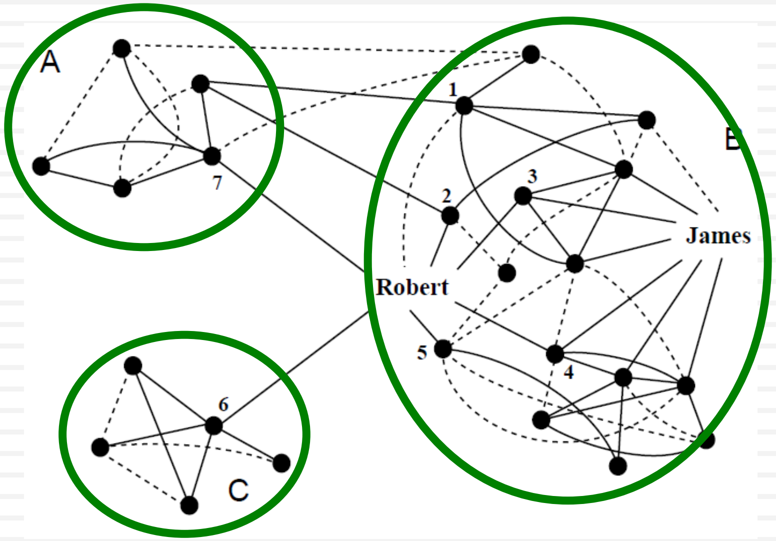
Social Network Data

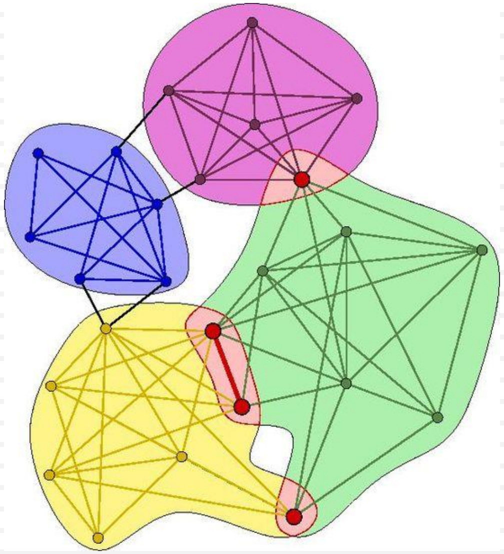


- **Zachary's Karate club network:**
  - Observe social ties and rivalries in a university karate club
  - During his observation, conflicts led the group to split

Community Detection

How to find communities?

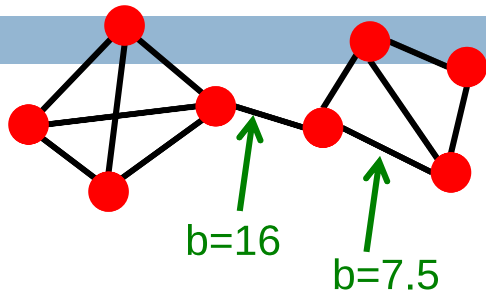


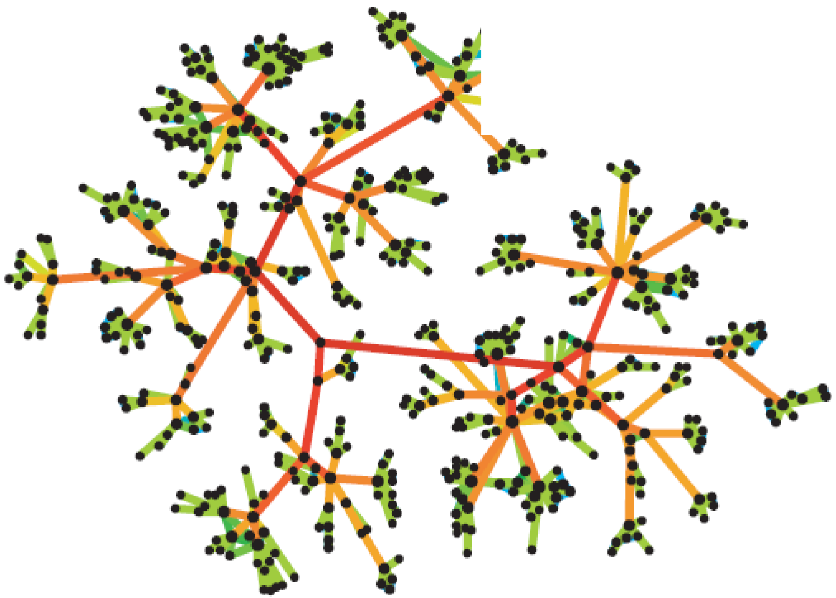


We will work with **undirected** (unweighted) networks

Strength of Weak Ties

- **Edge betweenness:** Number of shortest paths passing over the edge



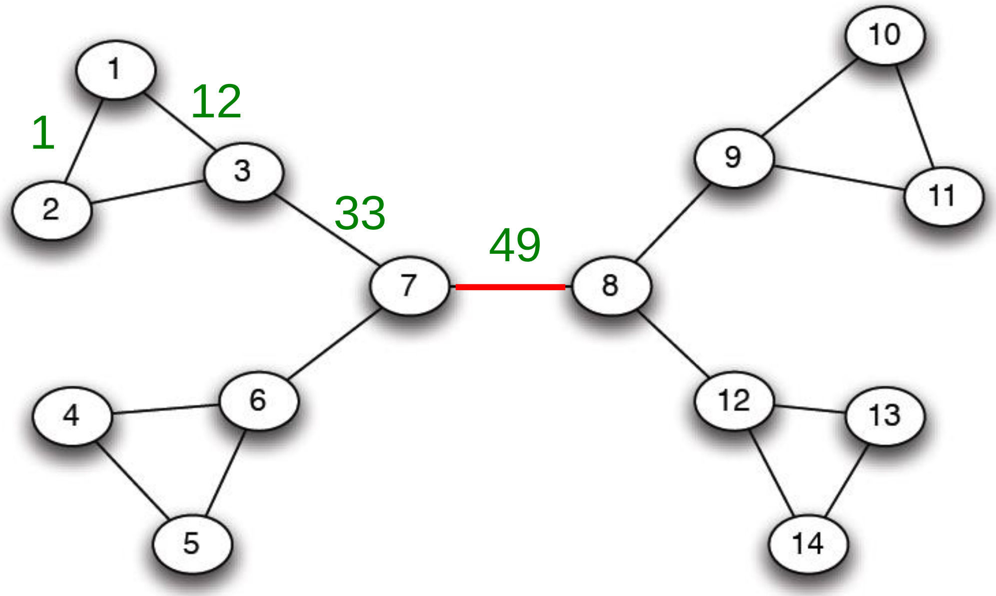


Edge betweenness in real network

Girvan-Newman

- Divisive hierarchical clustering based on the notion of edge **betweenness:**
  - **Number of shortest paths passing through the edge**
- **Girvan-Newman Algorithm:**
  - **Undirected unweighted networks**
  - Repeat until no edges are left:
    - Calculate betweenness of edges
    - Remove the edge with the highest betweenness (if two or more edges tie for highest score, remove all of them)
  - Connected components are communities
  - Gives a hierarchical decomposition of the network

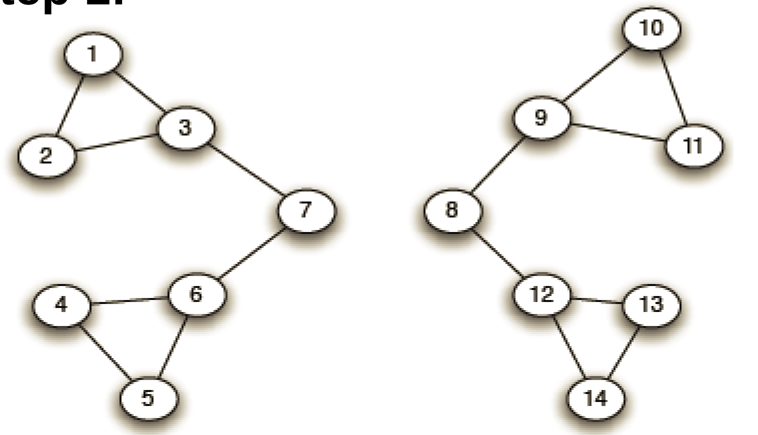
Girvan-Newman: Example



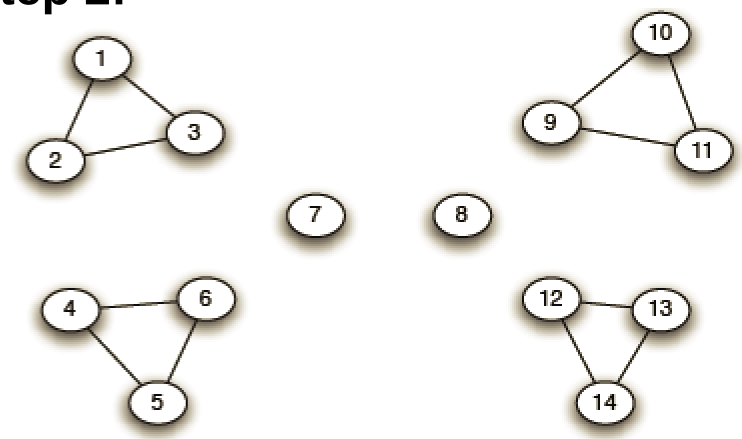
Need to re-compute betweenness at every step

Girvan-Newman: Example

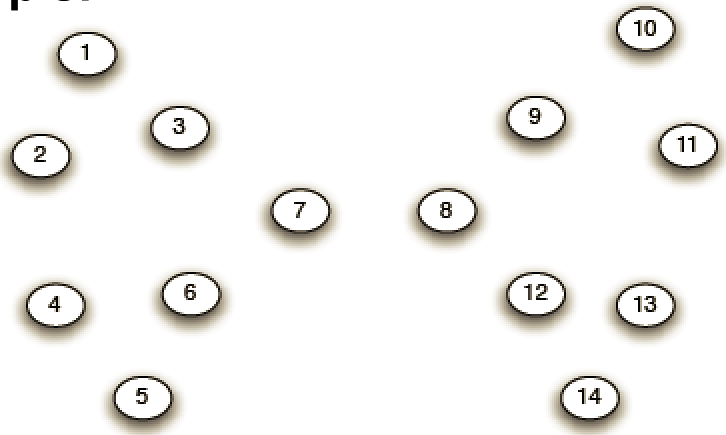
Step 1:



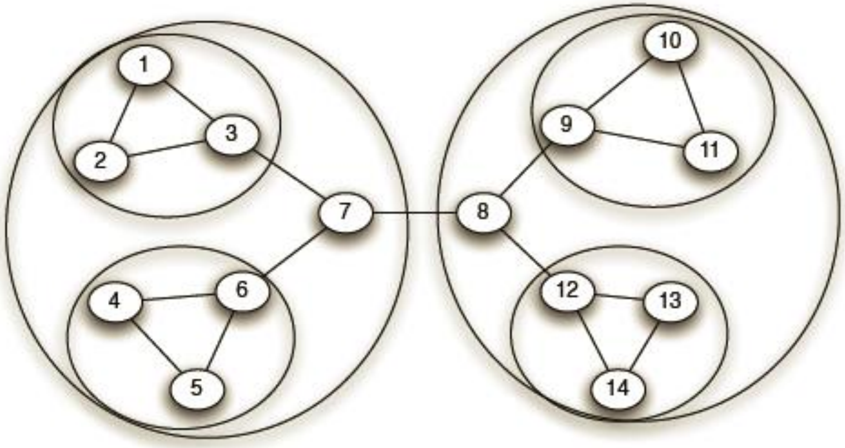
Step 2:



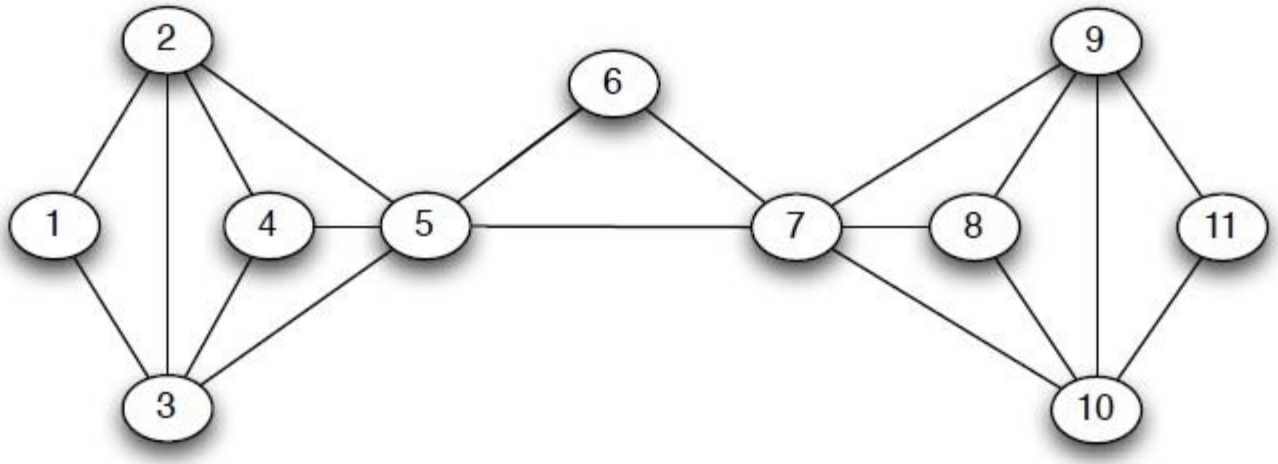
Step 3:



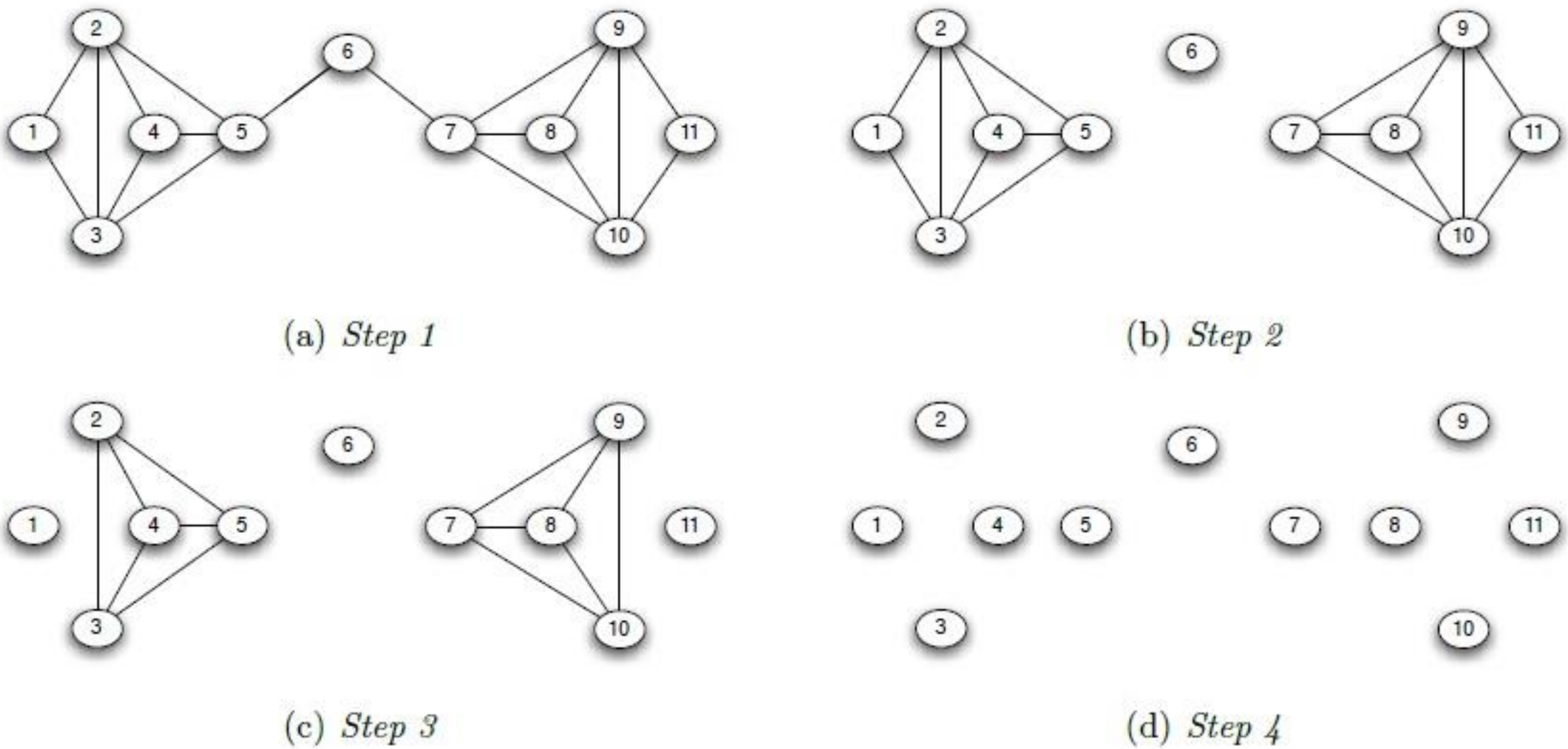
Hierarchical network decomposition:



Girvan-Newman: Example 2



Girvan-Newman: Example 2



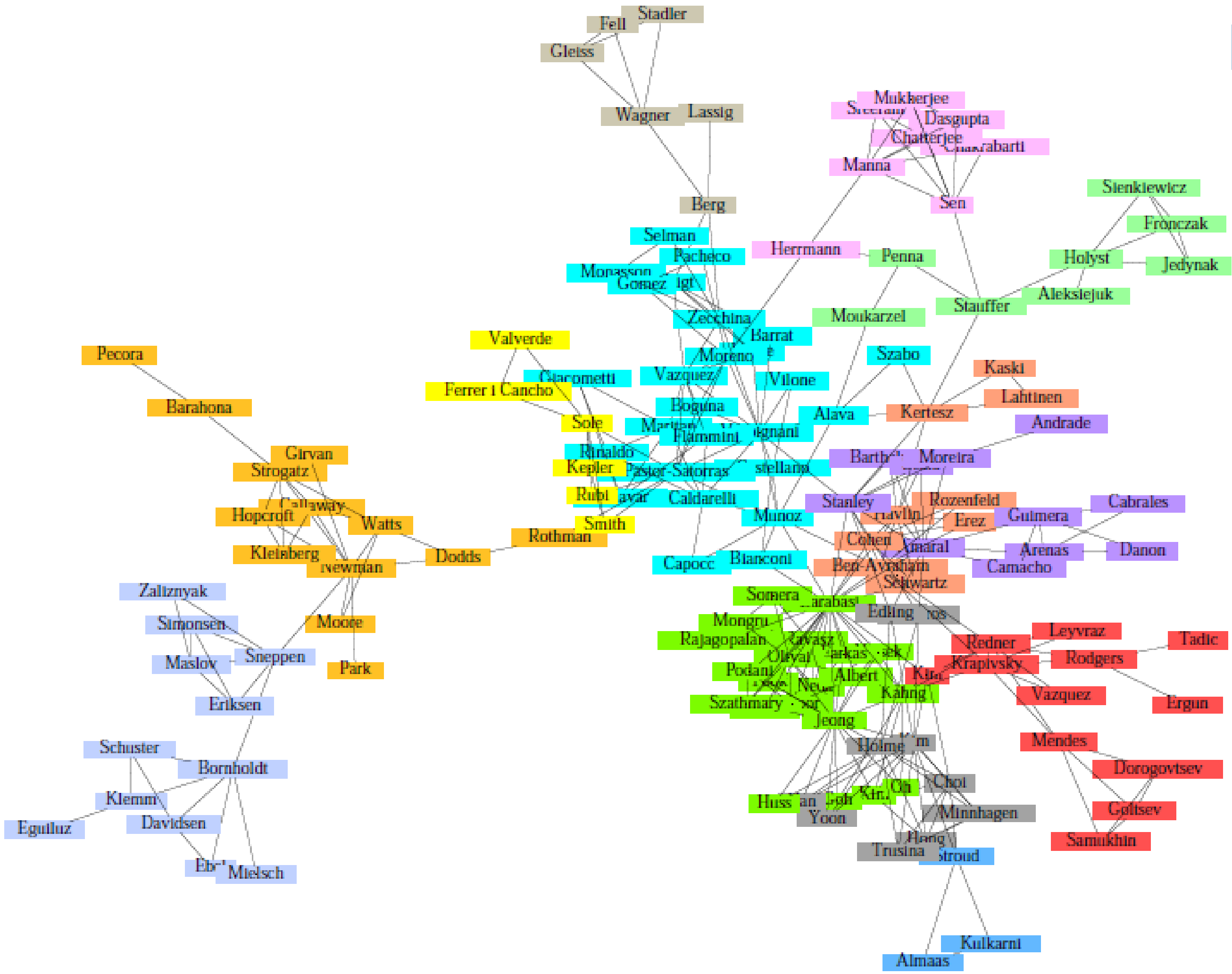
(a) *Step 1*

(b) *Step 2*

(c) *Step 3*

(d) *Step 4*

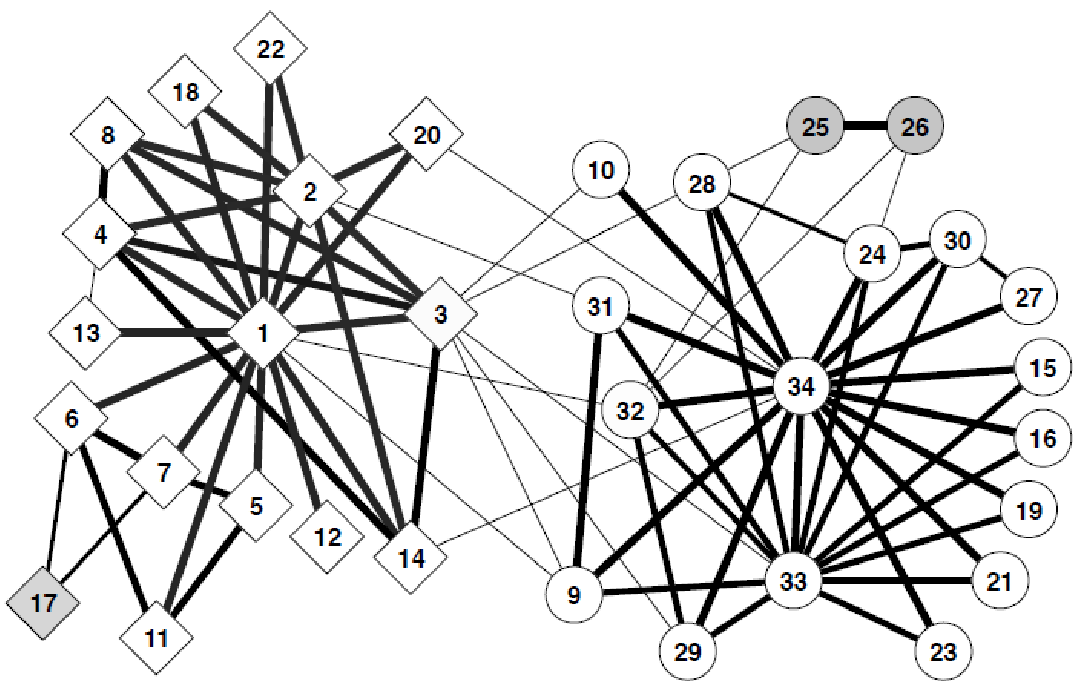
Girvan-Newman: Results

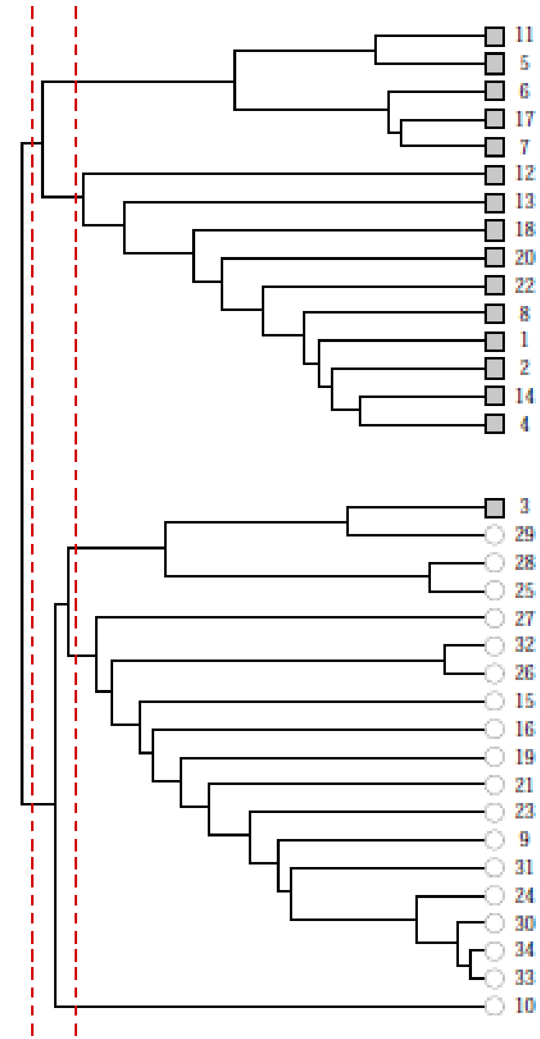


Communities in physics collaborations

Girvan-Newman: Results

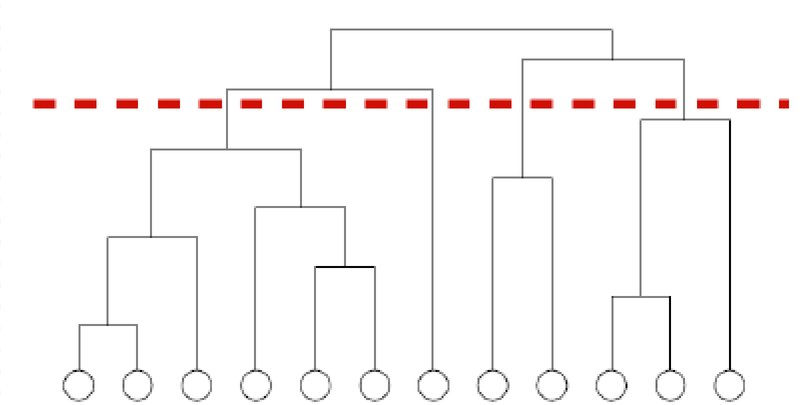
- **Zachary's Karate club:** Hierarchical decomposition





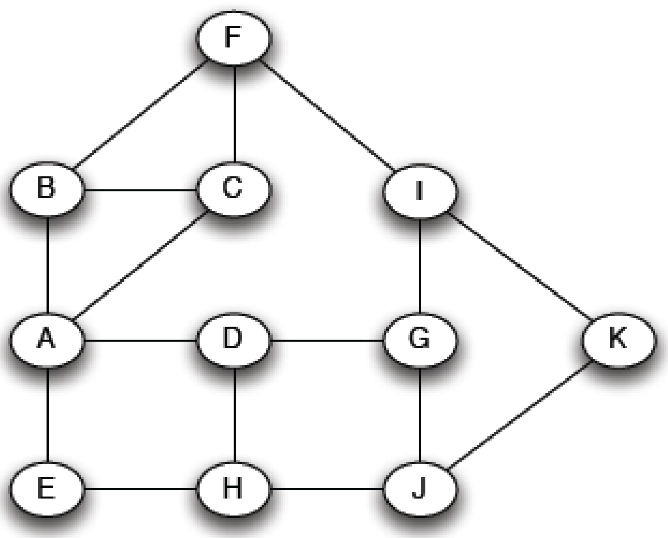
We need to resolve 2 questions

1. **How to compute betweenness?**
2. **How to select the number of clusters?**

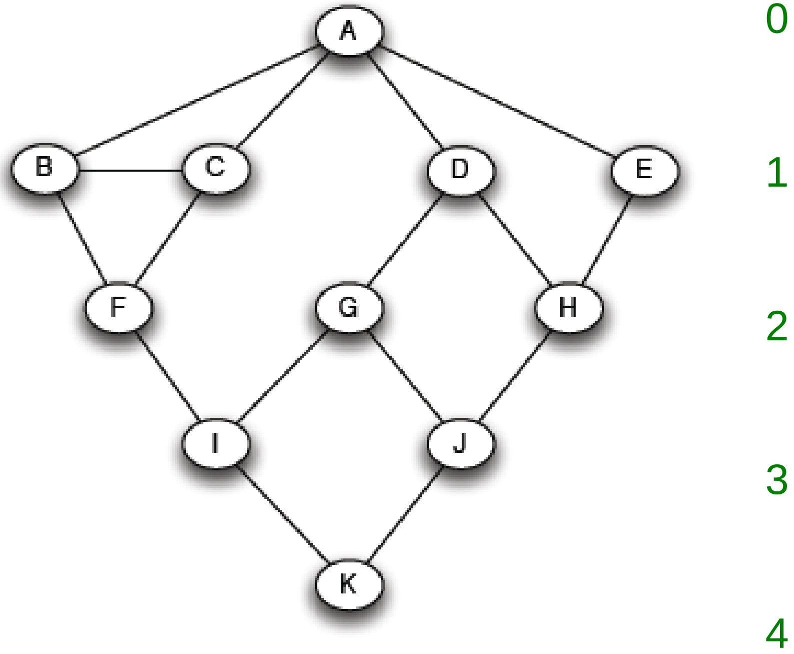


How to Compute Betweenness?

- **Want to compute betweenness of paths starting at node A**

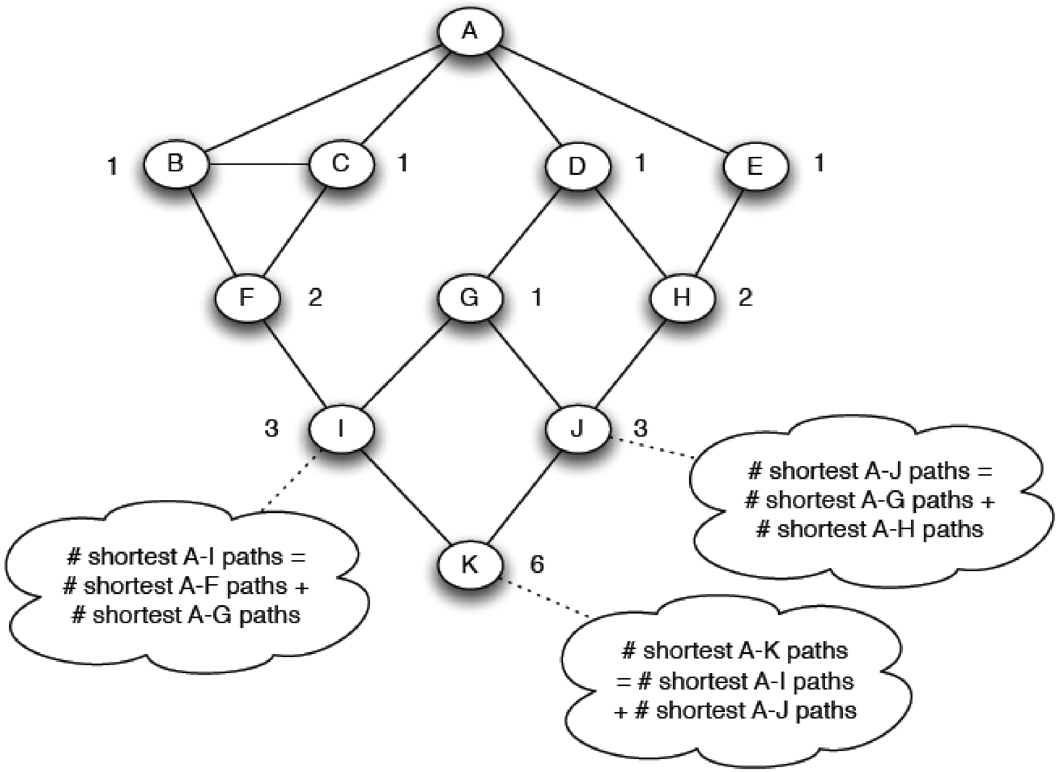


- **Breath first search starting from A:**



How to Compute Betweenness?

- **Count the number of shortest paths from *A* to all other nodes of the network:**
  - Start from the first layer



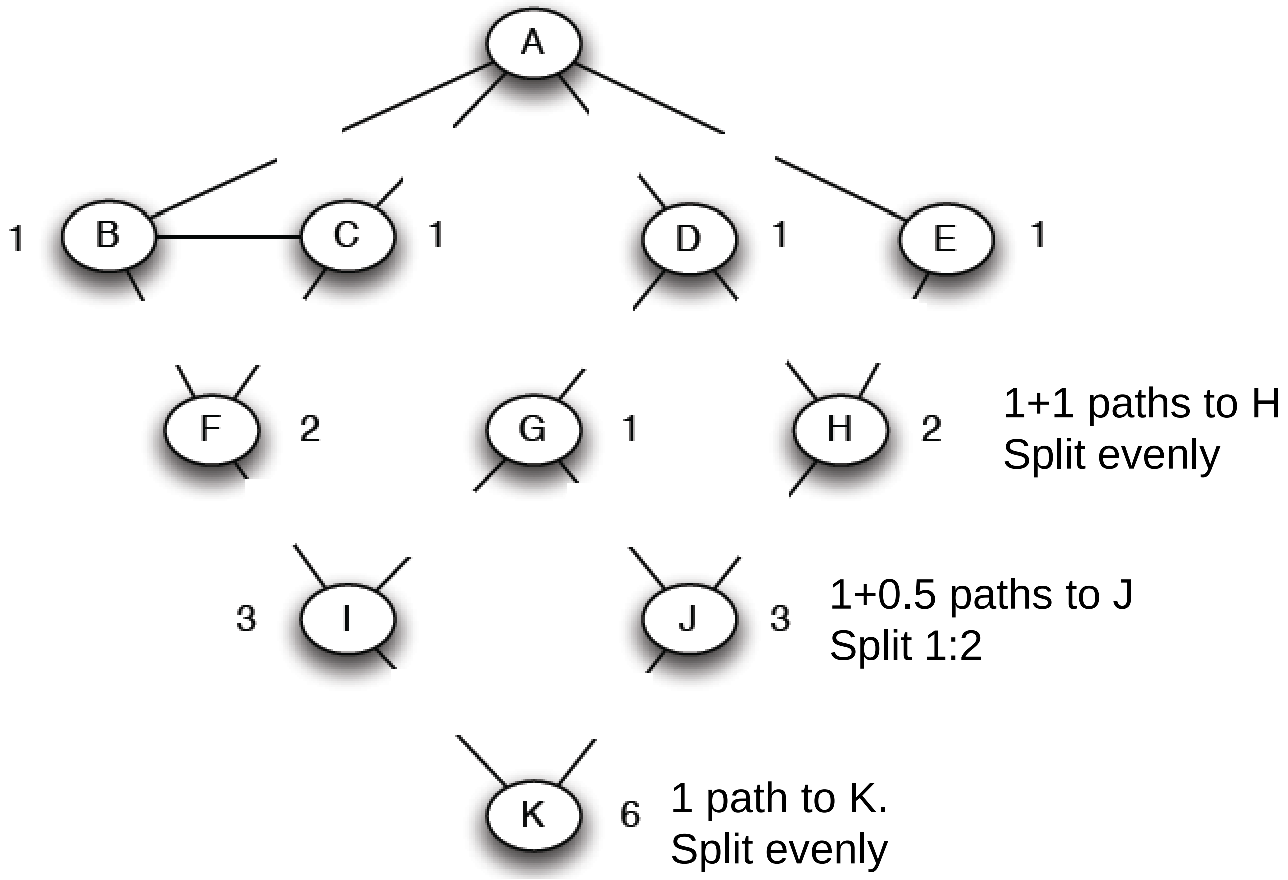
How to Compute Betweenness?

- **Compute betweenness by working up the tree**
- Each node other than the root is given a credit of 1, representing the shortest path to that node
- This credit may be divided among nodes and edges above, since there could be several different shortest paths to the node.

How to Compute Betweenness?

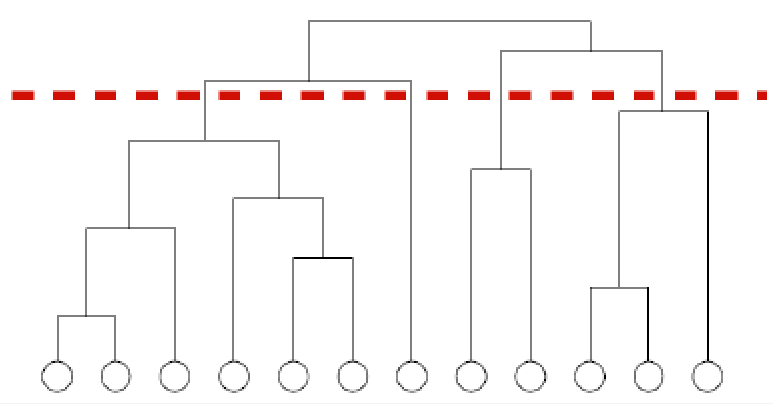
- **Compute betweenness by working up the tree:** If there are multiple paths count them fractionally

The algorithm:
•Add edge **flows**:
  -- node flow =
      1+∑child edges
  -- split the flow up based on the parent value
• Repeat the BFS procedure for each starting node *U*
•Sum up the flow values to get the betweenness
•Divide everything by two



We need to resolve 2 questions

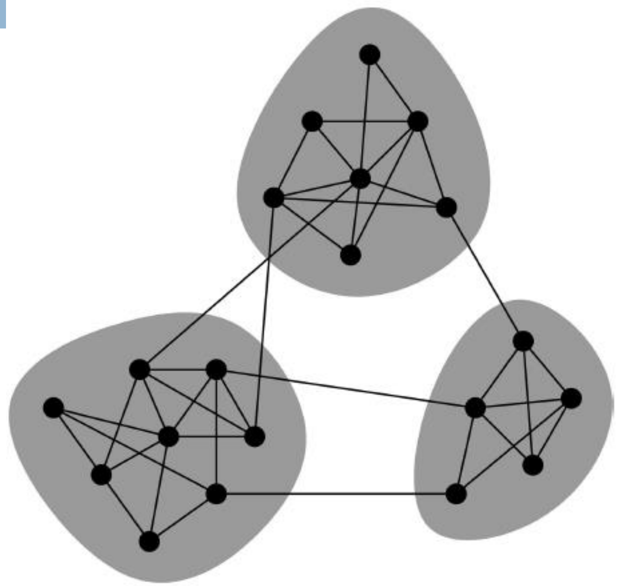
1. How to compute betweenness?
2. How to select the number of clusters?



Network Communities

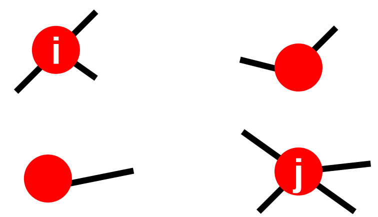
- **Communities:** sets of **tightly connected nodes**
- Define: **Modularity *Q***
  - A measure of how well a network is partitioned into communities
  - Given a partitioning of the network into groups $s \in S$:

$$Q \propto \sum_{s \in S} [\ (\#\text{ edges within group } s) - \underbrace{(\text{expected }\#\text{ edges within group } s)}_{\textbf{Need a null model!}}\ ]$$



Null Model

- Same degree distribution but random connections



- The expected number of edge between nodes i and j of degrees k_i and k_j equals to: $\frac{k_i k_j}{2m}$
  - m: total number of edges in the network
    - $\frac{1}{2}\sum_{i\in N}\sum_{j\in N}\frac{k_i k_j}{2m} = \frac{1}{2}\cdot\frac{1}{2m}\left(\sum_{i\in N} k_i\right)\left(\sum_{j\in N} k_j\right)$
    - $= \frac{1}{4m} 2m \cdot 2m = m$

Note:
$$\sum_{u\in N} k_u = 2m$$

Modularity

- **Modularity of partitioning S of graph G:**
  - Q ∝ $\sum_{s \in S}$ [(# edges within group *s*) – (expected # edges within group *s*)]
  - $Q(G,S) = \frac{1}{2m} \sum_{s \in S} \sum_{i \in s} \sum_{j \in s} \left(A_{ij} - \frac{k_i k_j}{2m} \right)$ A_{ij} = 1 if $i \rightarrow j$, 0 else

    $Q = 0$: the number of within-community edges is no better than random

  - It is positive if the number of edges within groups exceeds the expected number
    - possible presence of community structure
  - 0.3<Q<0.7 means significant community structure

Modularity: Number of clusters

- Modularity is useful for selecting the number of clusters: