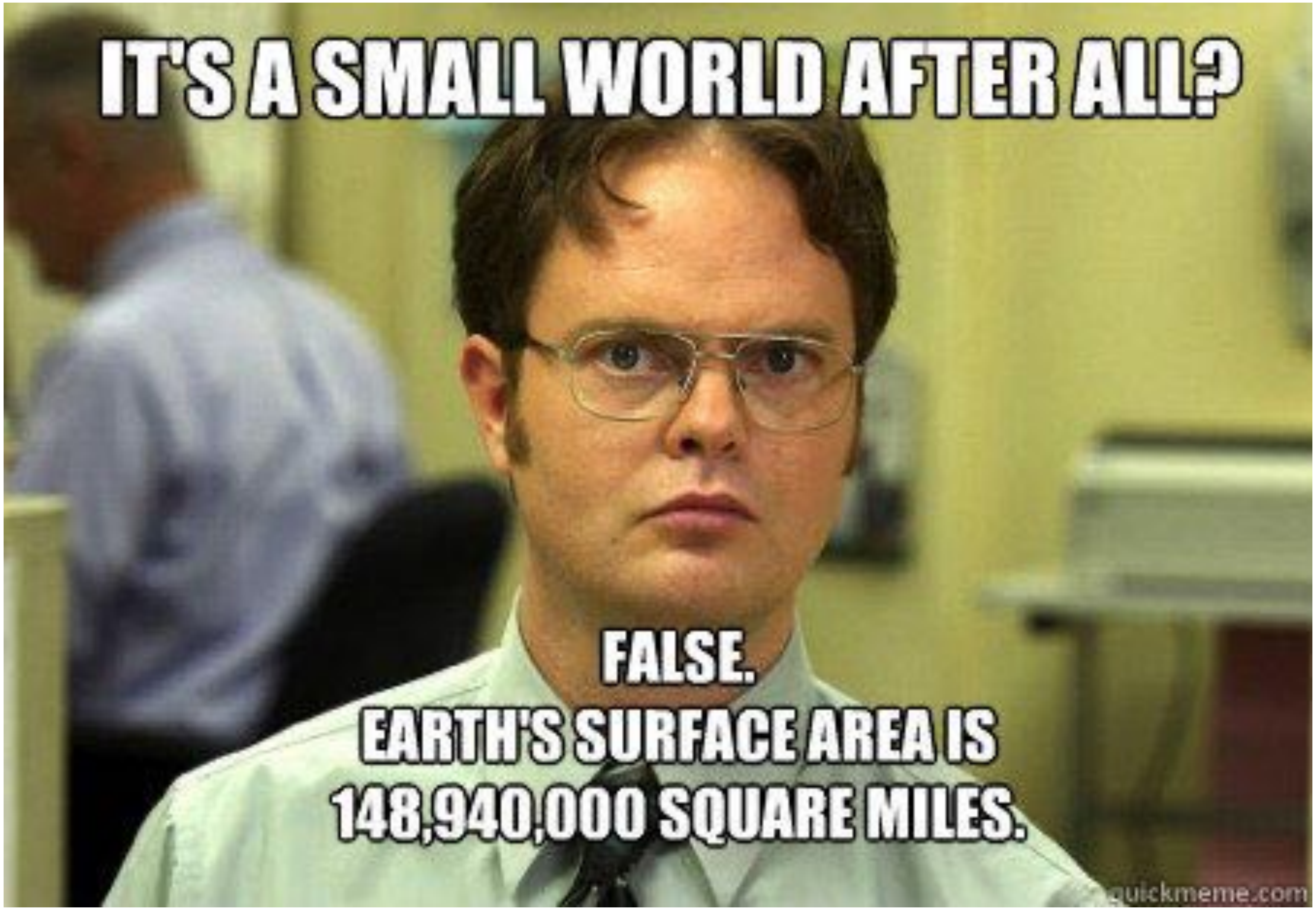
SMALL WORLDS



Valerio Restocchi / DBBA / lecture 03

LEARNING OUTCOMES

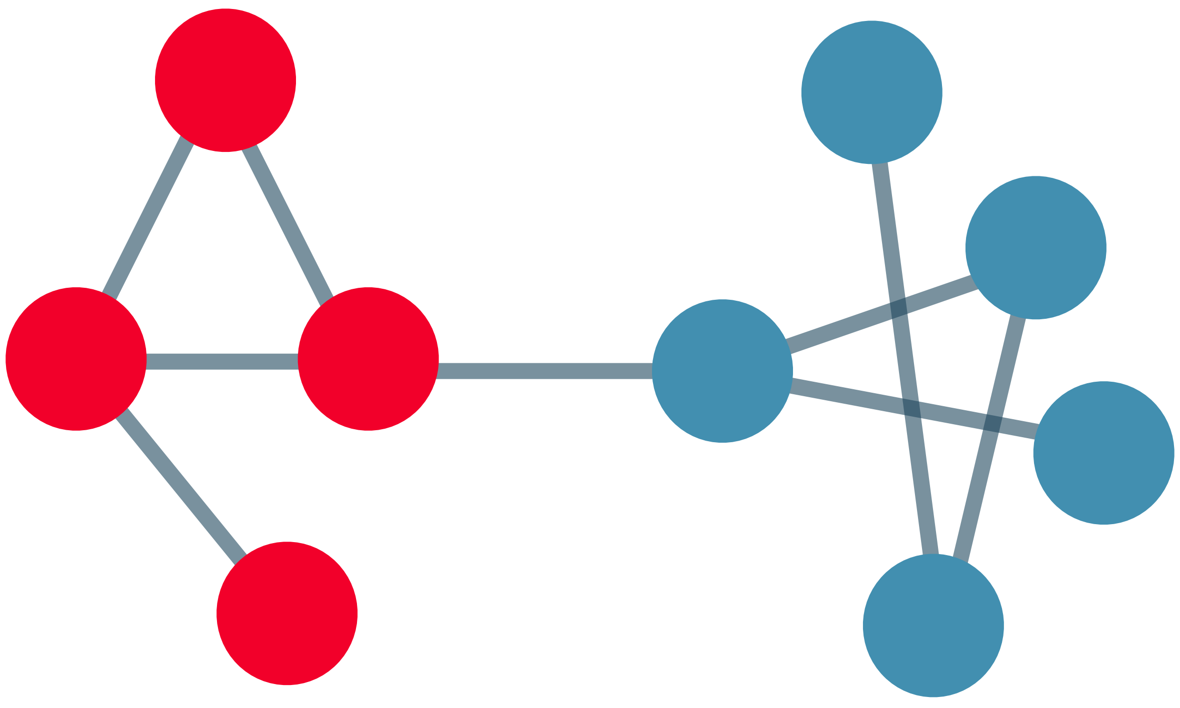
Learn about **similarity, paths, and triangles**

See how these concepts apply to **real-world networks**

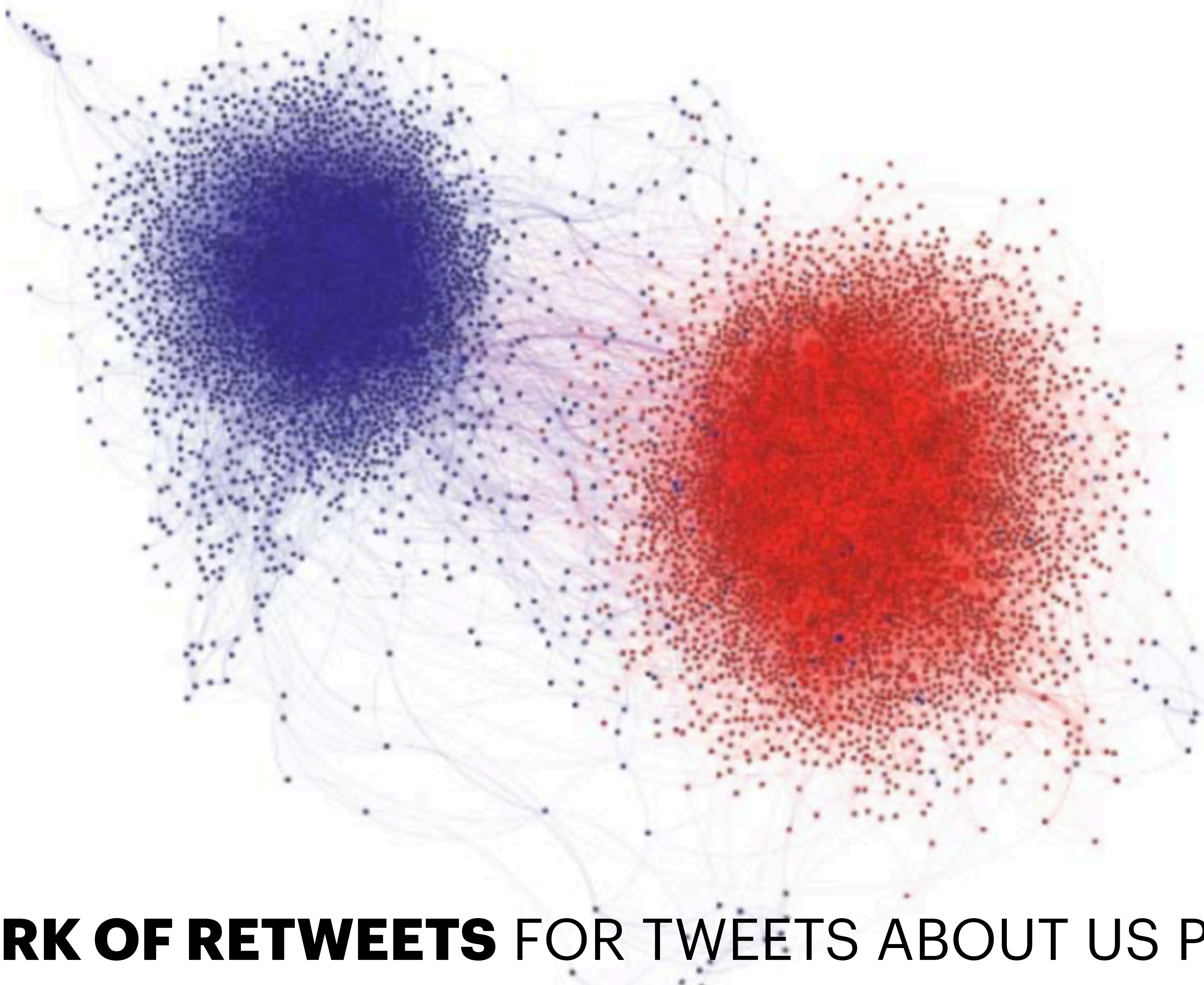
Learn **fun-facts** about networks to **impress your friends**

ASSORTATIVITY

ASSORTATIVITY



ASSORTATIVITY



NETWORK OF RETWEETS FOR TWEETS ABOUT US POLITICS
REDS ARE CONSERVATIVES AND BLUES ARE DEMOCRATS

ASSORTATIVITY

HOMOPHILY is what causes assortativity: WE TEND TO BECOME FRIENDS WITH PEOPLE WHO ARE LIKE US

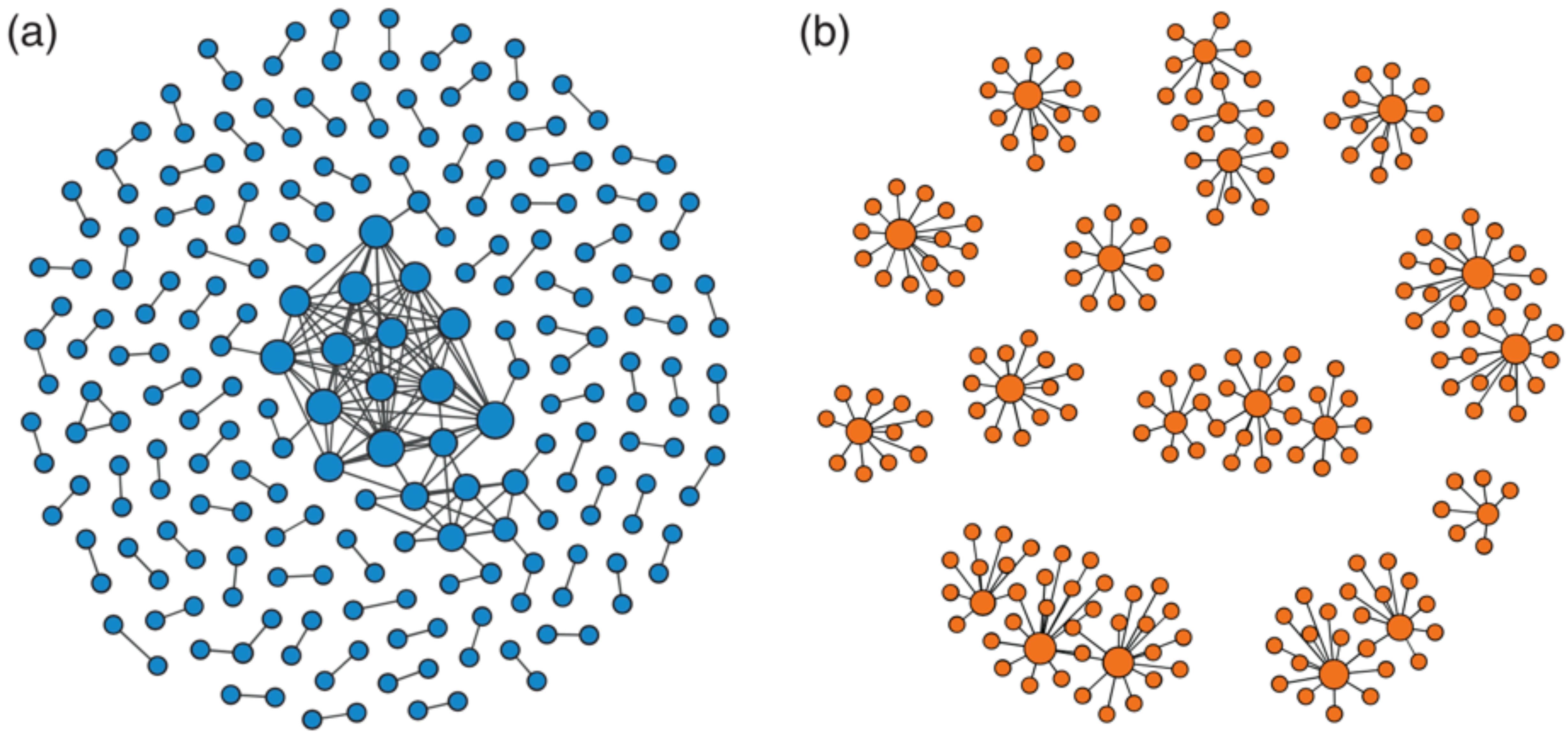
This, however, tends to cause **ECHO-CHAMBERS** and segregation

ASSORTATIVITY

Not only social networks: in some networks, nodes with **high** (low) **degree connect to** other nodes with **high** (low) **degree.**

This effect takes the name of **degree correlation**

ASSORTATIVITY



Network degree assortativity illustrated by (a) an assortative network and (b) a disassortative network.

ASSORTATIVITY

Some networks are **not** assortative. For example:

World wide web

Ecological networks

Biological networks

ASSORTATIVITY

Two ways to **compute** assortativity:

correlation between degrees of pairs of nodes (usually Pearson)

average degree of neighbours

ASSORTATIVITY

K-nearest neighbours

$$k_{nn}(i) = \frac{1}{k_i} \sum_j a_{ij} k_j$$

ASSORTATIVITY

K-nearest neighbours

Knn of node i

$$k_{nn}(i) = \frac{1}{k_i} \sum_j a_{ij} k_j$$

ASSORTATIVITY

K-nearest neighbours

$$k_{nn}(i) = \frac{1}{k_i} \sum_j a_{ij} k_j$$

Number of neighbours of i → k_i

ASSORTATIVITY

K-nearest neighbours

$$k_{nn}(i) = \frac{1}{k_i} \sum_j a_{ij} k_j$$

1 if i and j are neighbours,
0 otherwise

ASSORTATIVITY

K-nearest neighbours

$$k_{nn}(i) = \frac{1}{k_i} \sum_j a_{ij} k_j$$

Degree of node j

ASSORTATIVITY

K-nearest neighbours

$$k_{nn}(i) = \frac{1}{k_i} \sum_j a_{ij} k_j$$

Number of neighbours of i → k_i

1 if i and j are neighbours, 0 otherwise → a_{ij}

Degree of node j → k_j

ASSORTATIVITY

K-nearest neighbours function

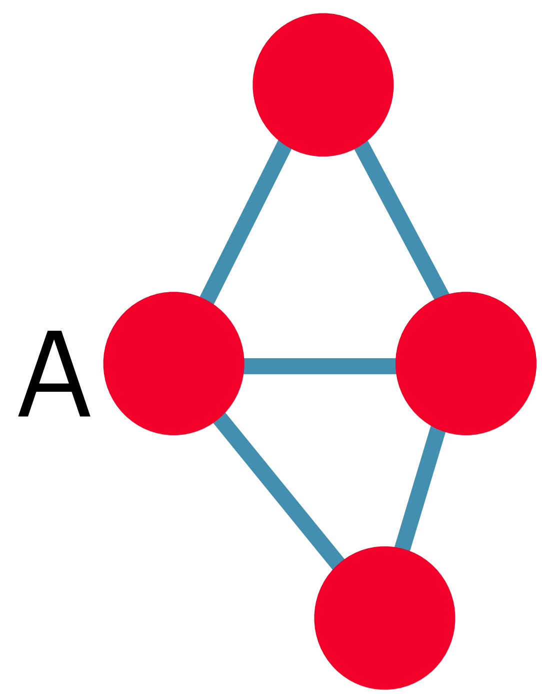
$$\langle k_{nn}(k) \rangle$$

The average degree of the neighbours of nodes of degree k

EXERCISE:

GIVEN THE FOLLOWING NETWORK COMPUTE

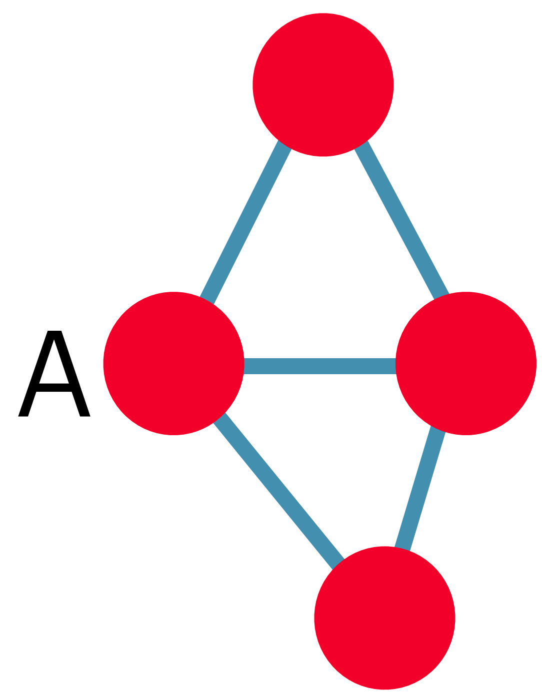
$$k_{nn}(A) \qquad \langle k_{nn}(2) \rangle$$



EXERCISE:

GIVEN THE FOLLOWING NETWORK COMPUTE

$$k_{nn}(A) \qquad \langle k_{nn}(2) \rangle$$

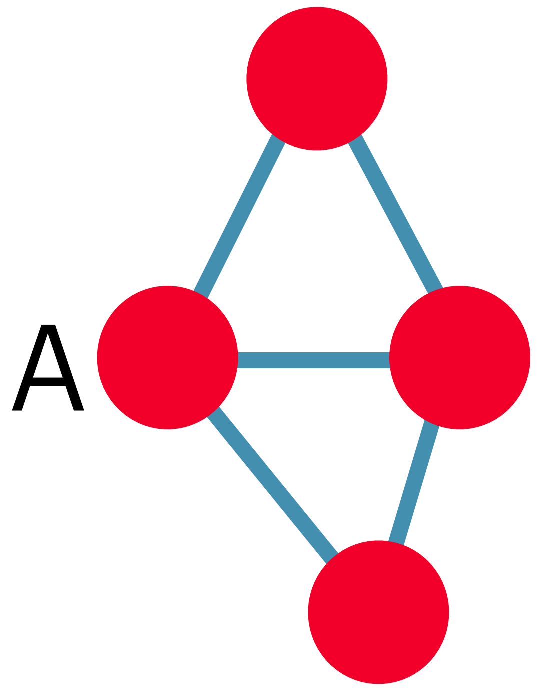


$$k_{nn}(A) = \frac{1}{k_A} \sum_j a_{ij} k_j$$

EXERCISE:

GIVEN THE FOLLOWING NETWORK COMPUTE

$$k_{nn}(A) \qquad \langle k_{nn}(2) \rangle$$

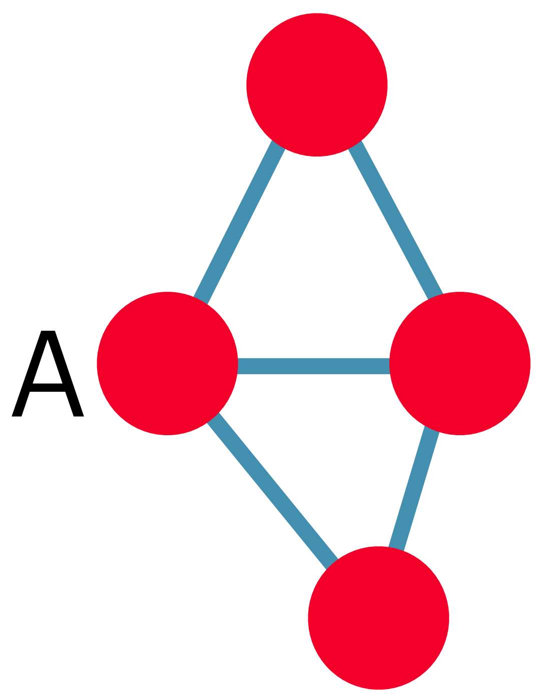


$$k_{nn}(A) = \frac{1}{3} \sum_j a_{ij} k_j$$

EXERCISE:

GIVEN THE FOLLOWING NETWORK COMPUTE

$$k_{nn}(A) \qquad \langle k_{nn}(2) \rangle$$

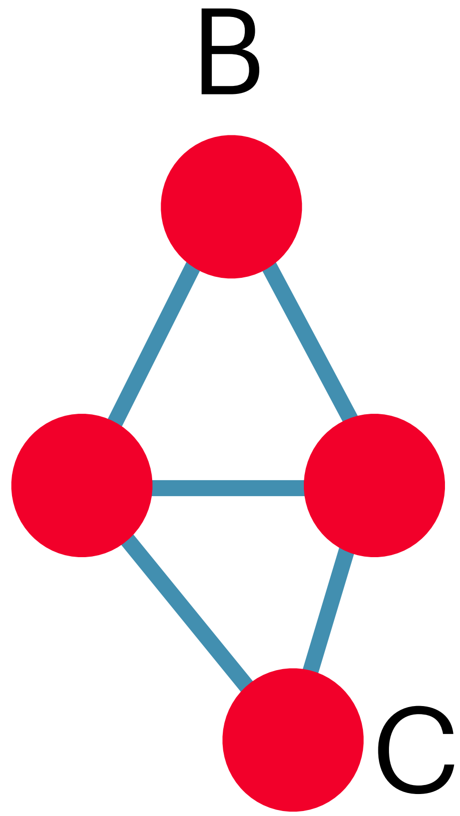


$$k_{nn}(A) = \frac{1}{3}(1 * 2 + 1 * 3 + 1 * 2) = \frac{7}{3}$$

EXERCISE:

GIVEN THE FOLLOWING NETWORK COMPUTE

$$k_{nn}(A) \qquad \langle k_{nn}(2) \rangle$$

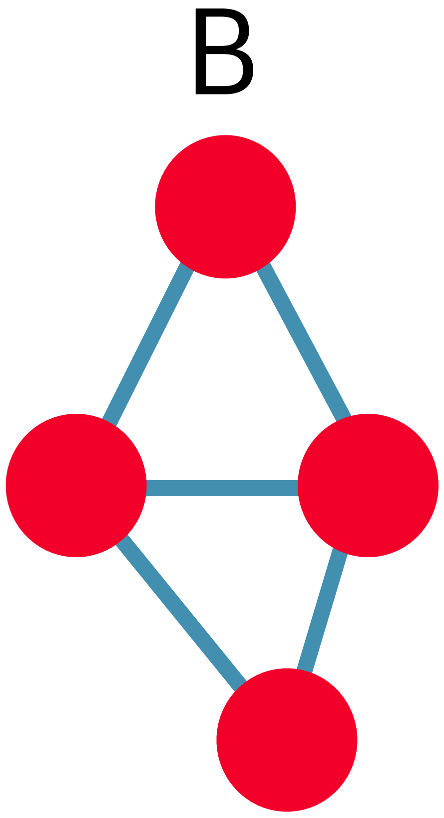


Which nodes have degree of 2?

EXERCISE:

GIVEN THE FOLLOWING NETWORK COMPUTE

$k_{nn}(A)$ $\langle k_{nn}(2) \rangle$

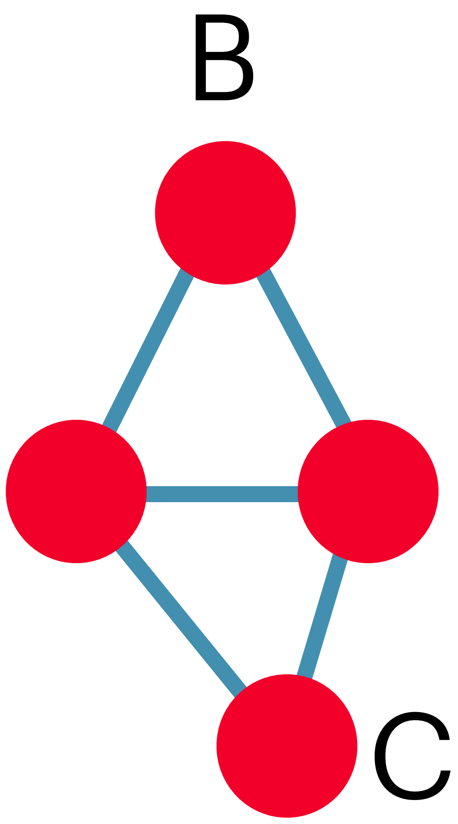


$k_{nn}(B) = ?$

EXERCISE:

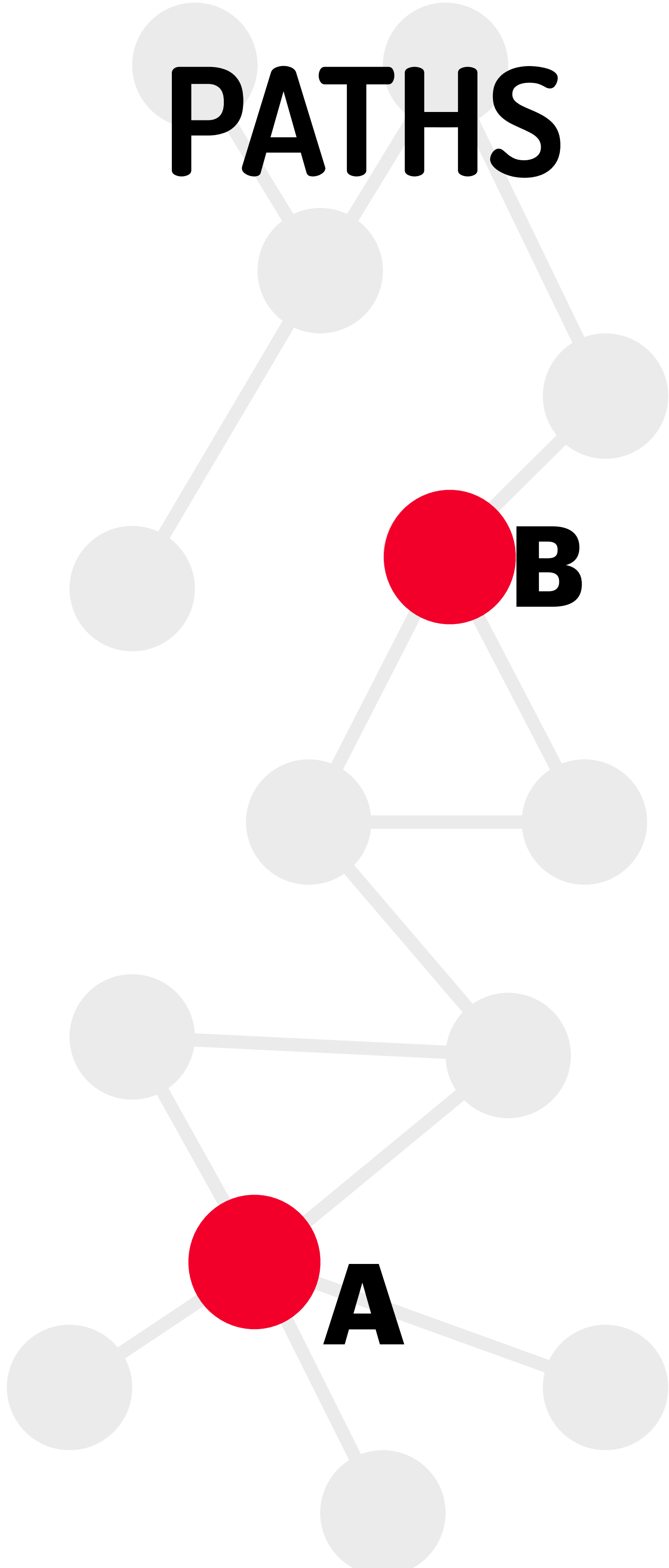
GIVEN THE FOLLOWING NETWORK COMPUTE

$$k_{nn}(A) \qquad \langle k_{nn}(2) \rangle$$

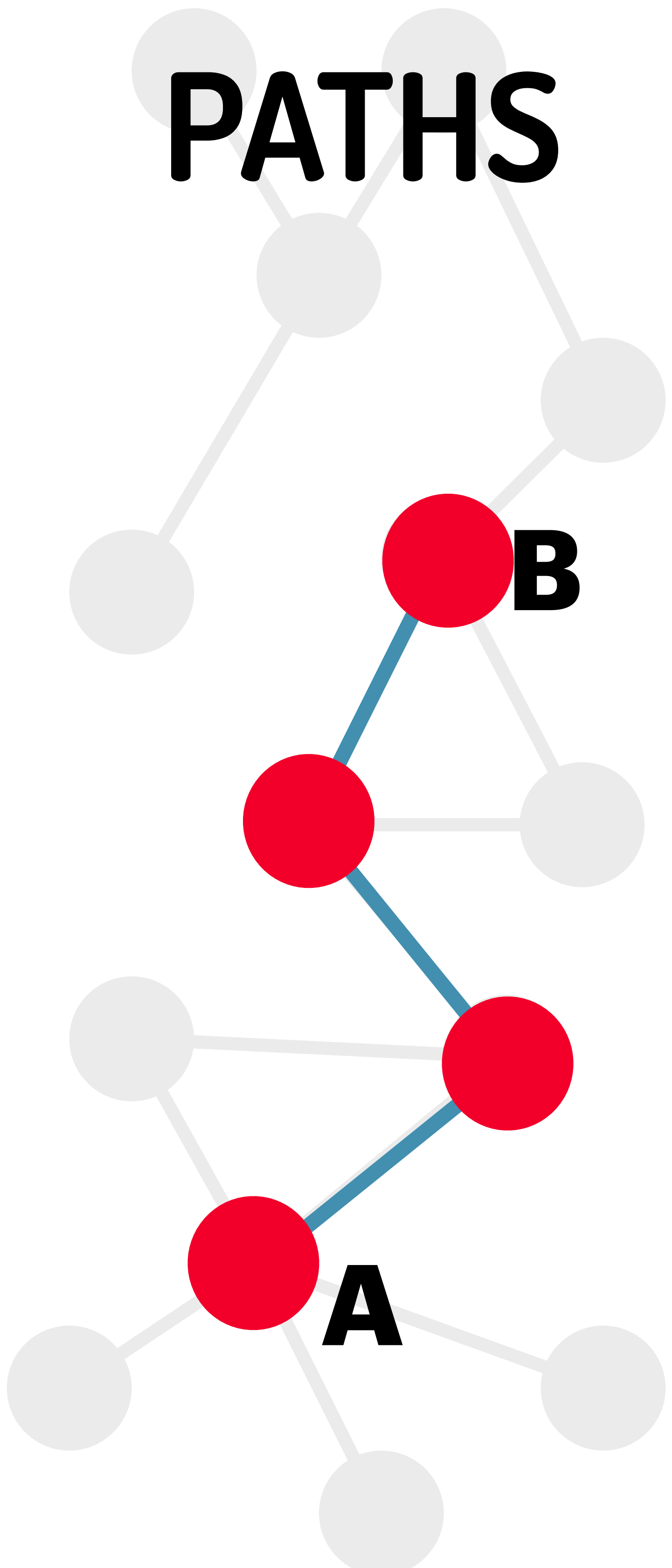


Pro tip: network is symmetric for these nodes

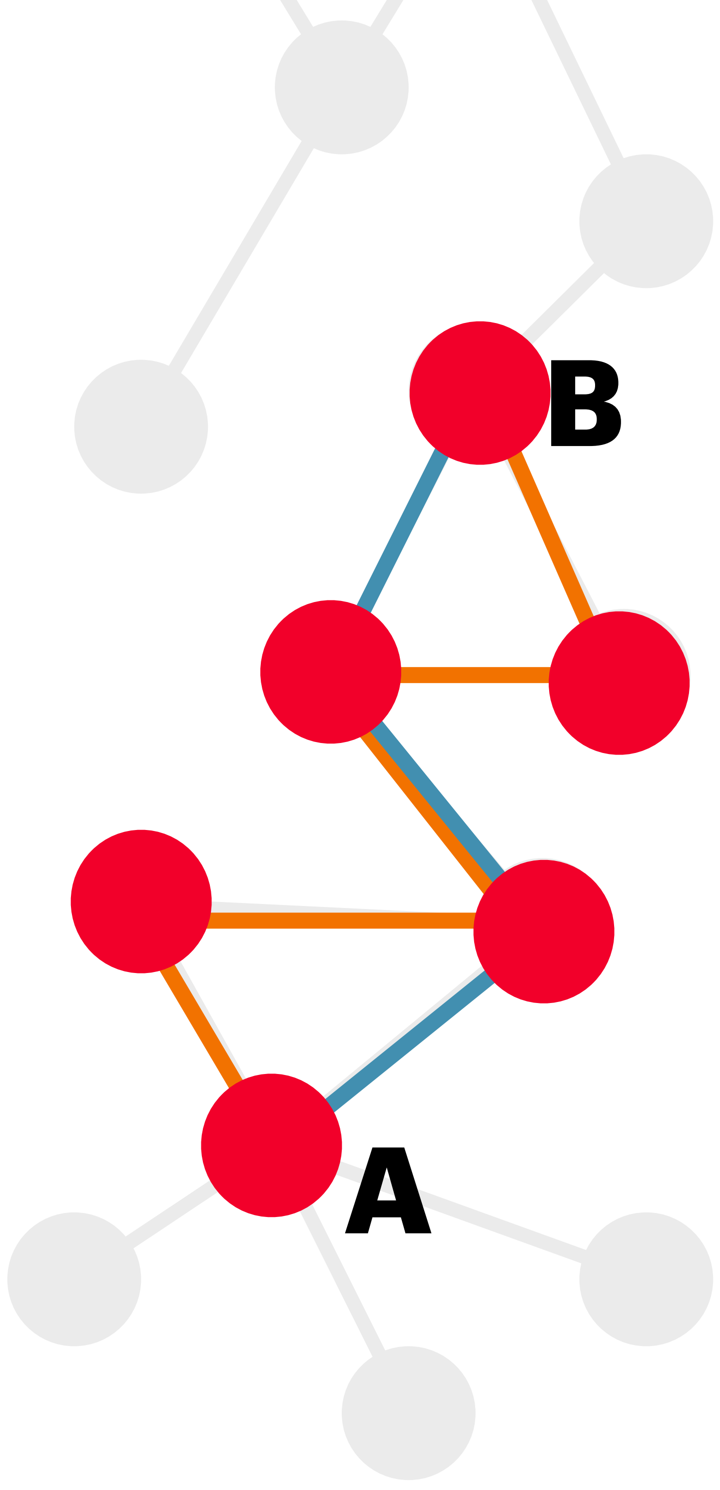
PATHS



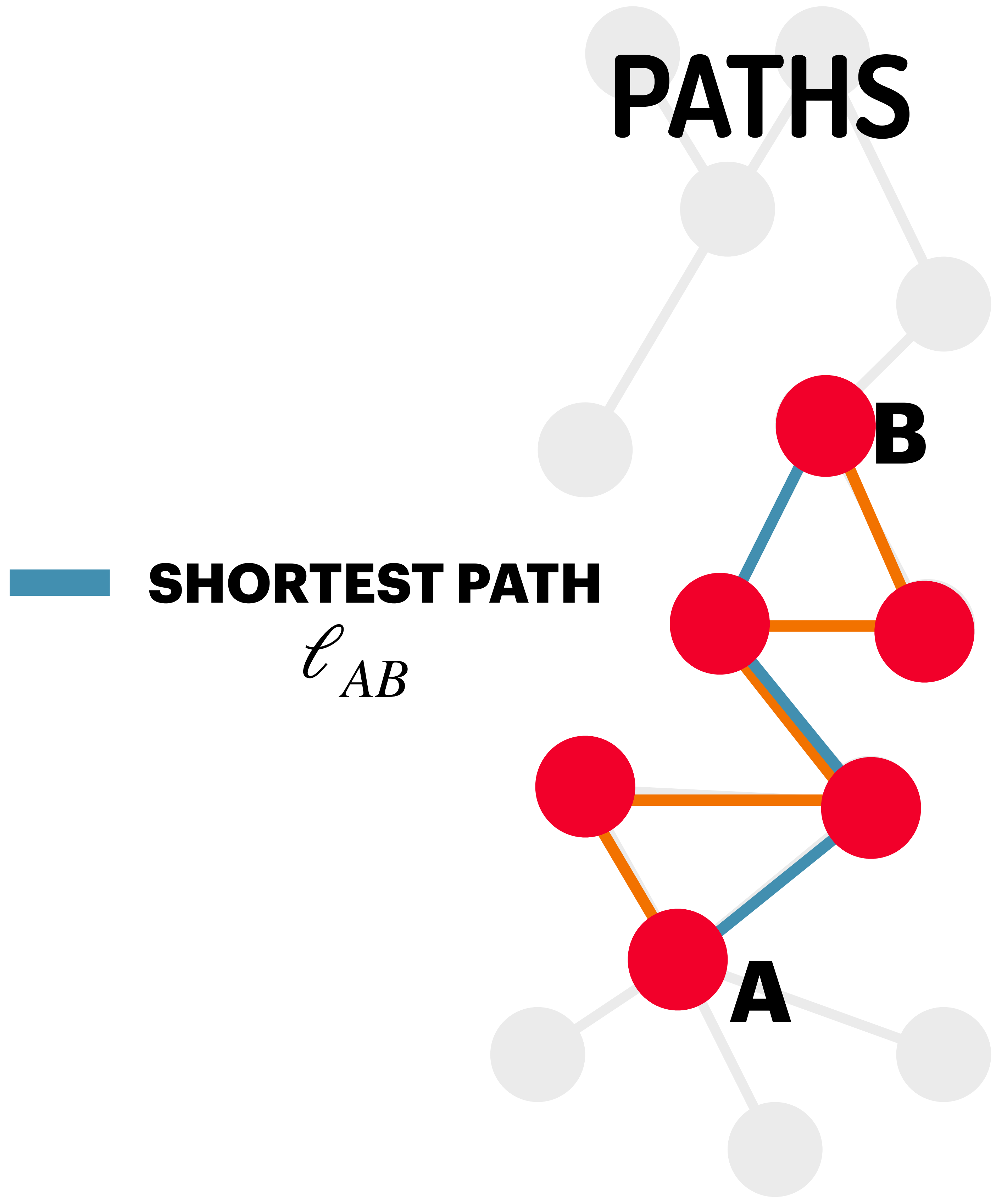
PATHS



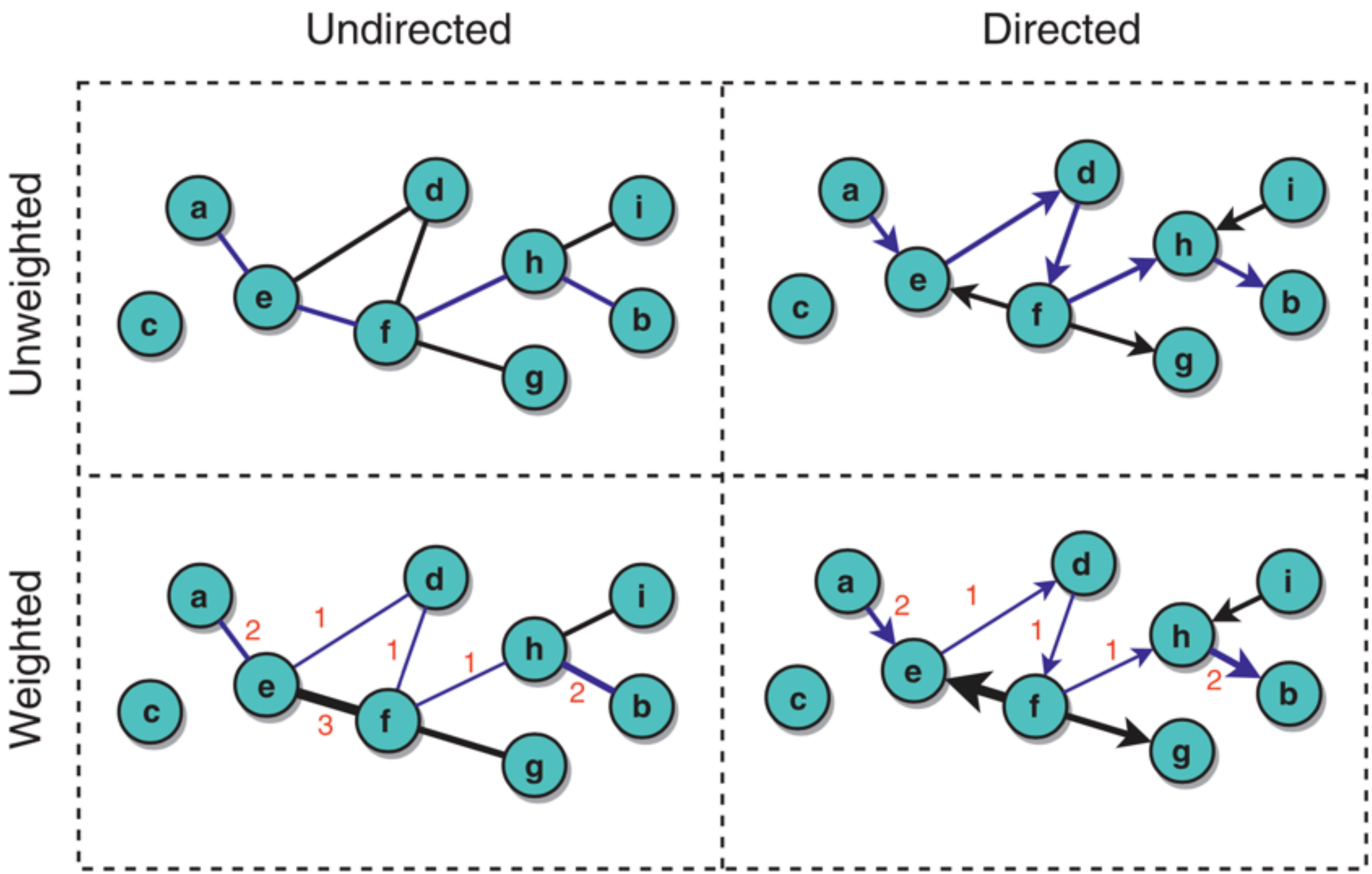
PATHS



PATHS

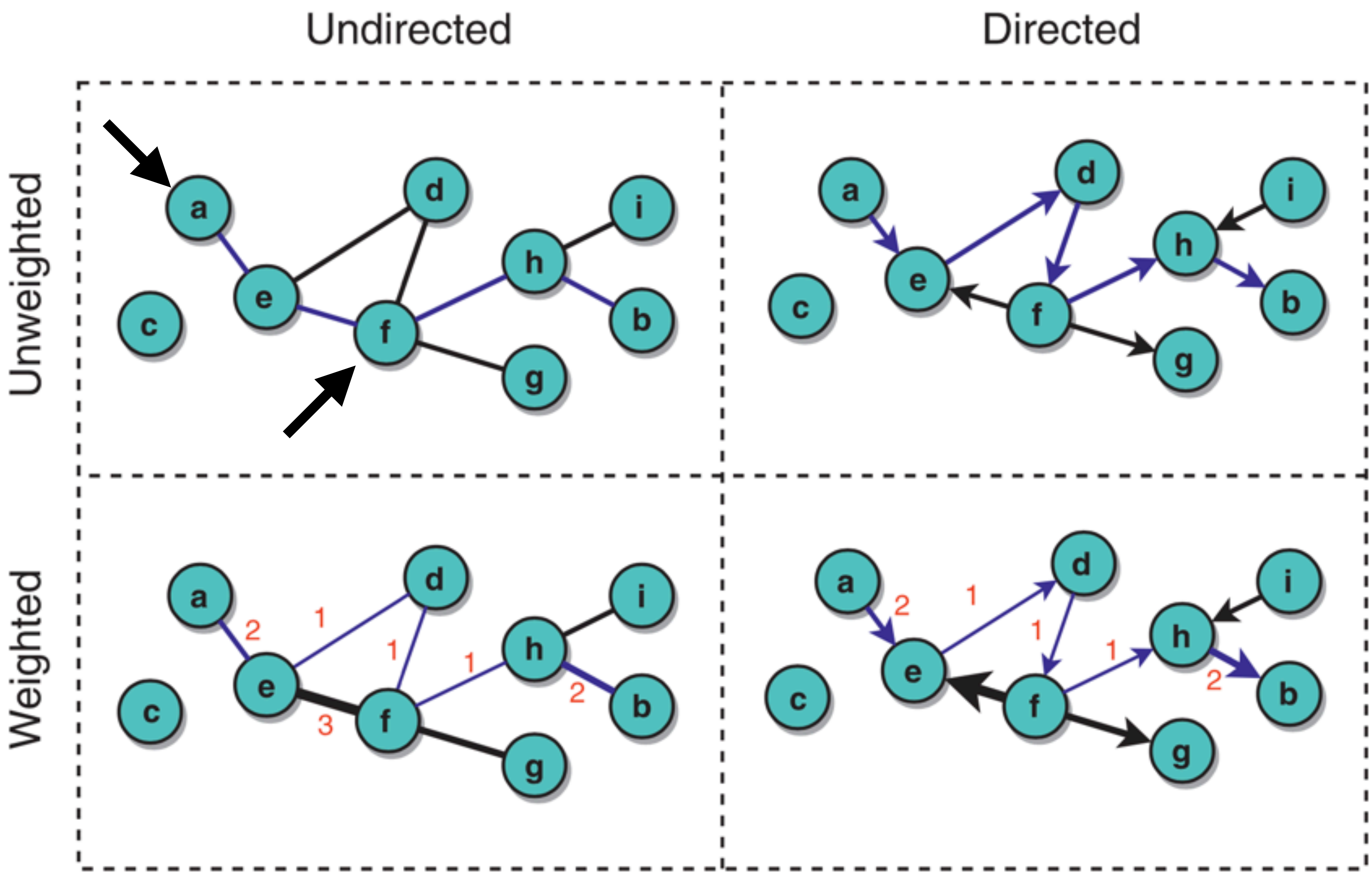


PATHS



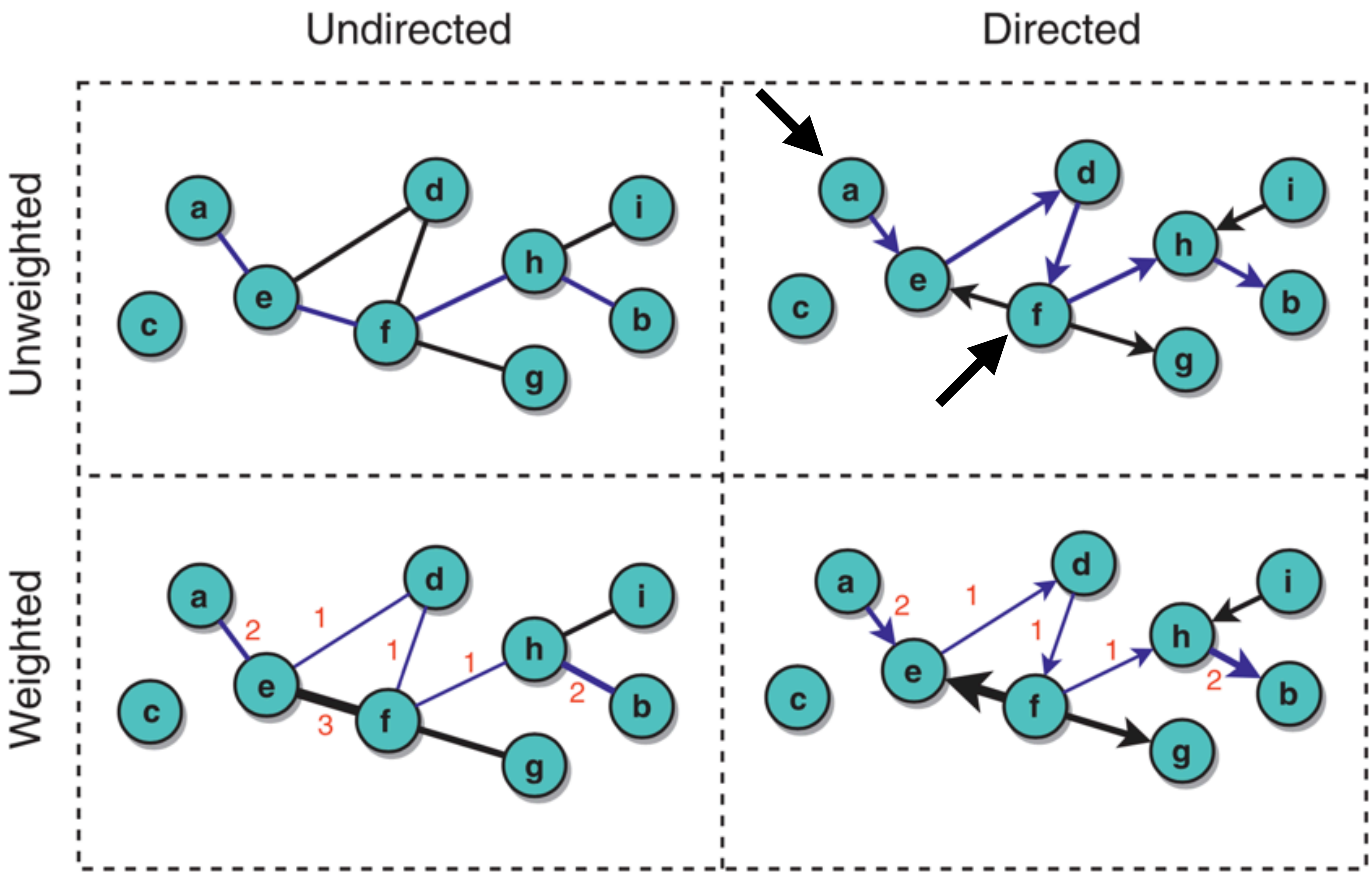
Undirected
Directed
Unweighted
Weighted
a
b
c
d
e
f
g
h
i
2
1
1
1
2
3
1
1
1
2
2

PATHS

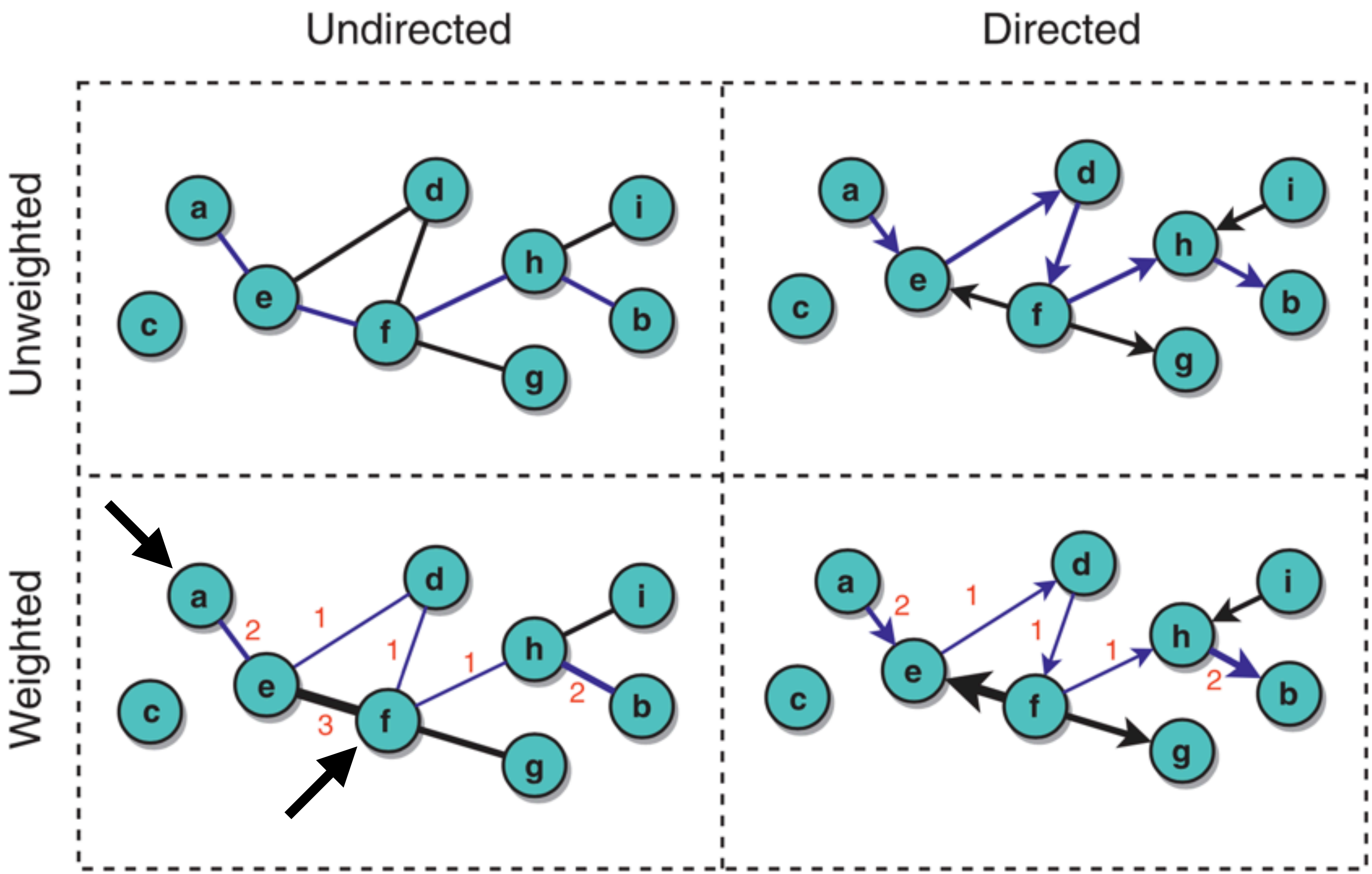


Undirected
Directed
Unweighted
Weighted
a
b
c
d
e
f
g
h
i
2
1
1
1
2
3

PATHS


Undirected
Directed
Unweighted
Weighted
a
b
c
d
e
f
g
h
i
2
1
1
1
3
2
2
1
1
1
2

PATHS


Undirected
Directed
Unweighted
Weighted
a
b
c
d
e
f
g
h
i
2
1
1
1
2
3
2
1
1
1
2

PATHS

AVERAGE PATH - UNDIRECTED NETWORK

$$\langle \ell \rangle = \frac{\sum_{i,j} \ell_{ij}}{\binom{N}{2}} = 2\frac{\sum_{i,j} \ell_{ij}}{N(N-1)}$$

PATHS

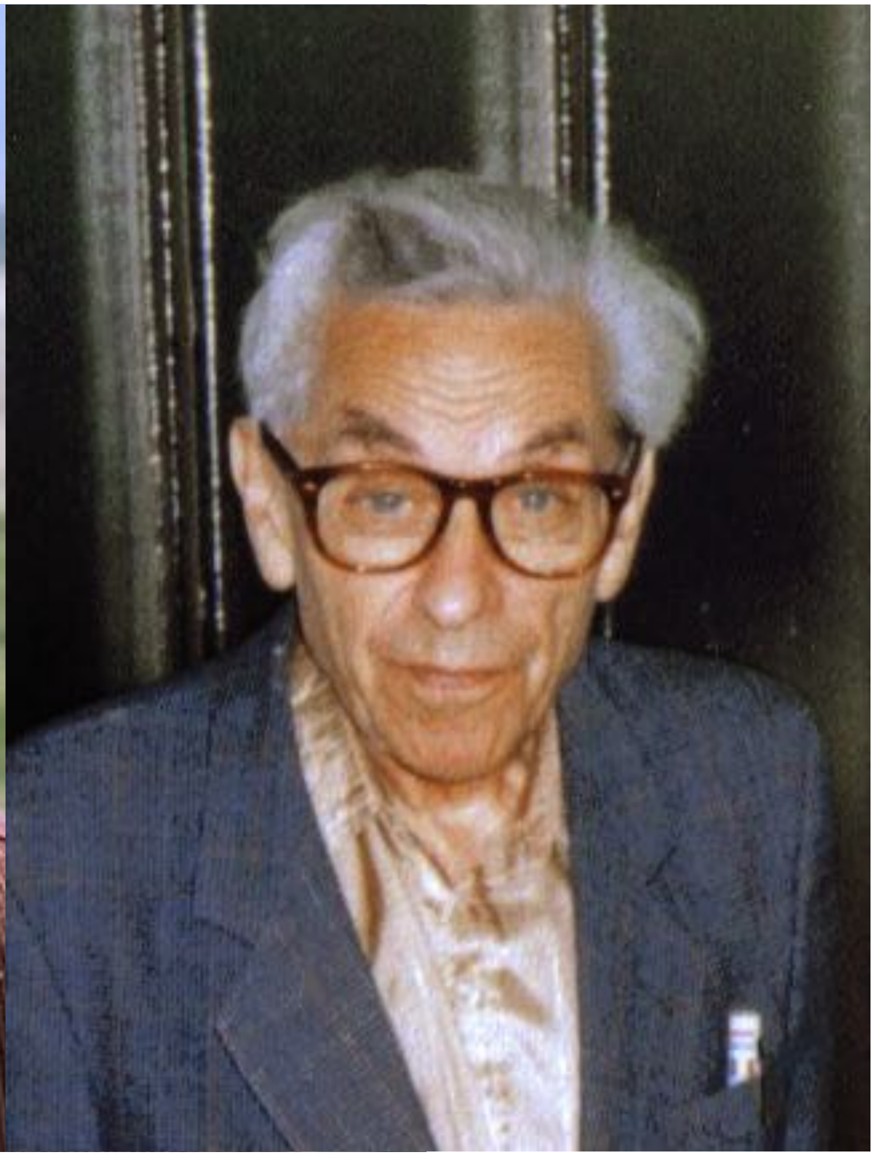
AVERAGE PATH - UNDIRECTED NETWORK

$$\langle \ell \rangle = \frac{\sum_{i,j} \ell_{ij}}{\binom{N}{2}} = 2\frac{\sum_{i,j} \ell_{ij}}{N(N-1)}$$

DIAMETER - UNDIRECTED NETWORK

$$\ell_{max} = max_{i,j} \ell_{ij}$$

SIX DEGREES OF SEPARATION



SIX DEGREES OF SEPARATION

An average path length is said to be **short** if:

$$\langle \ell \rangle \approx \log(N)$$

CONNECTEDNESS

A NETWORK IS **CONNECTED** IF THERE IS AT LEAST A **PATH** BETWEEN **ANY PAIR OF NODES**

CONNECTEDNESS

A NETWORK IS **CONNECTED** IF THERE IS AT LEAST A **PATH** BETWEEN **ANY PAIR OF NODES**

A **COMPONENT** IS A **CONNECTED SUB-GRAPH**

CONNECTEDNESS

A NETWORK IS **CONNECTED** IF THERE IS AT LEAST A **PATH** BETWEEN **ANY PAIR OF NODES**

A **COMPONENT** IS A **CONNECTED SUB-GRAPH**

THE **LARGEST** COMPONENT IS CALLED **GIANT COMPONENT**

CONNECTEDNESS

FOR DIRECTED NETWORKS COMPONENTS CAN BE:

WEAKLY CONNECTED:
IF CONNECTED ONLY **DISREGARDING THE DIRECTION** OF LINKS

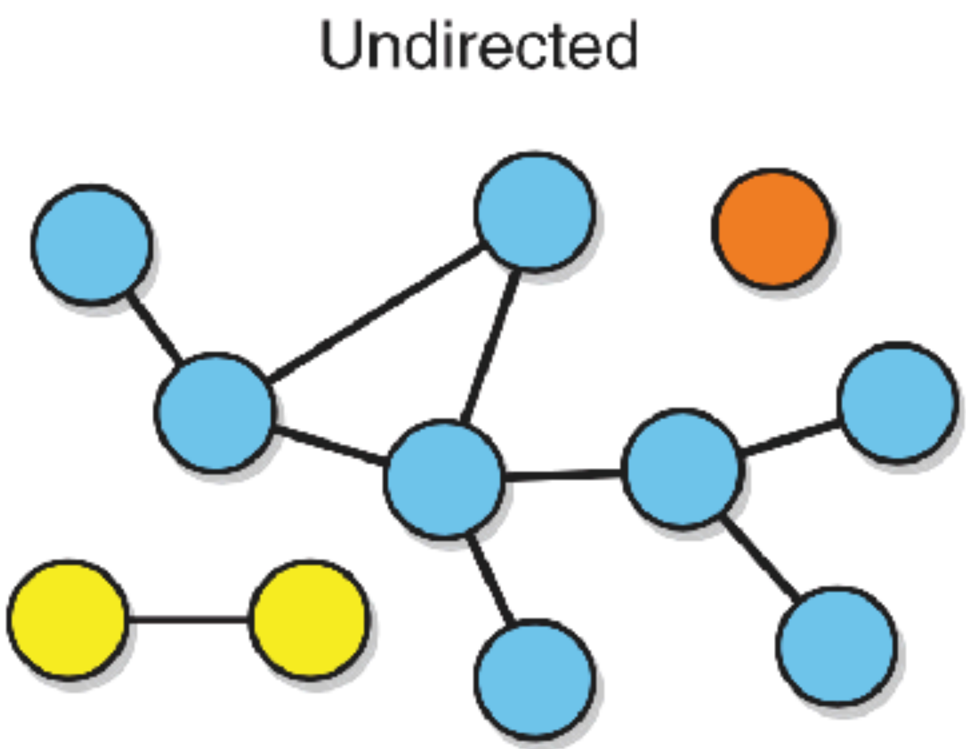
CONNECTEDNESS

FOR DIRECTED NETWORKS COMPONENTS CAN BE:

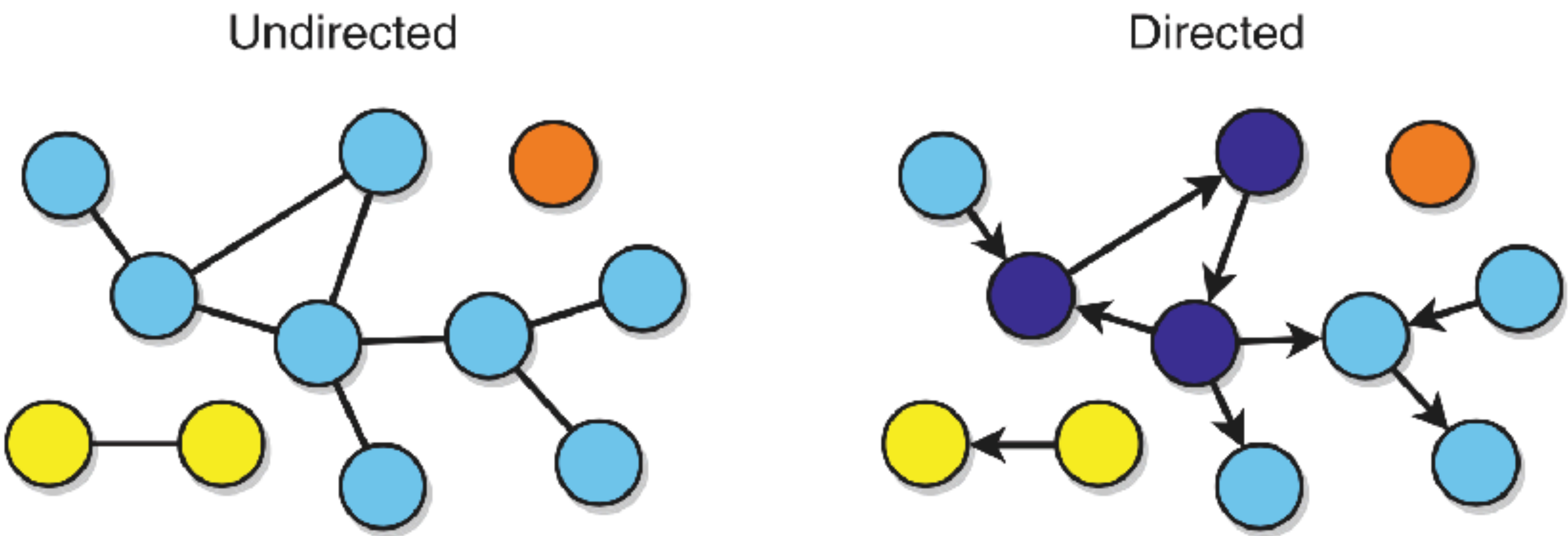
WEAKLY CONNECTED:
IF CONNECTED ONLY **DISREGARDING THE DIRECTION** OF LINKS

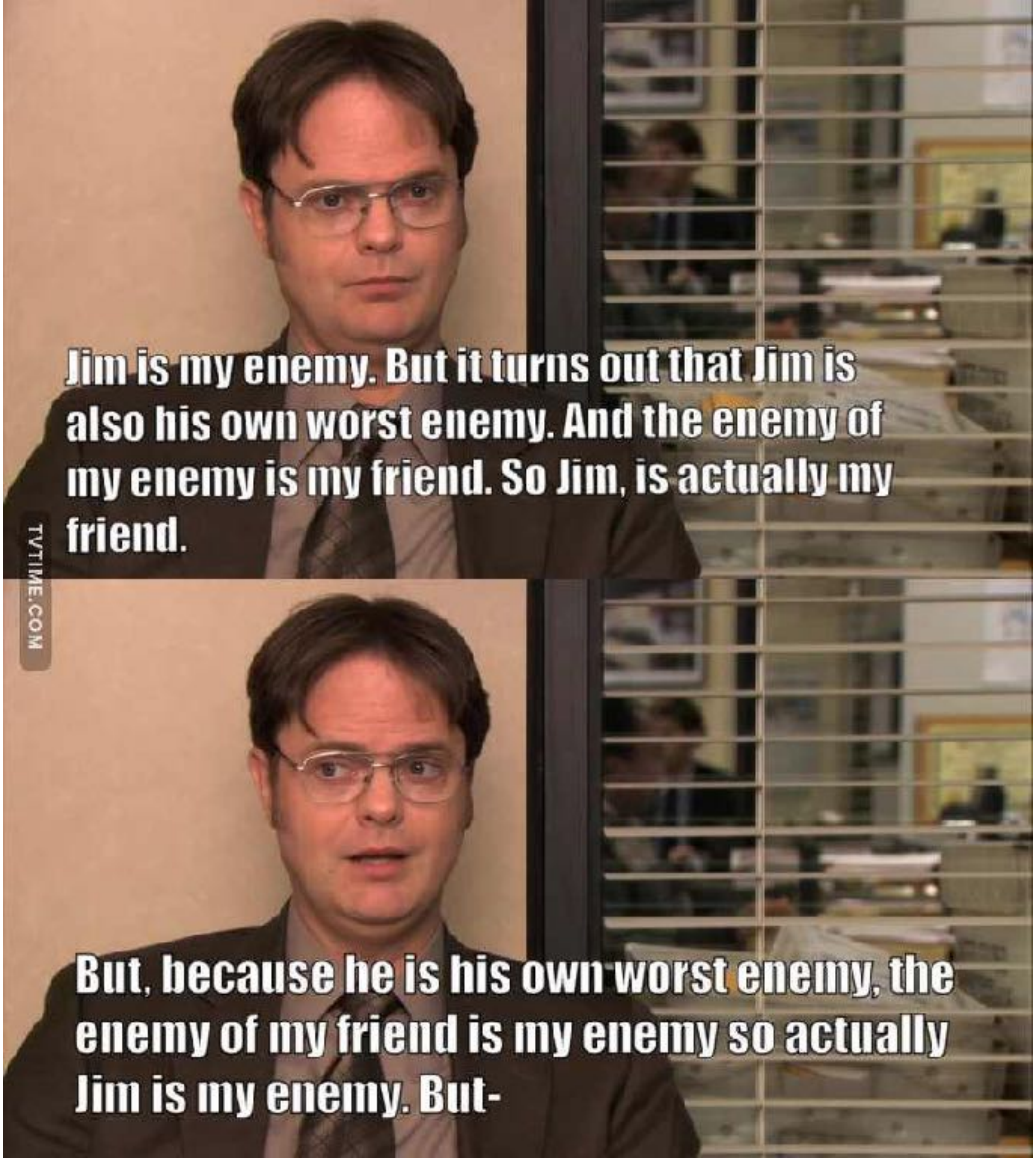
STRONGLY CONNECTED:
IF CONNECTED ALSO WHEN **CONSIDERING THE DIRECTION** OF LINKS

CONNECTEDNESS

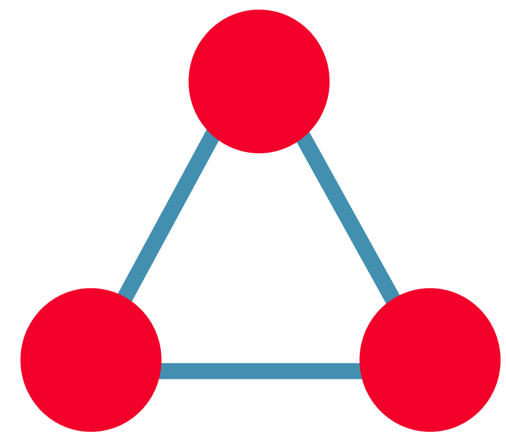


CONNECTEDNESS

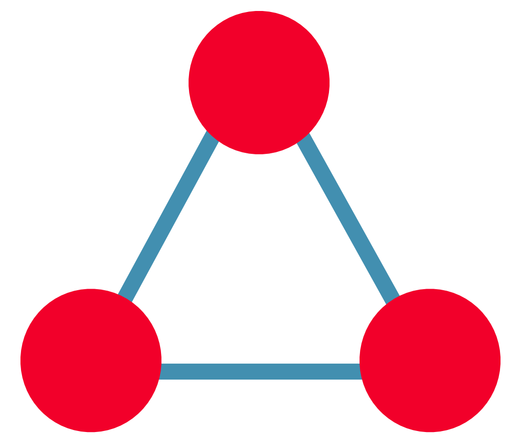



Jim is my enemy. But it turns out that Jim is also his own worst enemy. And the enemy of my enemy is my friend. So Jim, is actually my friend.
TVTIME.COM
But, because he is his own worst enemy, the enemy of my friend is my enemy so actually Jim is my enemy. But-

TRIANGLES AND FRIENDS

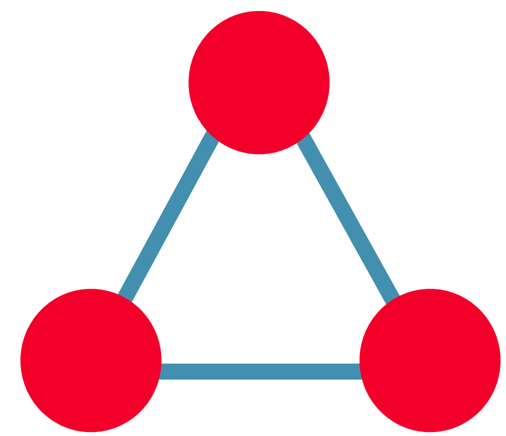


TRIANGLES AND FRIENDS



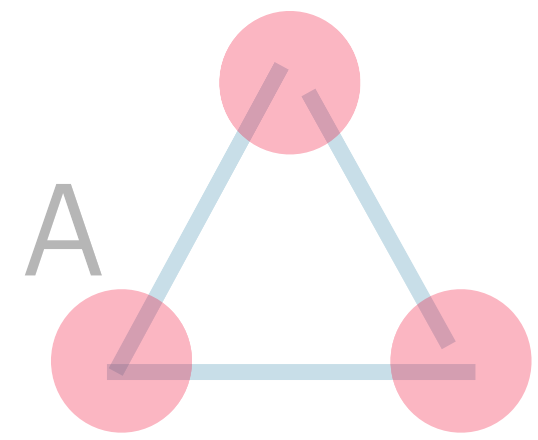
The **clustering coefficient** $C(i)$ of node i is the fraction of pairs of the neighbours of i that are **connected to each other**

TRIANGLES AND FRIENDS

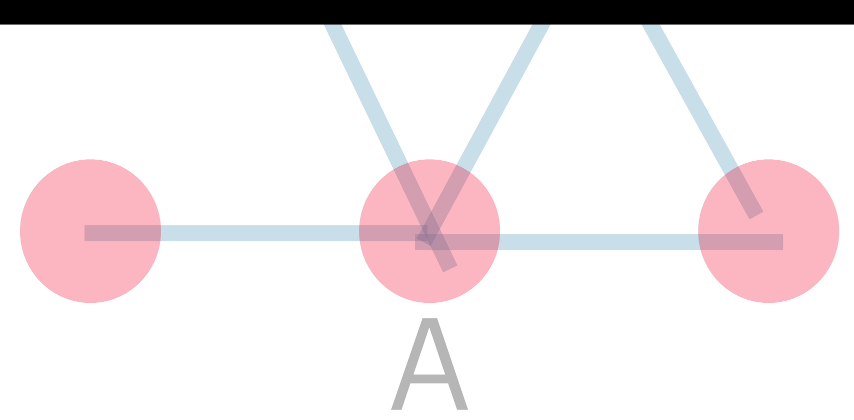


$$C(i) = \frac{\tau(i)}{\tau_{max}(i)} = \frac{2\tau(i)}{k_i(k_i - 1)}$$

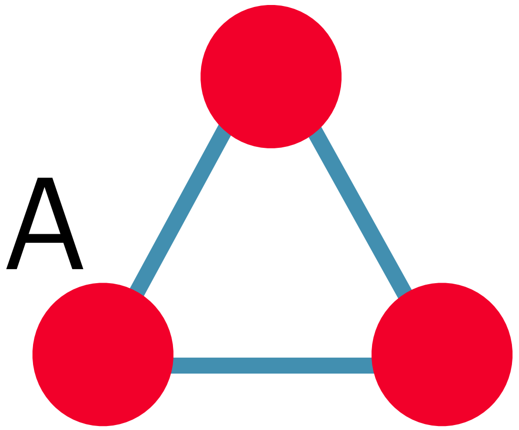
TRIANGLES AND FRIENDS



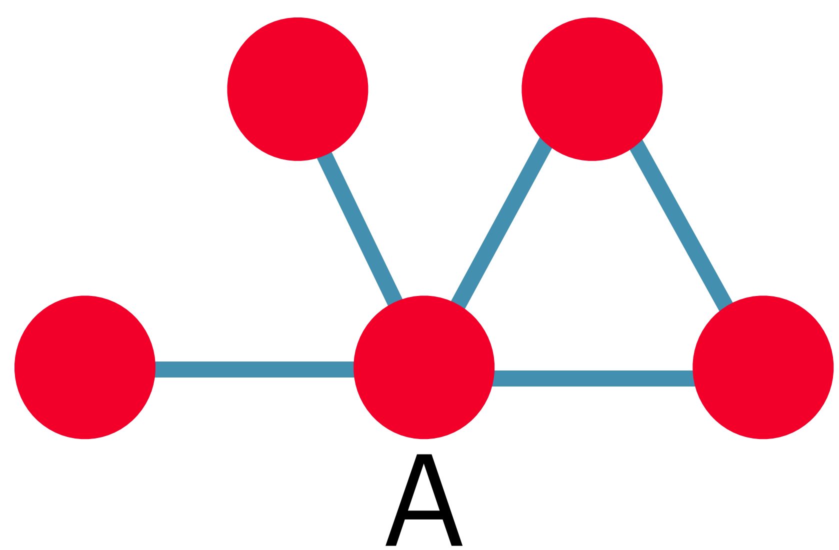
WHAT IS THE C(A) IN THE FOLLOWING NETWORKS?



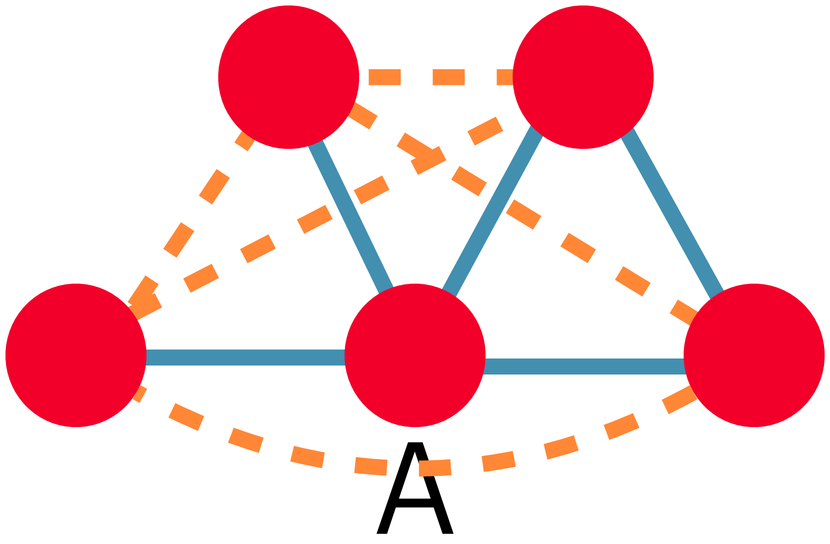
TRIANGLES AND FRIENDS



TRIANGLES AND FRIENDS

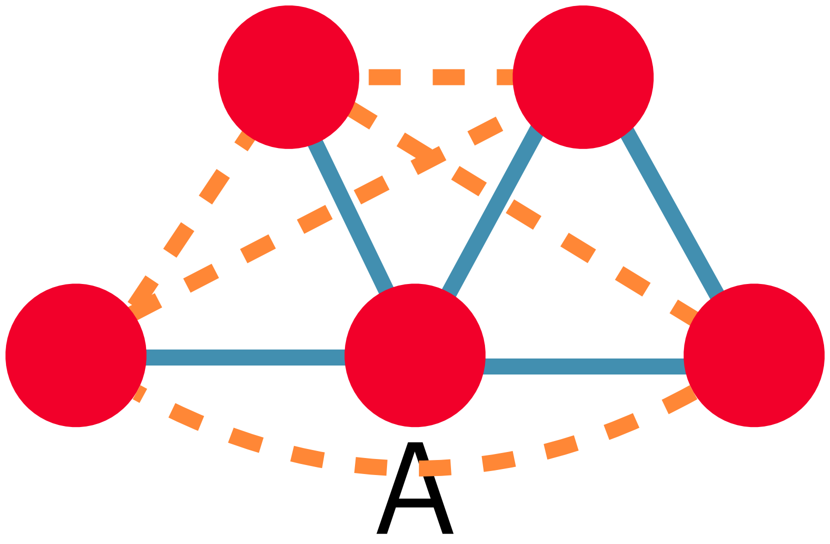


TRIANGLES AND FRIENDS

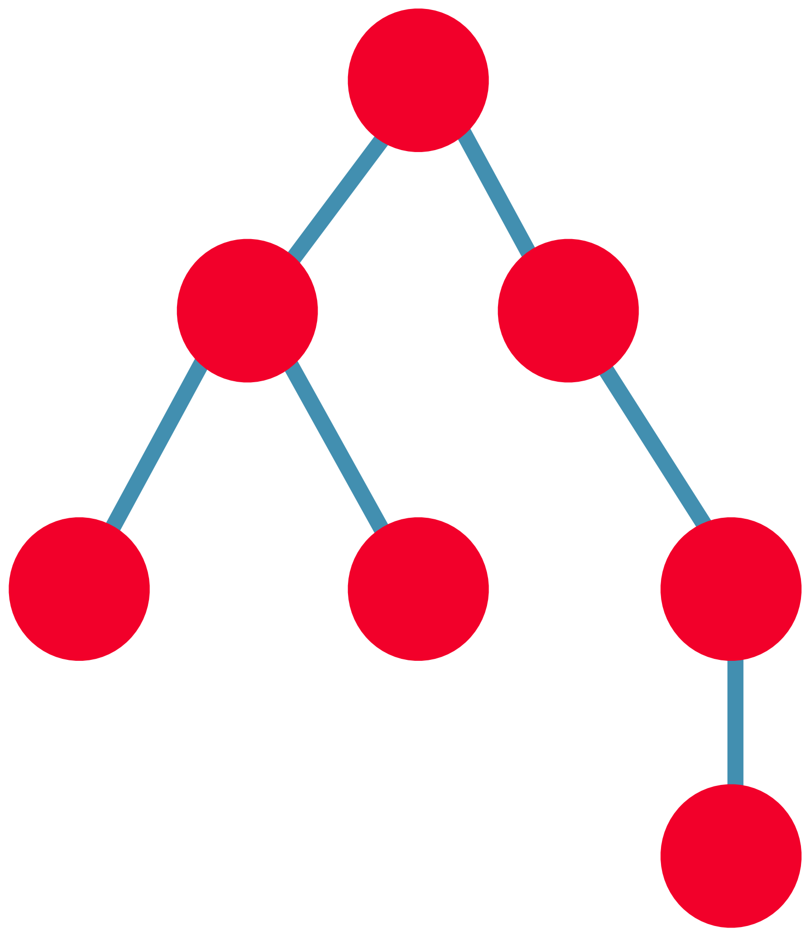


TRIANGLES AND FRIENDS

$$C(i) = \frac{\tau(i)}{\tau_{max}(i)} = \frac{2\tau(i)}{k_i(k_i - 1)}$$



TREES



TREES

ACYCLIC

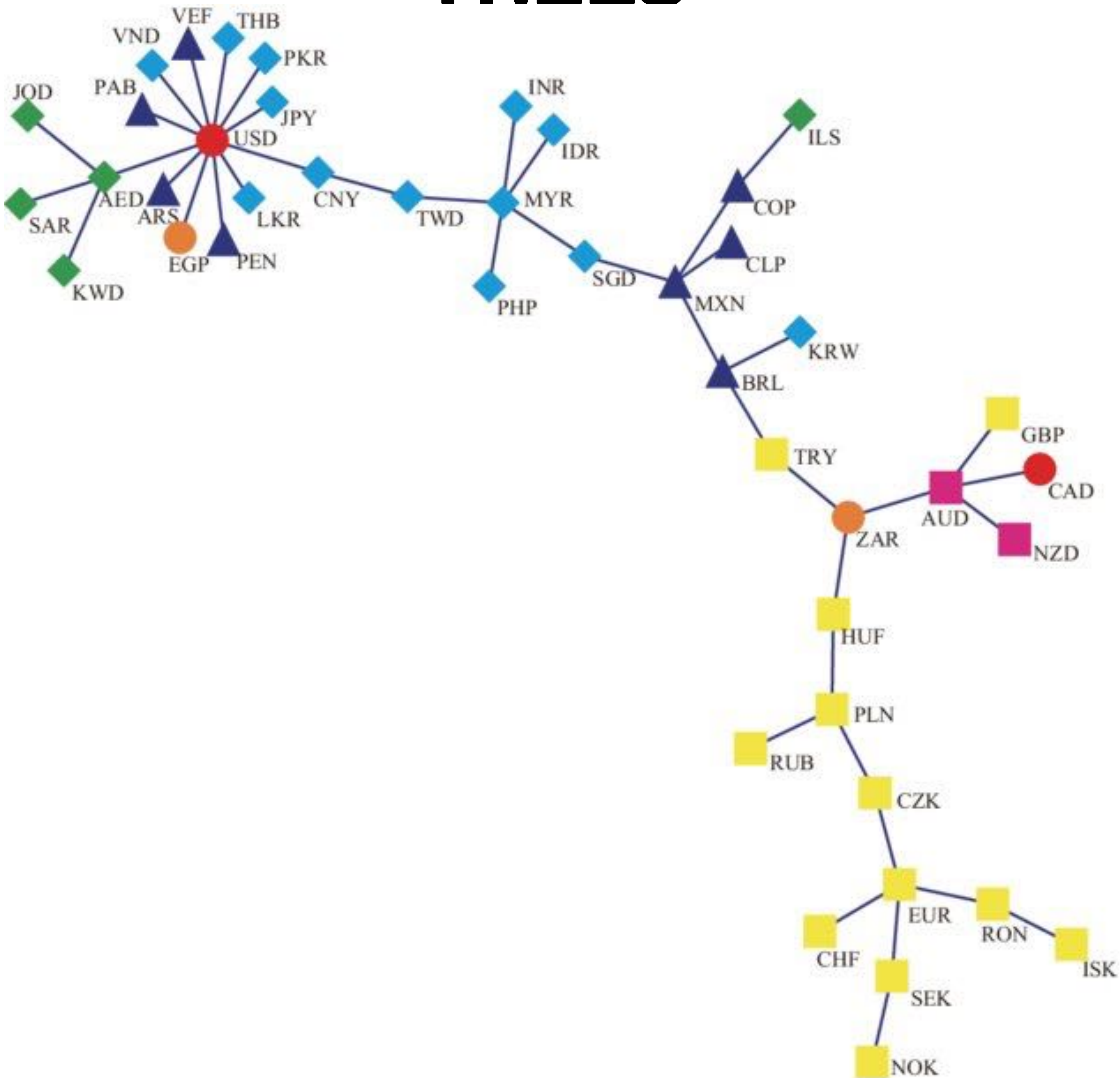
CONNECTED

(PLANAR)

TREES

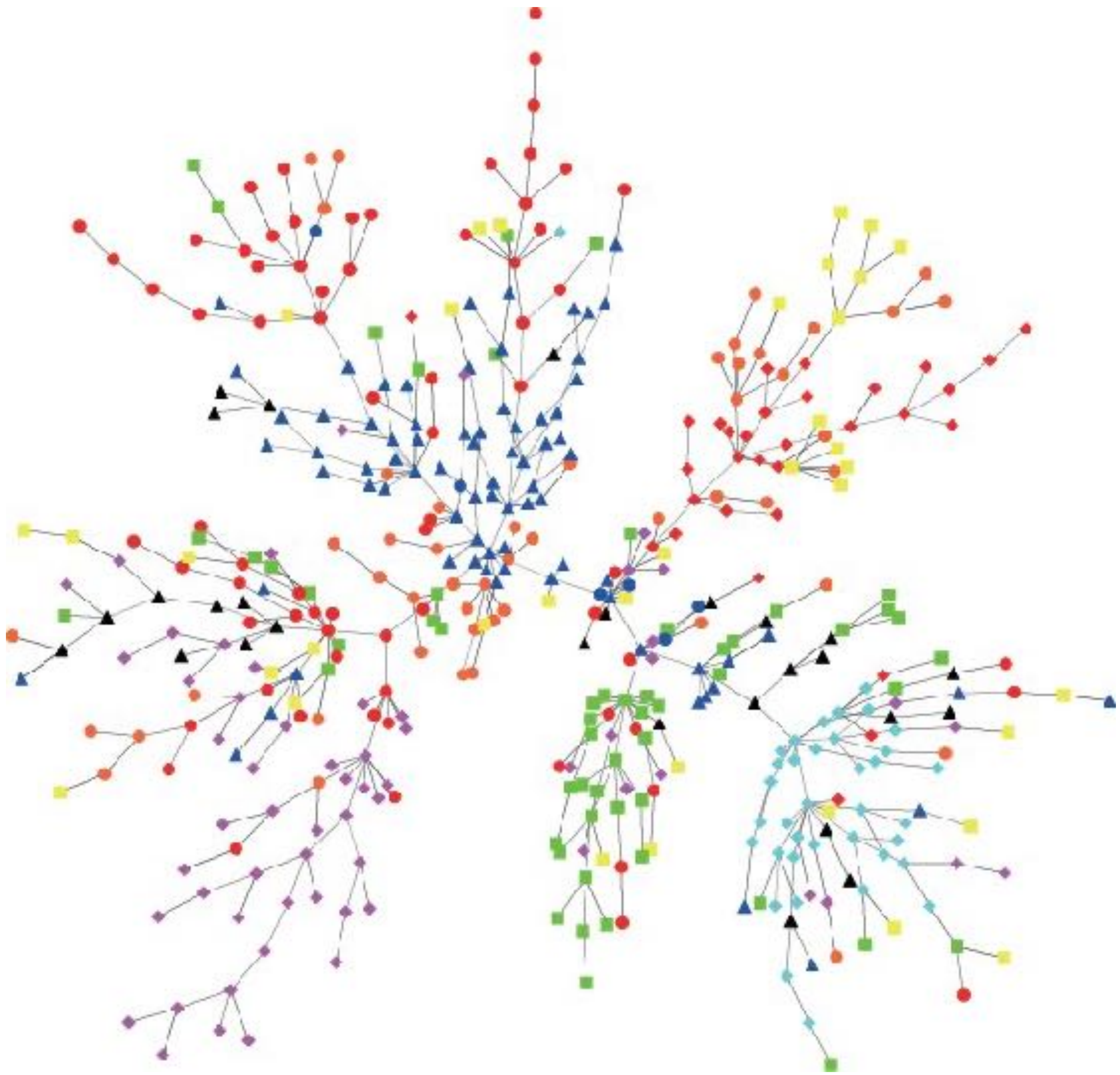
Maximum (minimum) spanning tree:
For each node only keep the connection with maximum (minimum) weight

TREES



VEF
THB
VND
PKR
JOD
PAB
JPY
USD
AED
SAR
ARS
EGP
PEN
LKR
KWD
CNY
TWD
INR
IDR
MYR
PHP
SGD
ILS
COP
CLP
MXN
KRW
BRL
TRY
GBP
CAD
AUD
ZAR
NZD
HUF
PLN
RUB
CZK
EUR
RON
ISK
CHF
SEK
NOK

TREES



SUMMARY

We learned about **clustering and distances**

We learned the **tree topology** and seen its applications

We can now perform basic **network analysis**!