

Finding Communities

Social Networks Analysis and Graph Algorithms

Prof. Carlos Castillo — <https://chato.cl/teach>

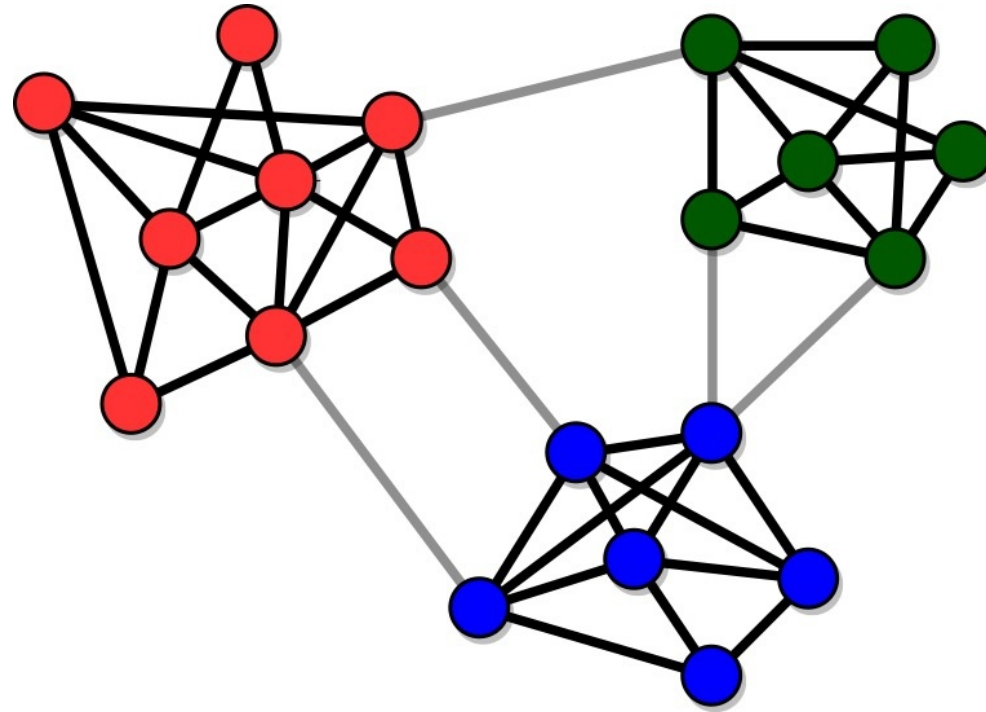


Universitat
Pompeu Fabra
Barcelona

Sources

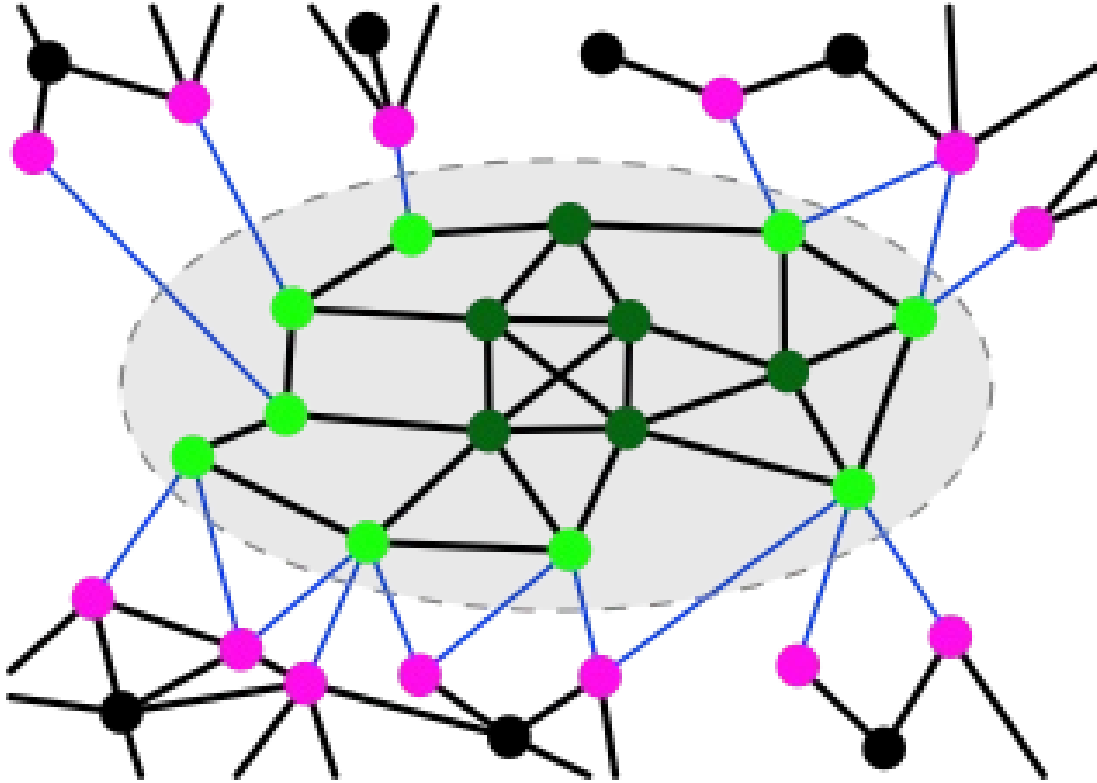
- A. L. Barabási (2016). Network Science – Chapter 09
- D. Easley and J. Kleinberg (2010). Networks, Crowds, and Markets – Chapter 03
- F. Menczer, S. Fortunato, C. A. Davis (2020). A First Course in Network Science – Chapter 06
- URLs cited in the footer of slides

Example with clear community structure



Characterizing one community

Communities are **connected** and **dense**



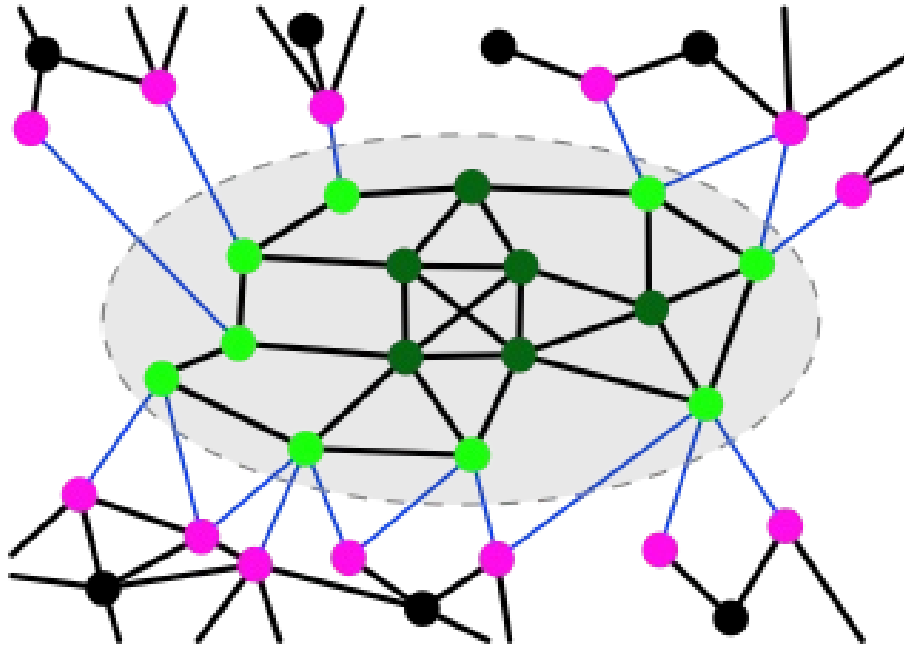
Given a community C

Internal degree $k^{\text{int}}(C)$ considers only nodes inside the community

External degree $k^{\text{ext}}(C)$ considers only nodes outside the community

$$k_i = k_i^{\text{int}}(C) + k_i^{\text{ext}}(C)$$

Strong community

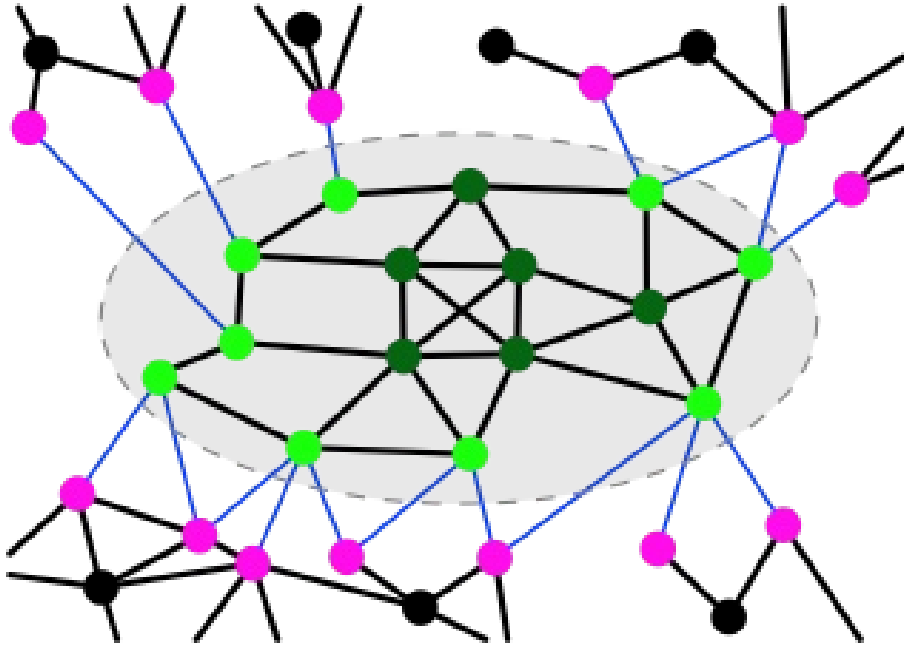


A community C is **strong** if **every** node i within the community satisfies:

$$k_i^{\text{int}}(C) > k_i^{\text{ext}}(C)$$

- Is the community of green nodes (dark green and light green) a strong community?
- What is the difference between dark green and light green nodes?

Weak community



A community C is **weak** if on **aggregate** nodes satisfy:

$$\sum_{i \in C} k_i^{\text{int}}(C) > \sum_{i \in C} k_i^{\text{ext}}(C)$$

- All communities satisfying the strong property satisfy the weak one

Exercise

A community C is **strong** if, for all nodes i within the community:

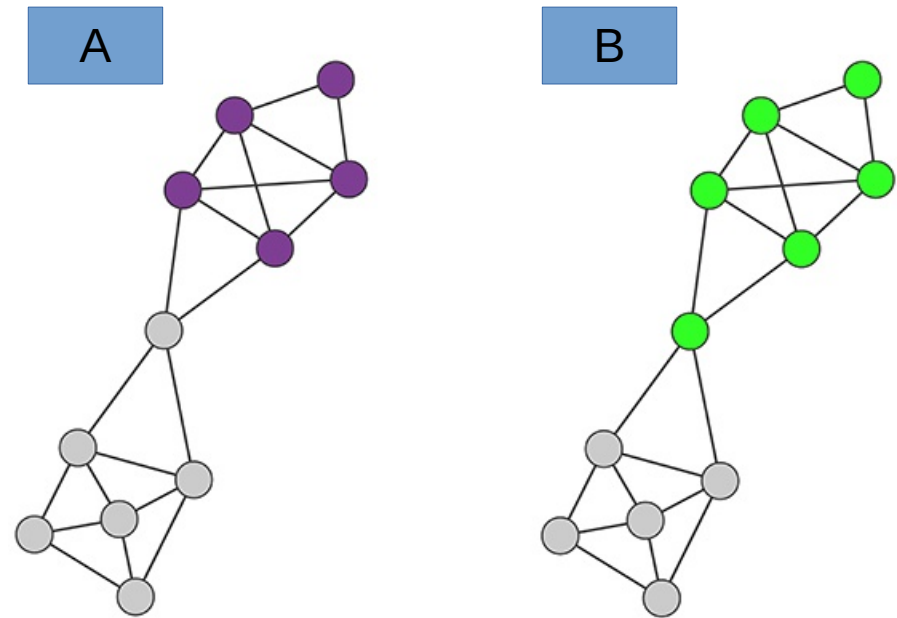
$$k_i^{\text{int}}(C) > k_i^{\text{ext}}(C)$$

A community C is **weak** if:

$$\sum_{i \in C} k_i^{\text{int}}(C) > \sum_{i \in C} k_i^{\text{ext}}(C)$$

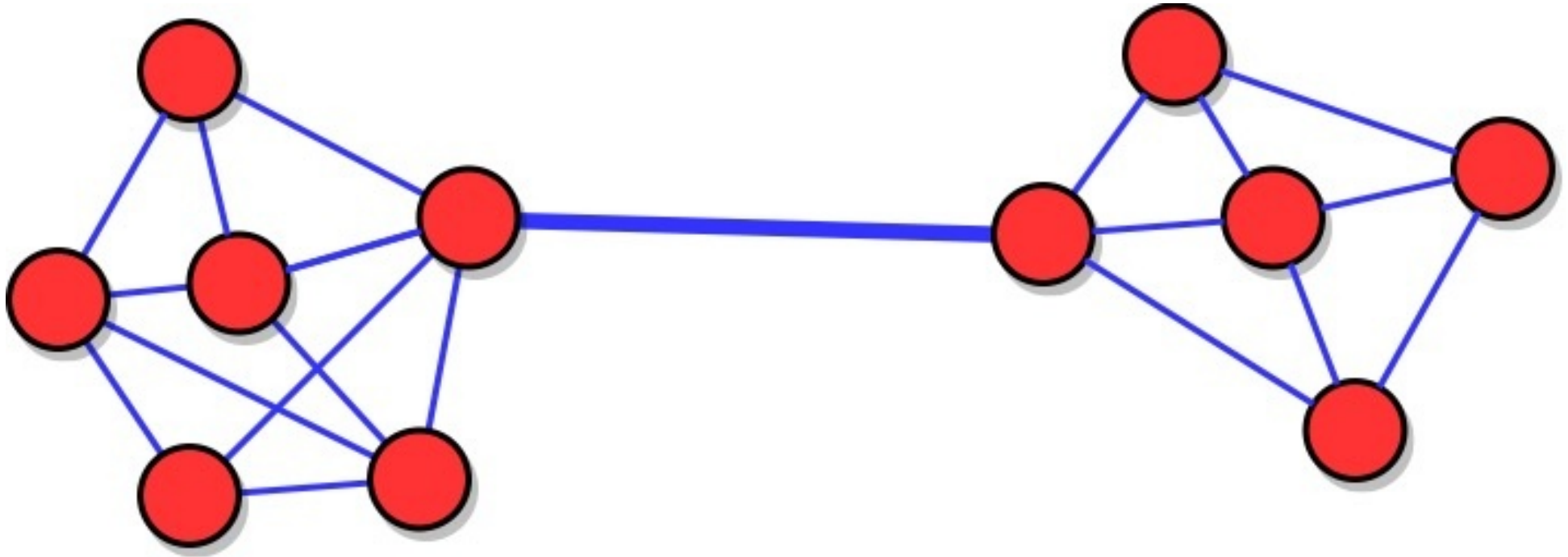
Is **community A** strong, weak, both?

Is **community B** strong, weak, both?

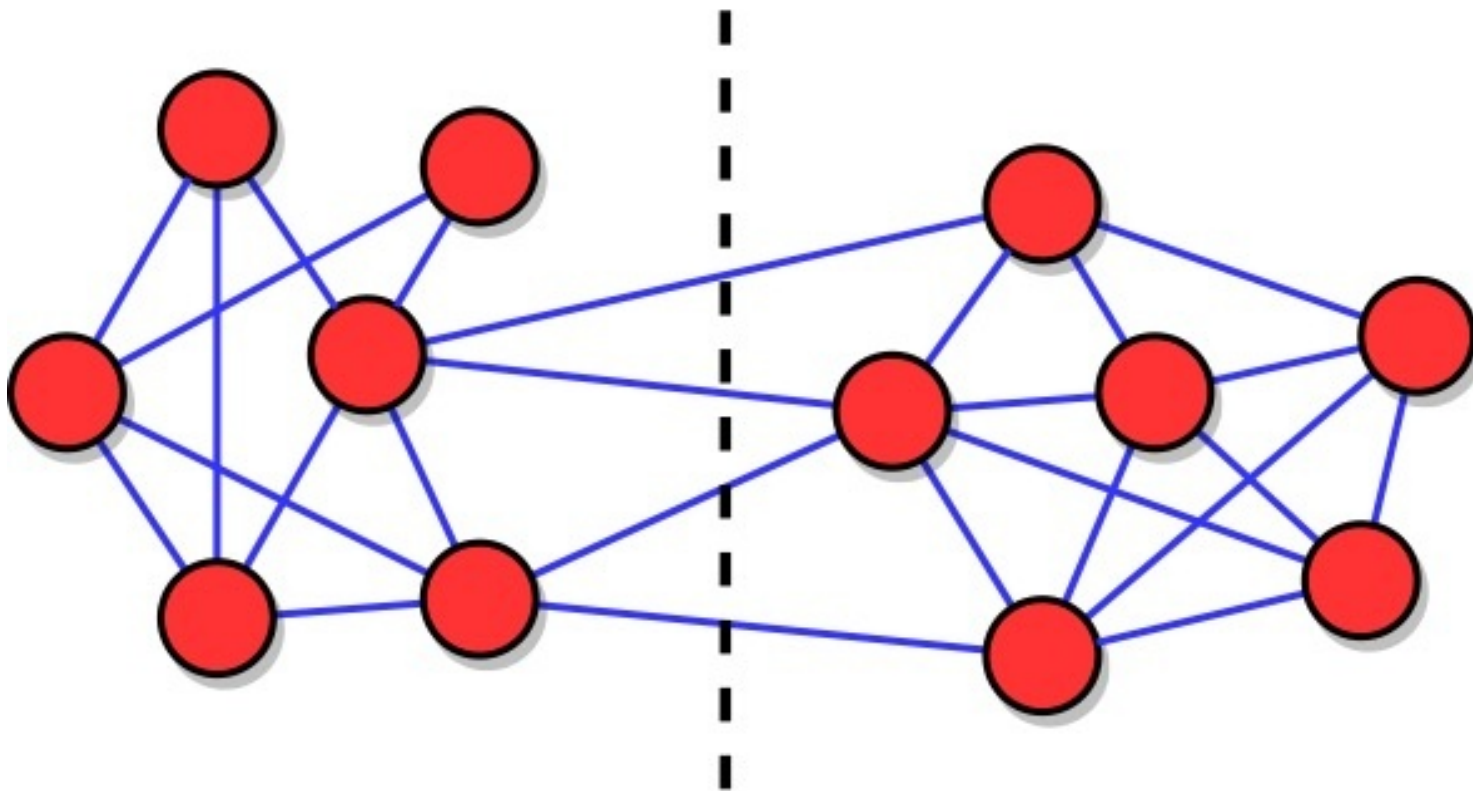


Finding two communities: network bisection

A graph that is easy to bisect



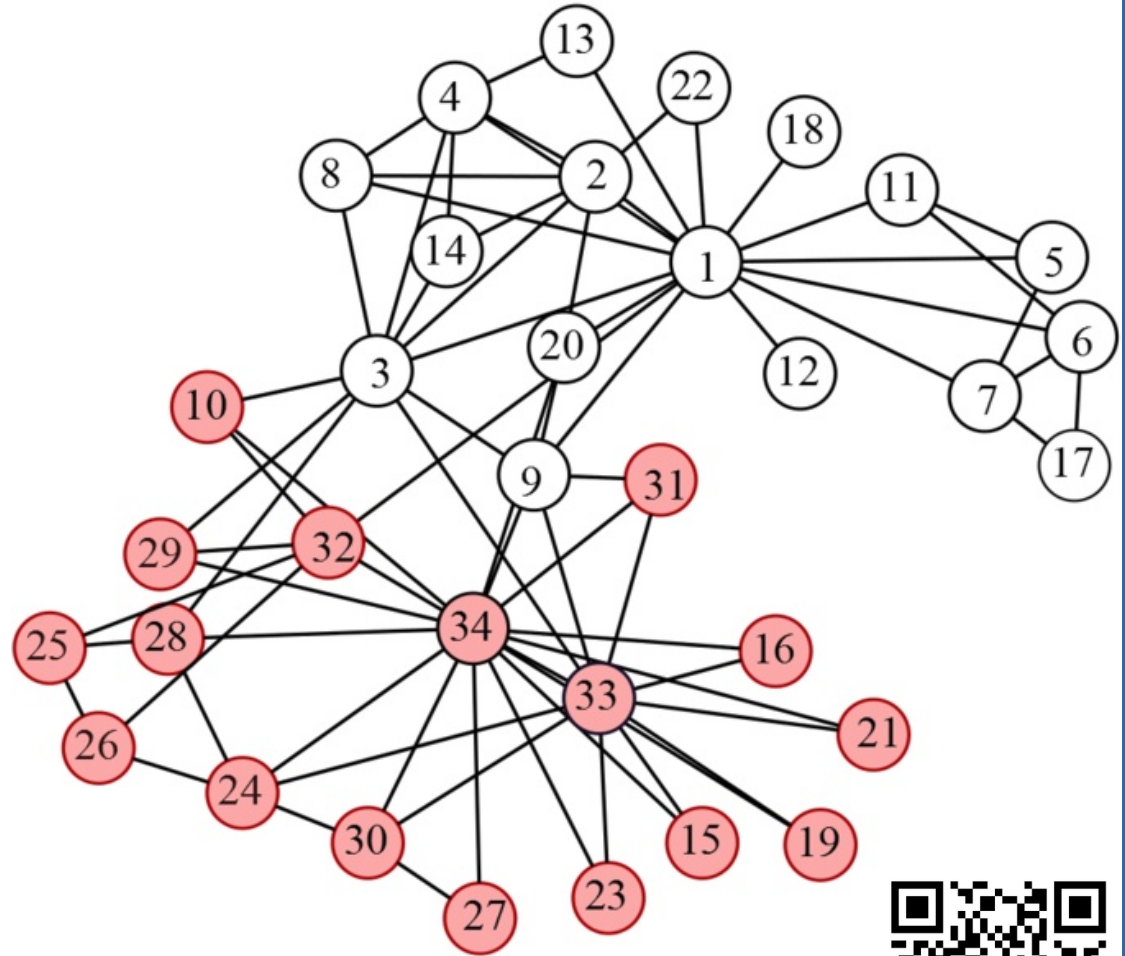
Graph bisection: finding a minimal “cut”



Simple exercise

Cut size under bisection

- What is the size of the white-red cut?
- If node 9 goes to the red component, what is the size of the white-red cut?



Pin board: <https://upfbarcelona.padlet.org/chato/4qz0k8ro0zquen1>

Finding multiple communities: a divisive method

Hierarchical graph partitioning

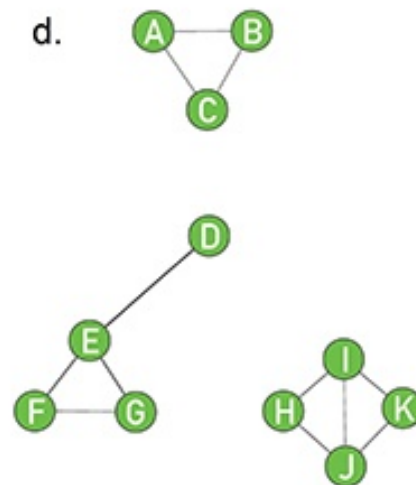
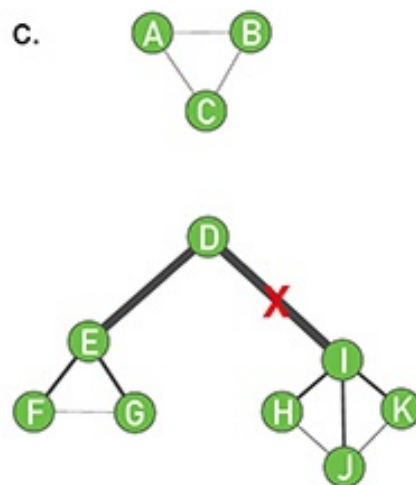
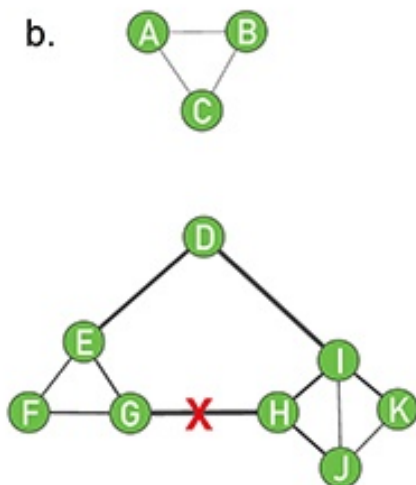
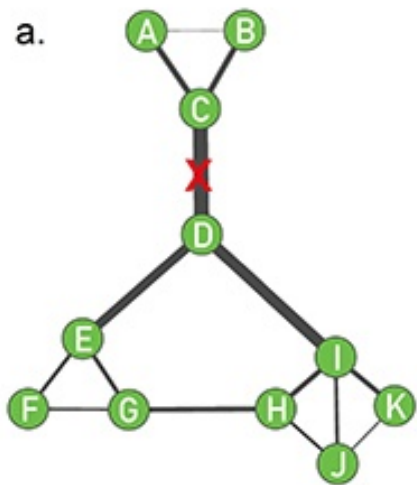
Until there are edges in the graph

Find an edge e that bridges two communities

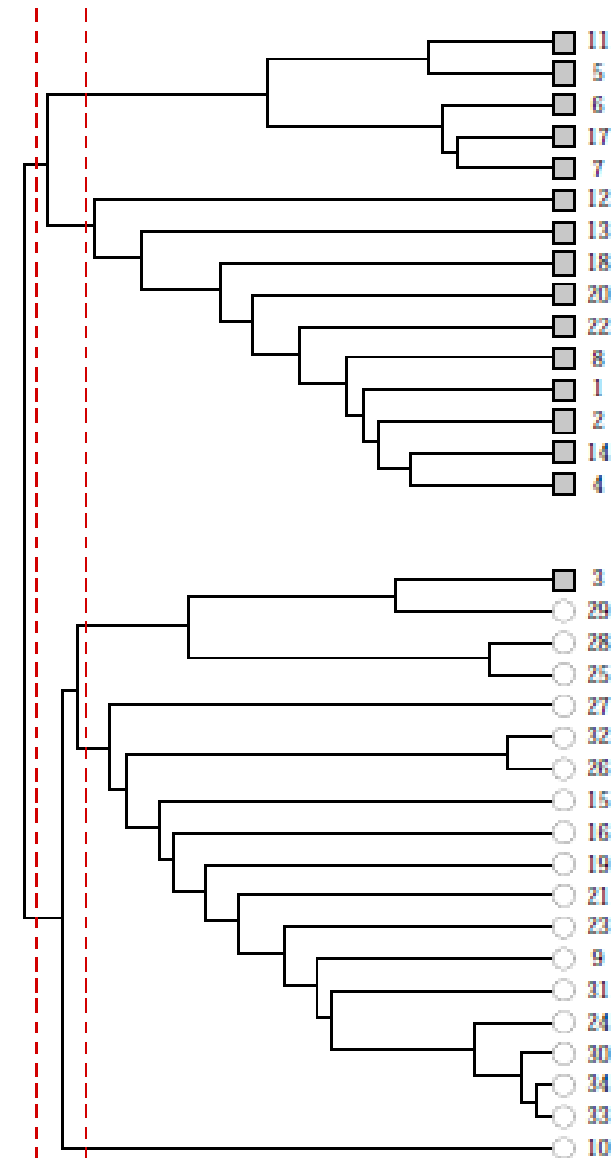
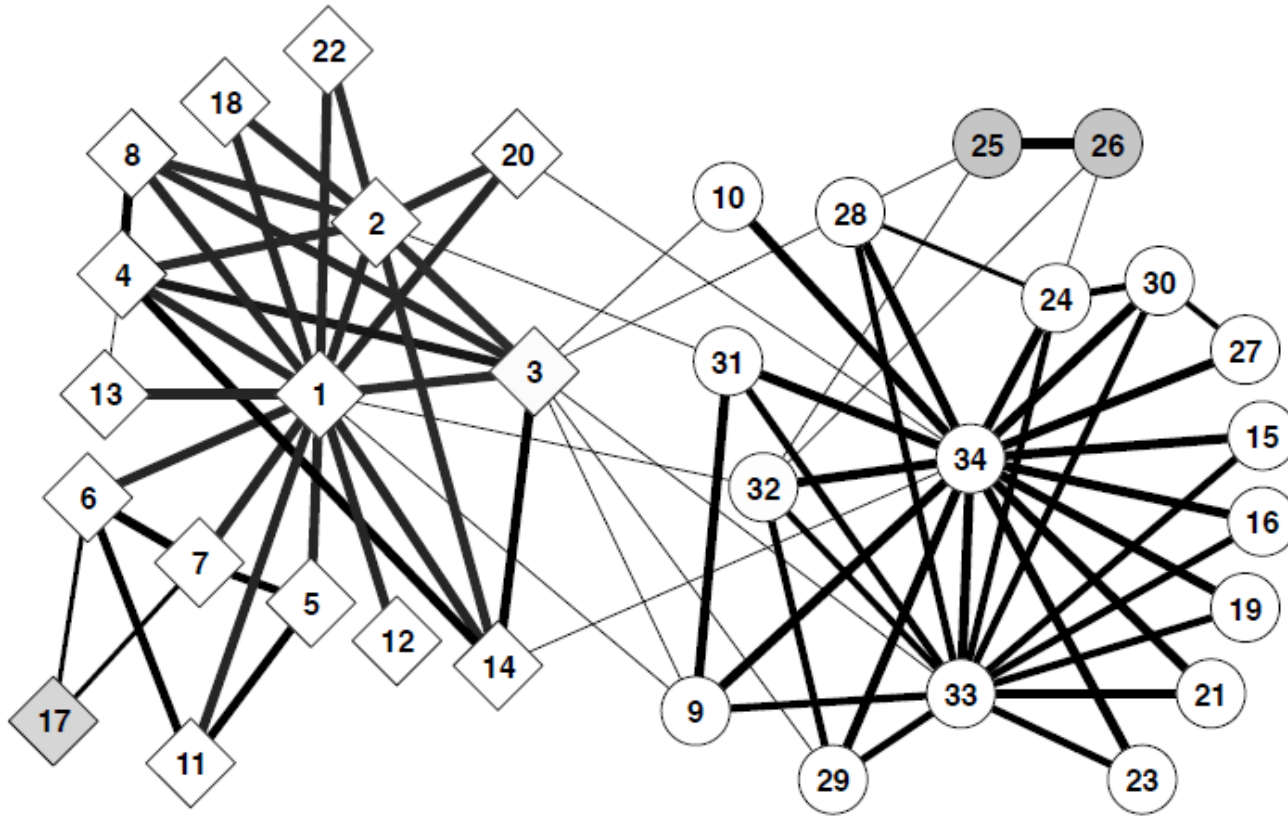
Remove edge e

The Girvan-Newman algorithm

- Repeat:
 - Compute edge betweenness
 - Remove edge with larger betweenness



Example: Karate Club



Quantifying multiple communities: modularity

Measuring a partition in a graph

- **Modularity** (or one of its variants) is a popular method to determine how good a partition is on a graph
- It compares the **observed number of internal links** in each partition, against the **expected number of internal links** if those internal links had been placed at random

Modularity of a partition

$$Q = \frac{1}{L} \sum_C \left(L_C - \frac{k_C^2}{4L} \right)$$

- L = number of links in the network
- L_C = number of internal links in community C
- k_C = sum of degree of nodes in C

Modularity of a partition (cont.)

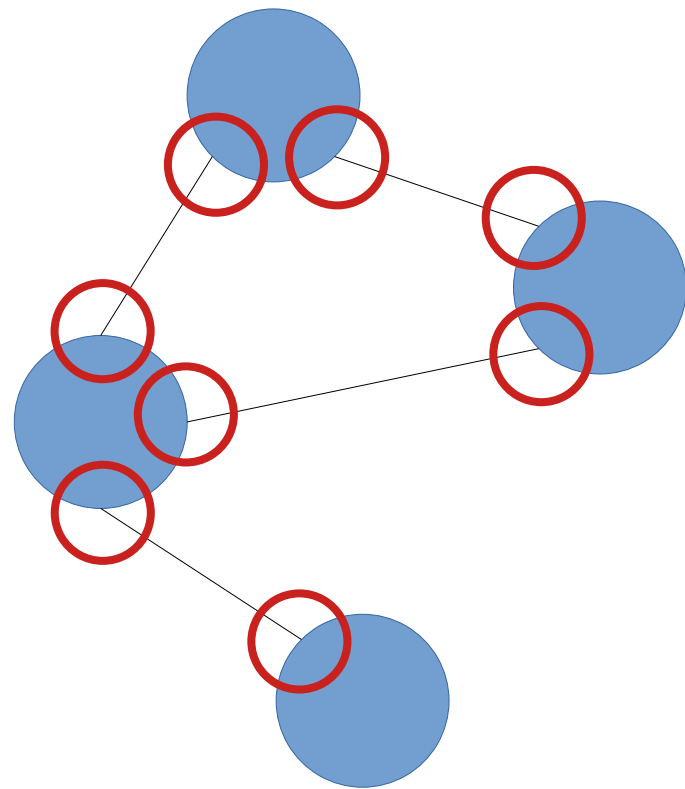
$$Q = \frac{1}{L} \sum_C \left(L_C - \frac{k_C^2}{4L} \right) \longrightarrow$$

Expression in parenthesis is the difference between observed and expected internal links in community C

- L = number of links in the network
- L_C = number of internal links in community C
- k_C = sum of degree of nodes in C
- $k_C^2/4L$ = **expected** number of internal links in community C

Where does $k_c^2/4L$ comes from?

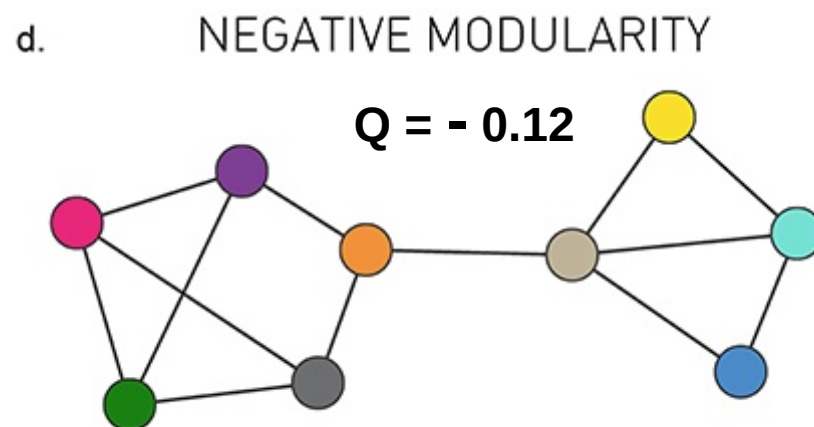
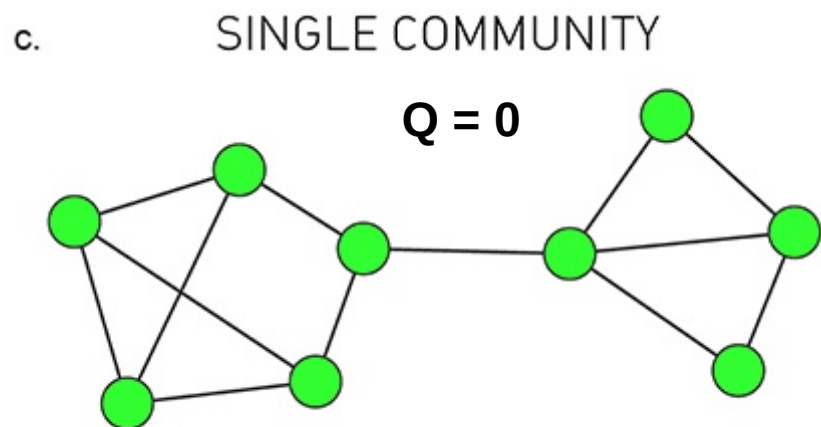
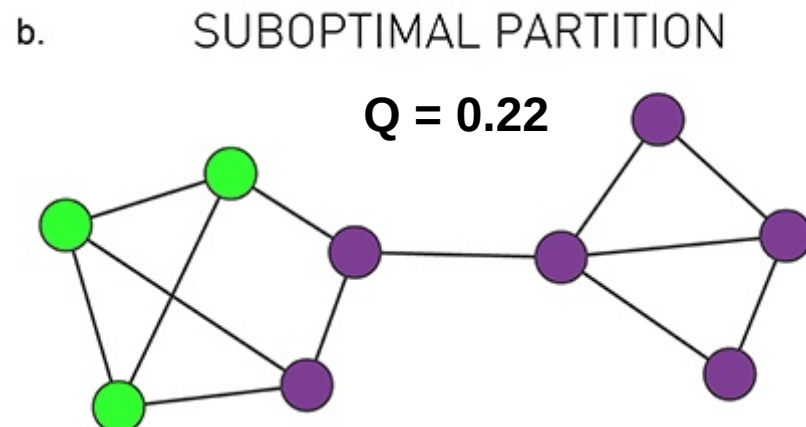
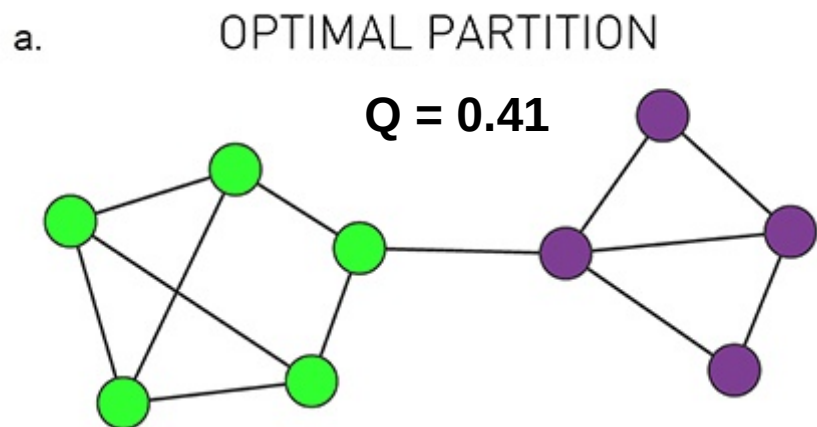
- A **link “stub”** is a connection between a link and a node
- There are $2L$ stubs in a network
- There are as many stubs as the sum of the degree of nodes



Modularity formula explained

$$Q = \frac{1}{L} \sum_C \left(L_C - \frac{k_C^2}{4L} \right)$$

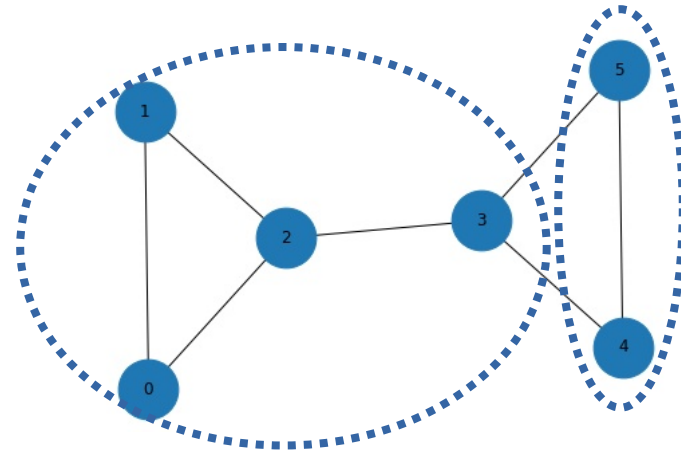
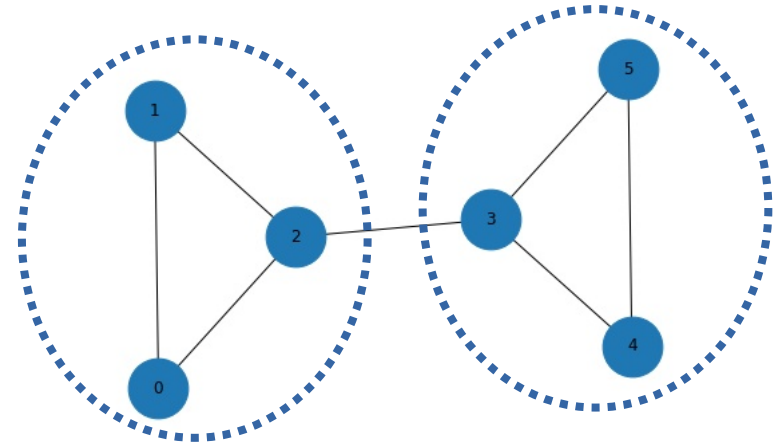
- There are L_C internal links in C
- Total number of stubs in nodes in C is k_C
- Total number of stubs in the network is $2L$
- Probability of choosing two stubs in C : $(k_C/2L)^2 = k_C^2/4L^2$
- The **expected number** of links joining two stubs in C is
 $L(k_C^2/4L^2) = k_C^2/4L$
- The **observed number** is L_C



Exercise

- What is the modularity of the partition $\{0, 1, 2\}, \{3, 4, 5\}$?
- What is the modularity of the partition $\{0, 1, 2, 3\}, \{4, 5\}$?

$$Q = \frac{1}{L} \sum_C \left(L_C - \frac{k_C^2}{4L} \right)$$



Summary

Things to remember

- Strong and weak community
- The concept of “cut” in graph bisection
- Girvan-Newman’s algorithm
- Modularity

Practice on your own

- Check the modularity computations in the example on the slide marked ★ : (a) optimal partitioning into two communities, (b) suboptimal partitioning into two communities, (c) all the nodes in a single community, (d) one community per node
- You can check your answers with [networkx.algorithms.community.modularity](https://networkx.org/documentation/stable/reference/algorithms/community/modularity.html)