Small Worlds

cont.

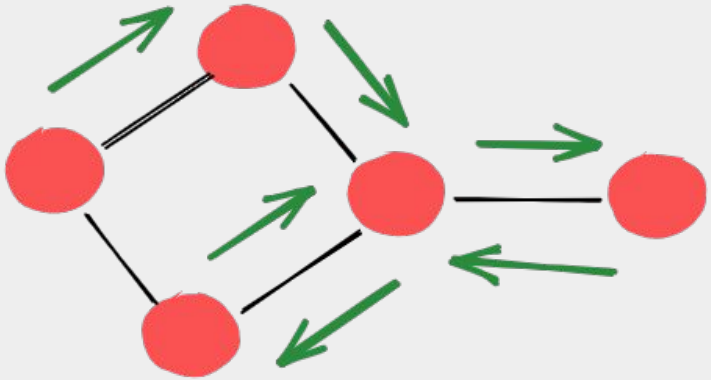
ivanovitch.silva@ufrn.br
@ivanovitchm

UFRN UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE

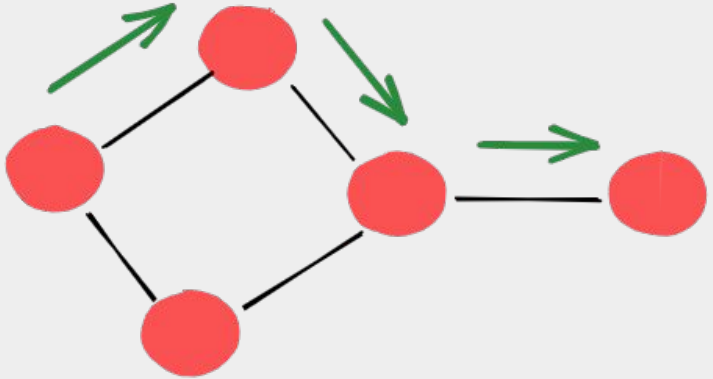
DCA

Paths
Walks
Distances

A network really shines when you use it for what it is for: **exploring its connections**.

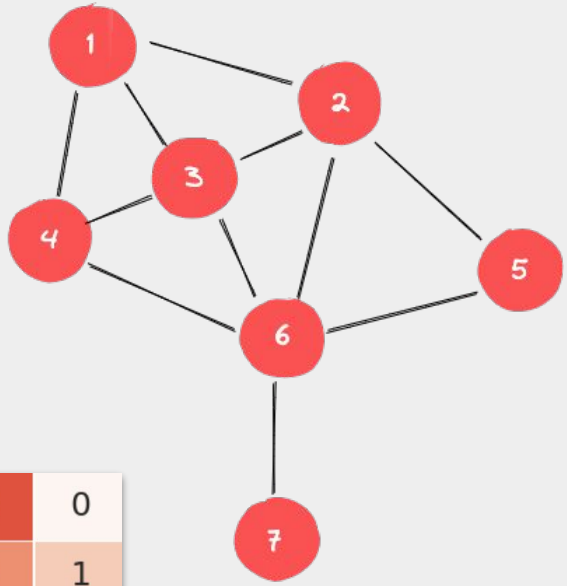


An example of a walk of lenght six in the network



An example of a path of lenght three in the network

Walks and Matrices



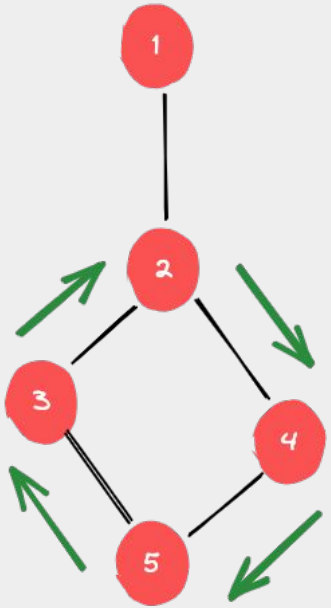
Adjacent Matrix (A)

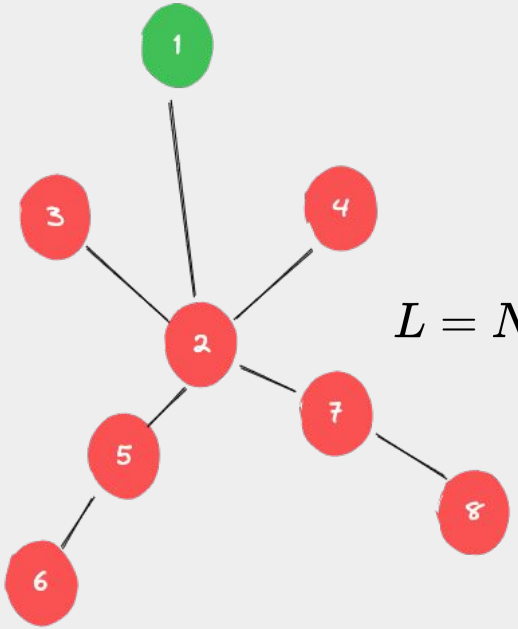
0	1	1	1	0	0	0
1	0	1	0	1	1	0
1	1	0	1	0	1	0
1	0	1	0	0	1	0
0	1	0	0	0	1	0
0	1	1	1	1	0	1
0	0	0	0	0	1	0

A^2

3	1	2	1	1	3	0
1	4	2	3	1	2	1
2	2	4	2	2	2	1
1	3	2	3	1	1	1
1	1	2	1	2	1	1
3	2	2	1	1	5	0
0	1	1	1	1	0	1

Cycles





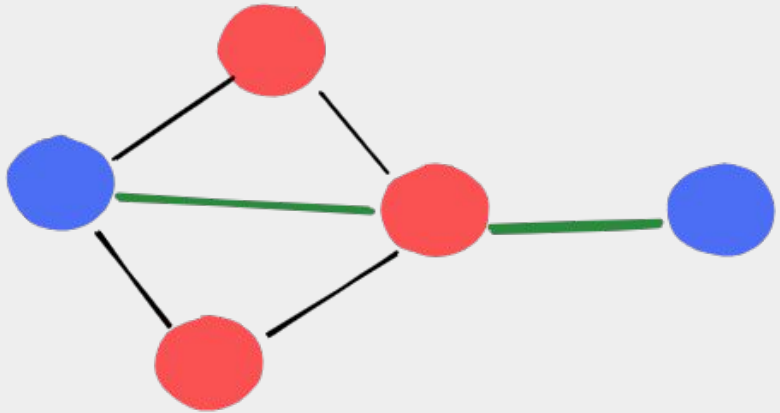
$$L = N - 1$$

```
# Return all cycles of G
nx.cycle_basis(G)
[[3, 5, 4, 2]]

# G is a tree?
nx.is_tree(G)
False
```

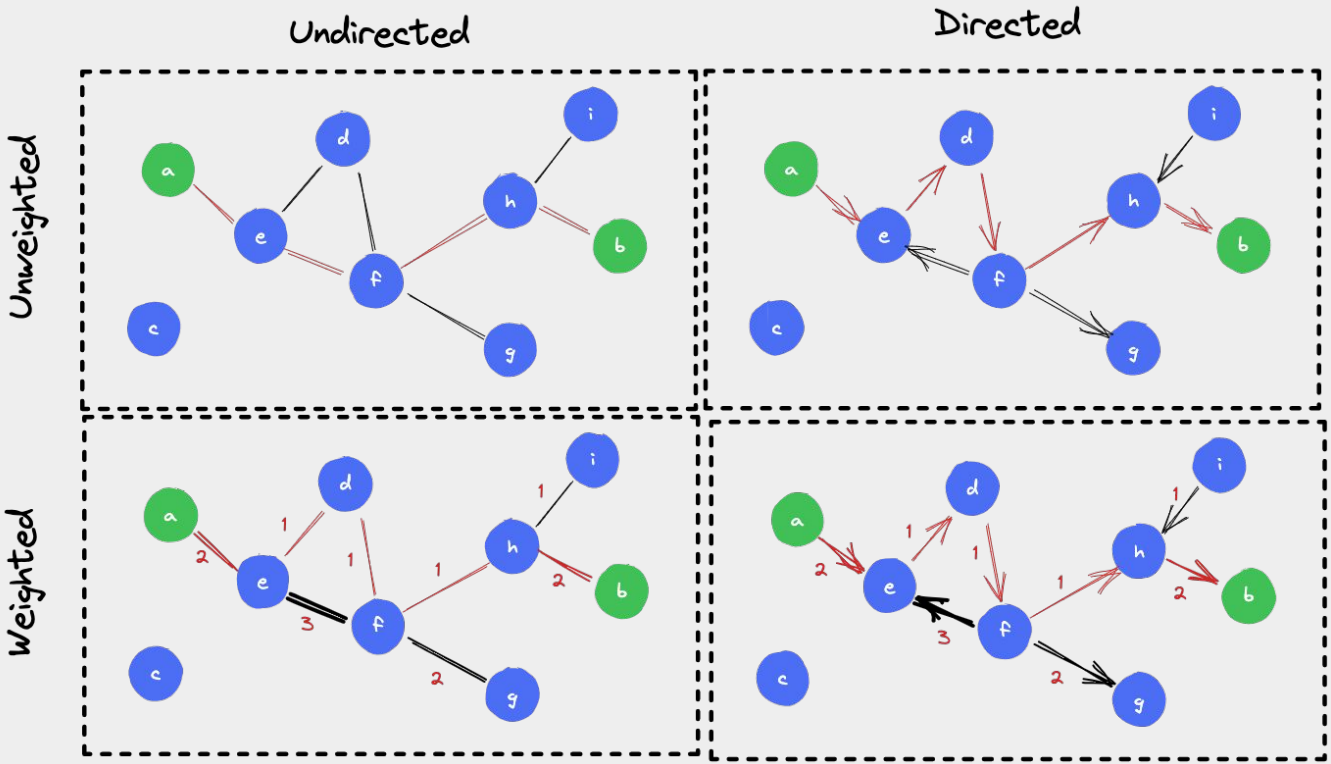
```
# Return all cycles of G
nx.cycle_basis(G)
[]

# G is a tree?
nx.is_tree(G)
True
```

Distance

The concept of a **path** is the basis of the definition of **distance** among nodes in a network. The natural distance measure between two nodes is defined as the minimum number of links that must be traversed in a path connecting the two nodes. Such a path is called the **shortest path**, and its length is called the shortest-path length.



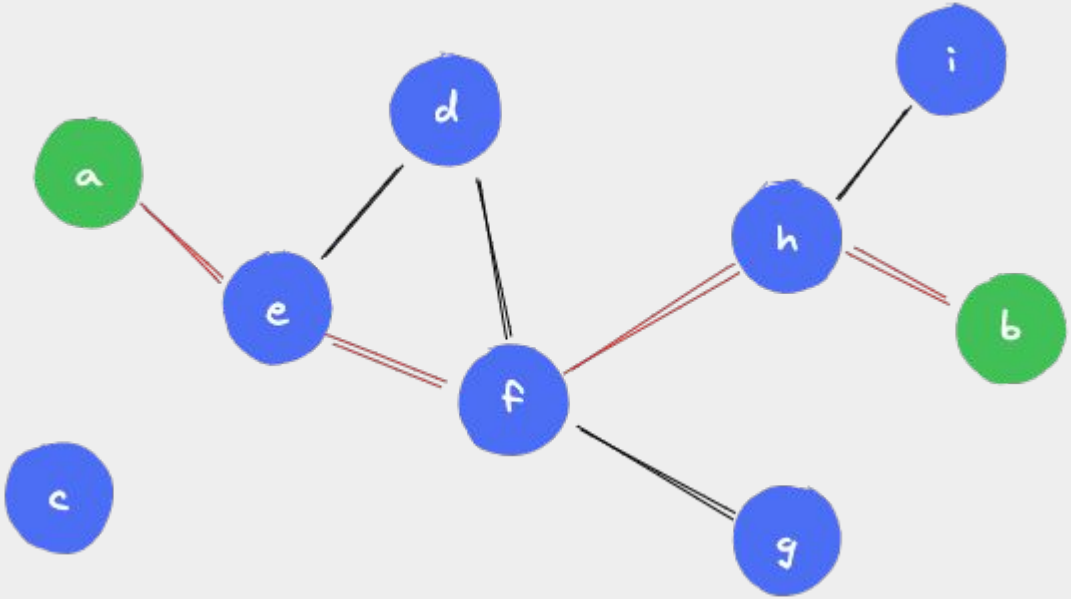
Shortest Path



Average Shortest Path Length

$$\langle l \rangle = \frac{\sum_{ij} l_{ij}}{\binom{N}{2}} = \frac{2\sum_{ij} l_{ij}}{N(N-1)}$$

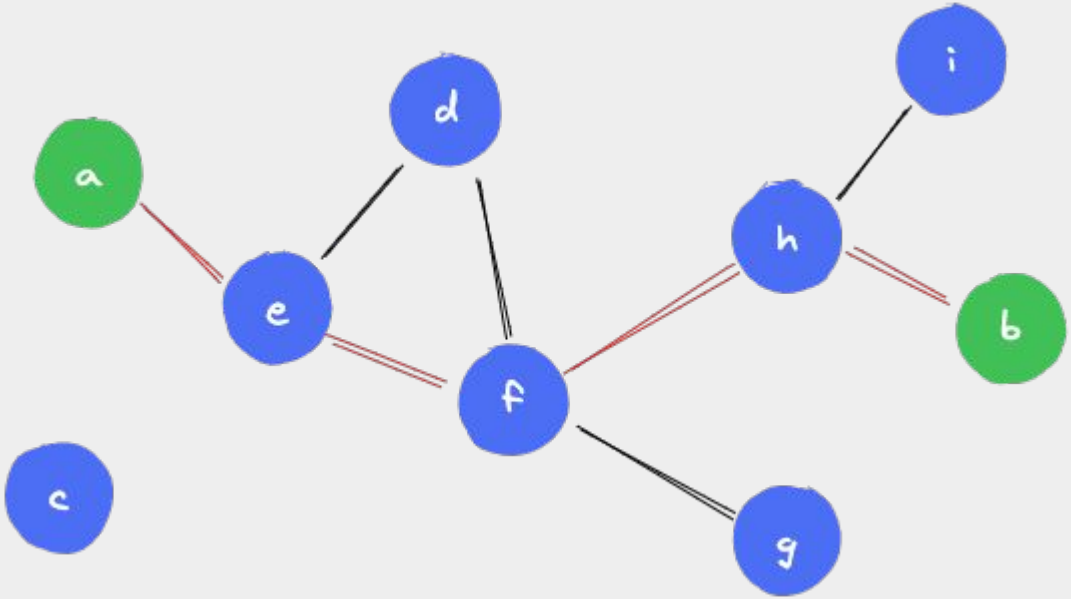
$$\langle l \rangle = \frac{\left(\sum_{ij} \frac{1}{l_{ij}}\right)^{-1}}{\binom{N}{2}}$$



Diameter of Network

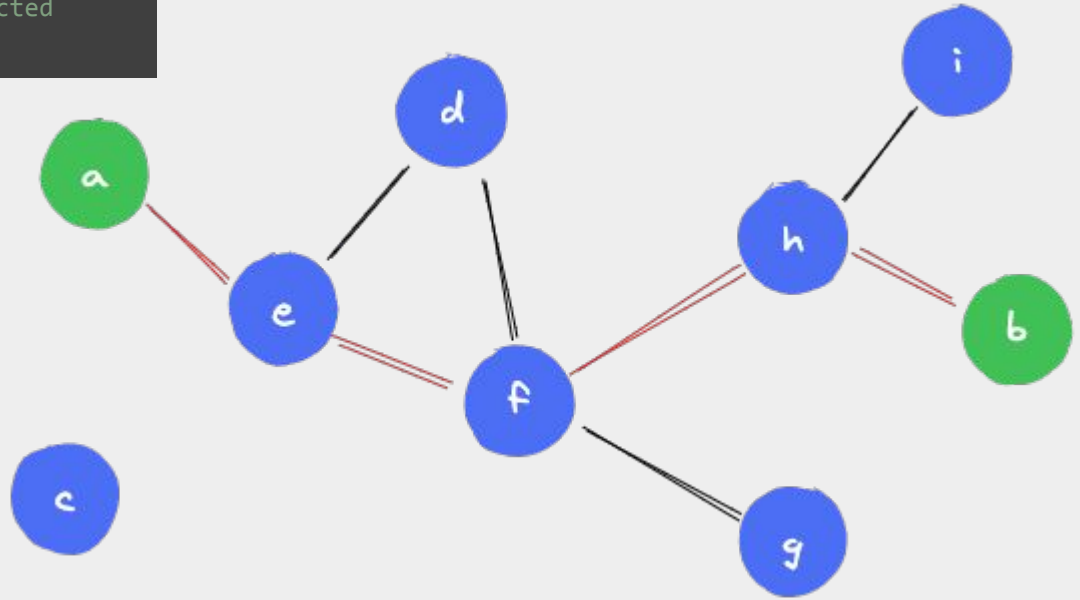
$$l_{max} = \max_{ij} \; l_{ij}$$

(src, dest)	(b,c)	....
(a,b)	-	
a - e - f - h - b	(b,d)	
(a,c)	b - h - f - d	
-	(b,e)	
(a,d)	b - h - f - e	
a - e - d	(b,f)	
(a,e)	b - h - f	
a - e	....	
....		



```
nx.has_path(G, 'a', 'c')            # False
nx.has_path(G, 'a', 'b')            # True
nx.shortest_path(G, 'a', 'b')       # ['a','e','f',h','b']
nx.shortest_path_length(G,'a','b') # 4
nx.shortest_path(G, 'a')            # dictionary
nx.shortest_path_length(G, 'a')     # dictionary
nx.shortest_path(G)                 # all pairs
nx.shortest_path_length(G)          # all pairs
nx.average_shortest_path_length(G) # error
G.remove_node('c')                  # make G connected
nx.average_shortest_path_length(G) # now okay
```

```
{'a': {'a': ['a'],
  'b': ['a', 'e', 'f', 'h', 'b'],
  'd': ['a', 'e', 'd'],
  'e': ['a', 'e'],
  'f': ['a', 'e', 'f'],
  'g': ['a', 'e', 'f', 'g'],
  'h': ['a', 'e', 'f', 'h'],
  'i': ['a', 'e', 'f', 'h', 'i']},
 'b': {'a': ['b', 'h', 'f', 'e', 'a'],
```



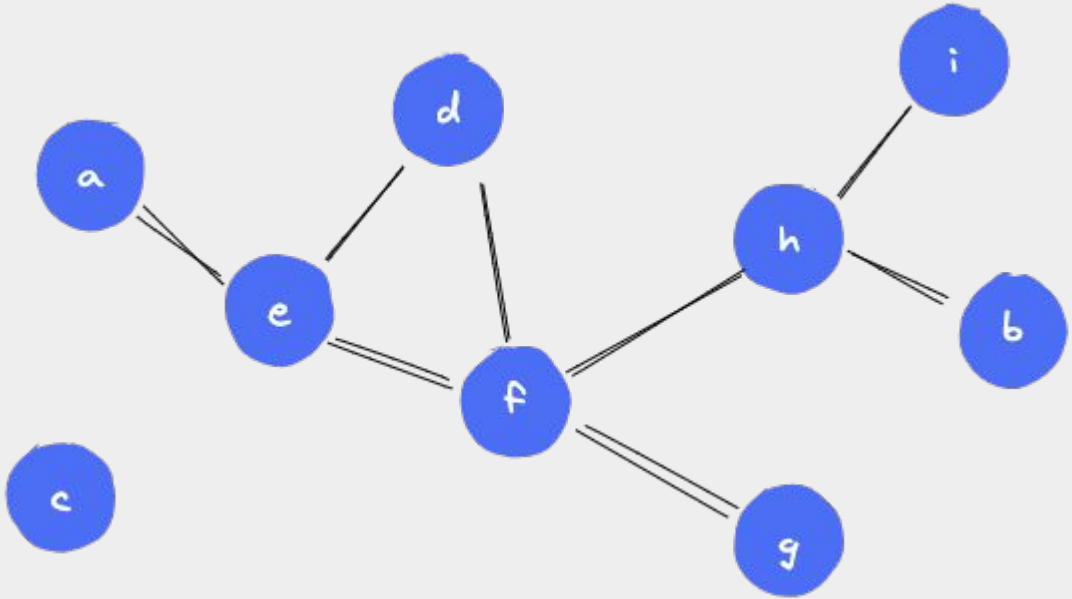

```
{'a': ['a'],
 'b': ['a', 'e', 'f', 'h', 'b'],
 'd': ['a', 'e', 'd'],
 'e': ['a', 'e'],
 'f': ['a', 'e', 'f'],
 'g': ['a', 'e', 'f', 'g'],
 'h': ['a', 'e', 'f', 'h'],
 'i': ['a', 'e', 'f', 'h', 'i']}
```

```
{'a': 0, 'b': 4, 'd': 2, 'e': 1, 'f': 2, 'g': 3, 'h': 3, 'i': 4}
```

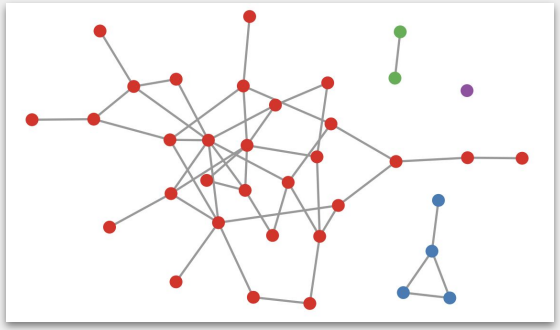
NetworkX
Network Analysis in Python

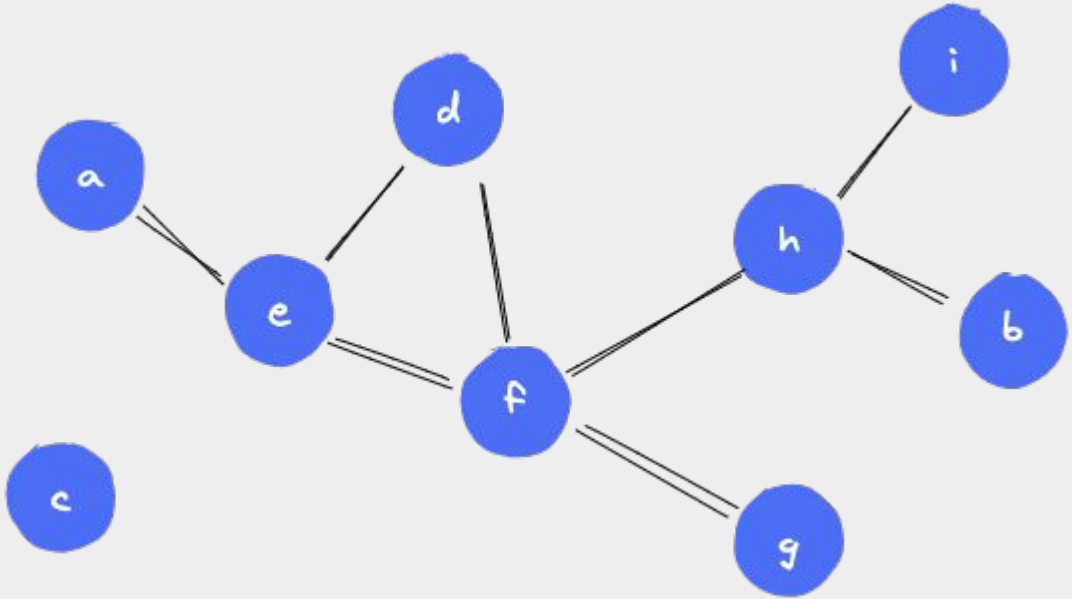
Connected Components

If two nodes cannot be connected by a walk, then they are on different **connected components**. Connected components are subgraphs whose nodes can be reached from one another by following the edges of the network.

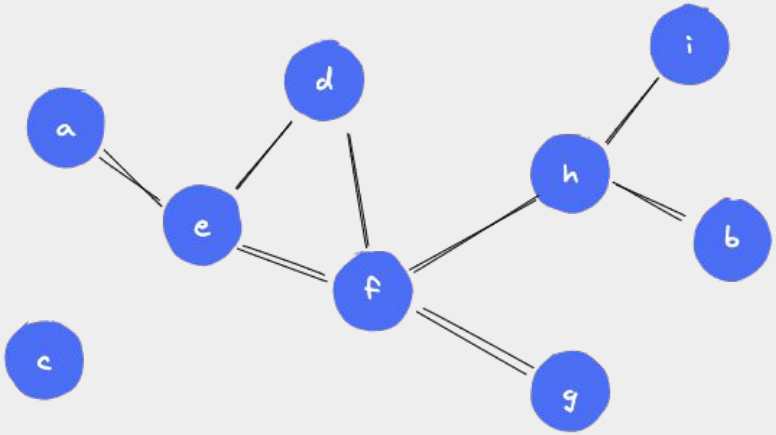


Giant Connected Components (GCC)



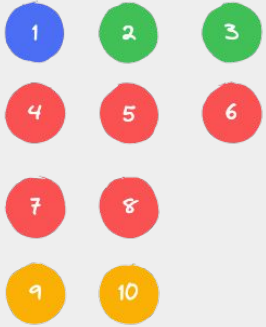


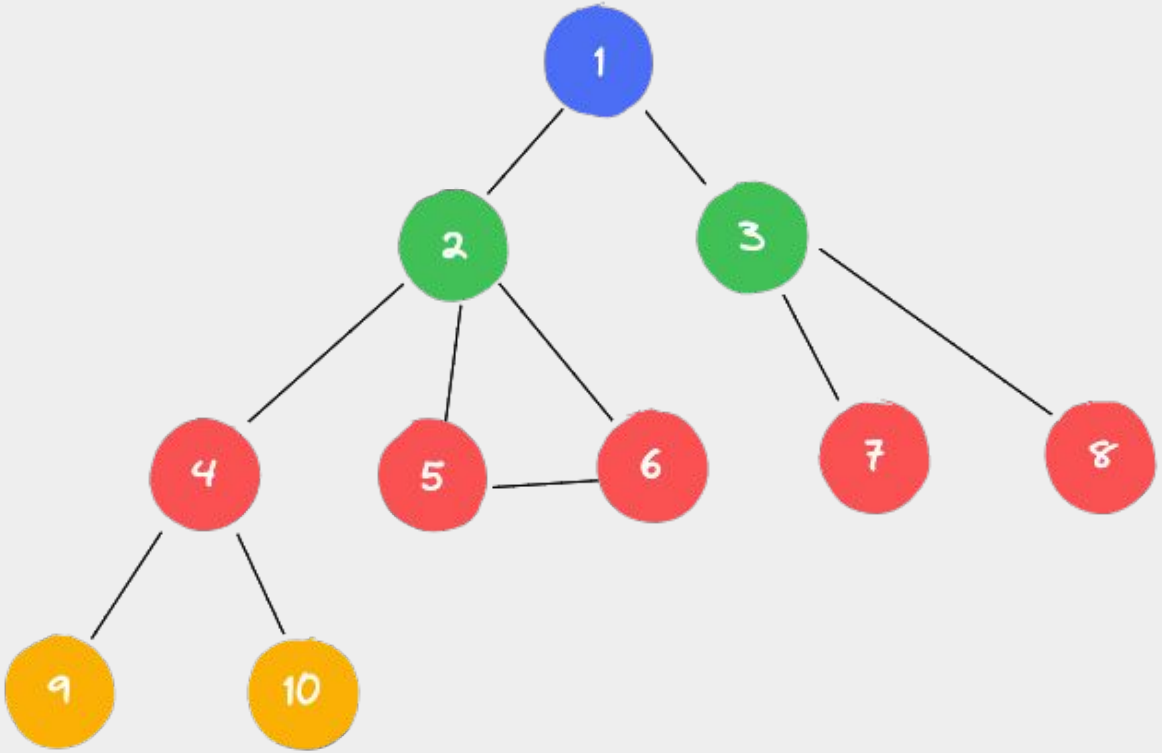
```
# G is connected or not?
nx.is_connected(G)
False

# interact under all connected component of G
for component in nx.connected_components(G):
        print(component)
{'a', 'i', 'e', 'g', 'h', 'd', 'b', 'f'}
{'c'}

# how many connected components has G?
nx.number_connected_components(G)
2

# which connected component is a node N?
nx.node_connected_component(G,"a")
{'a', 'b', 'd', 'e', 'f', 'g', 'h', 'i'}
```



Breadth-First Search (BFS) vs Depth-First Search (DFS)

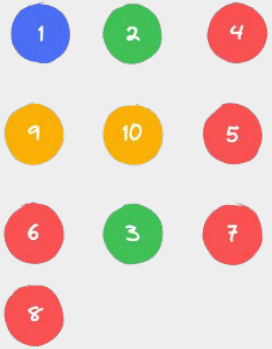
Visited Nodes

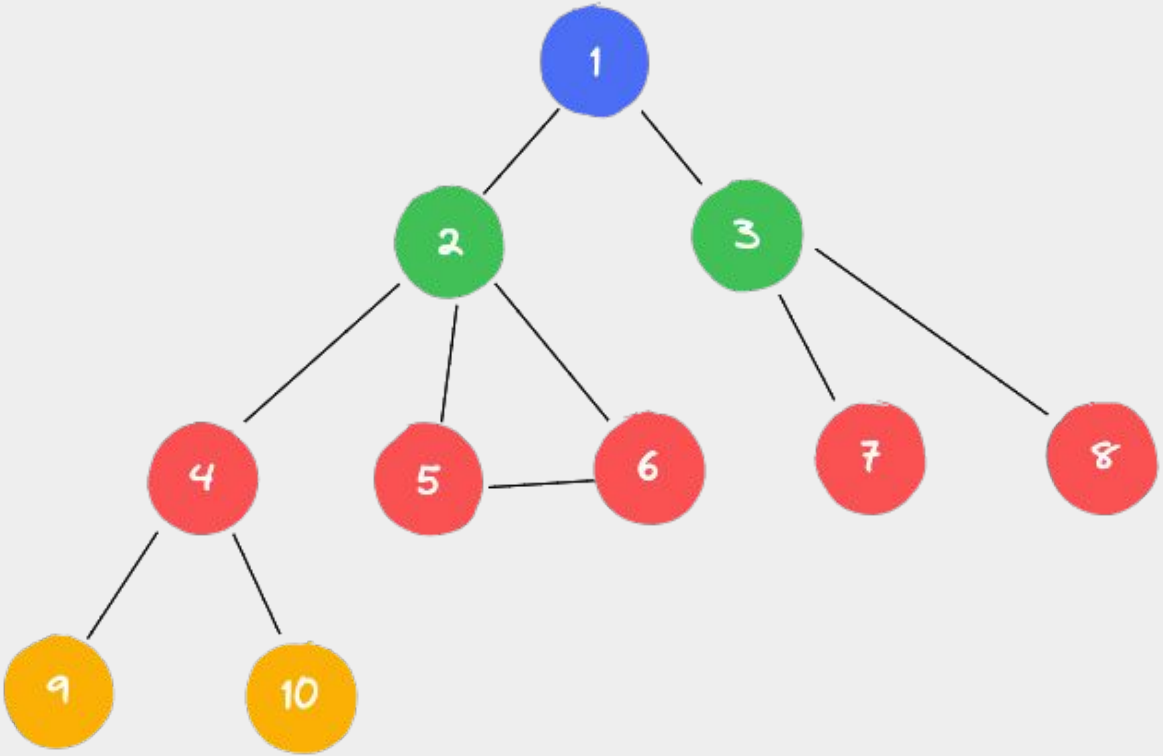




Breadth-First Search (BFS) vs Depth-First Search (DFS)

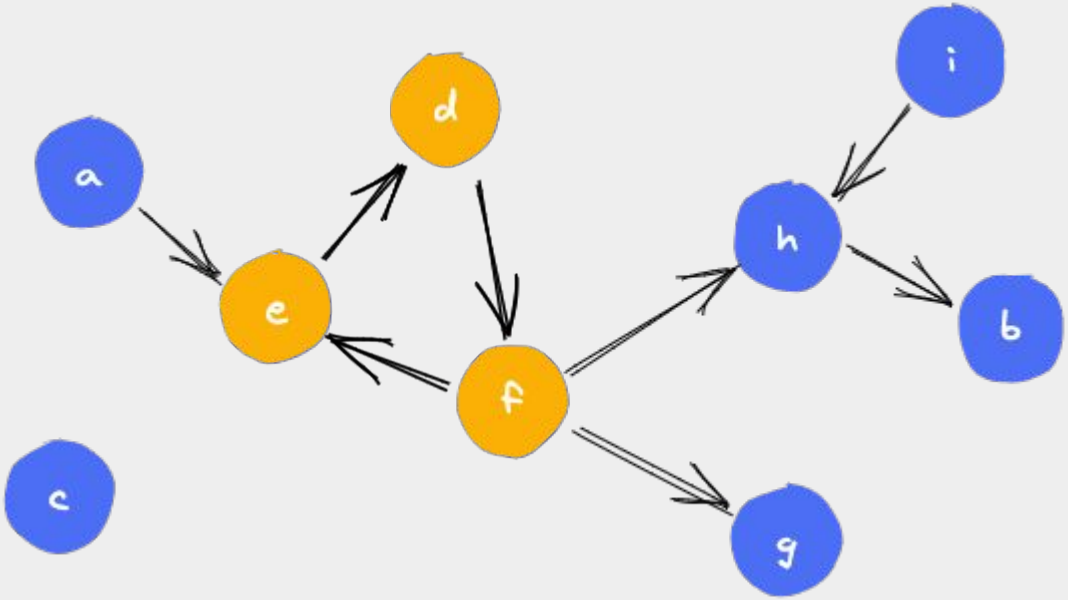
Visited Nodes



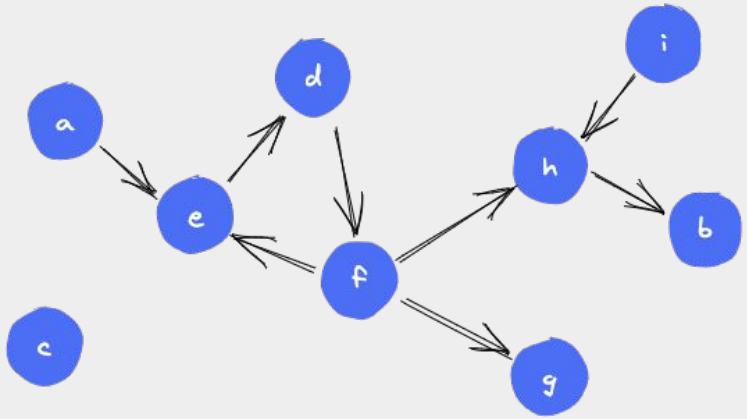


Strongly & Weakly Components (SCC vs WCC)

What does it mean a strongly or weakly network?



```
nx.is_strongly_connected(G)
False

nx.is_weakly_connected(G)
False

list(nx.weakly_connected_components(G))
[{'a', 'b', 'd', 'e', 'f', 'g', 'h', 'i'}, {'c'}]

list(nx.strongly_connected_components(G))
[{'b'}, {'h'}, {'g'}, {'d', 'e', 'f'}, {'a'}, {'i'}, {'c'}]

nx.number_strongly_connected_components(G)
7
```



NetworkX
Network Analysis in Python

Social Distance
Six Degrees of Separation
Friend of a Friend

Contents lists available at ScienceDirect

Data in Brief

journal homepage: www.elsevier.com/locate/dib

Data Article

COVID-19: A scholarly production dataset report for research analysis

Breno Santana Santos[a,b,*], Ivanovitch Silva[a], Marcel da Câmara Ribeiro-Dantas[c], Gisliany Alves[a], Patricia Takako Endo[d], Luciana Lima[a]

[a] Universidade Federal do Rio Grande do Norte (UFRN), Rio Grande do Norte, Brazil
[b] Núcleo de Pesquisa e Prática em Inteligência Competitiva (NUPIC), Universidade Federal de Sergipe (UFS), Itabaiana, SE, Brazil
[c] Institut Curie (UMR168), Sorbonne Université (EDITE), Paris, France
[d] Universidade de Pernambuco (UPE), Pernambuco, Brazil

ARTICLE INFO

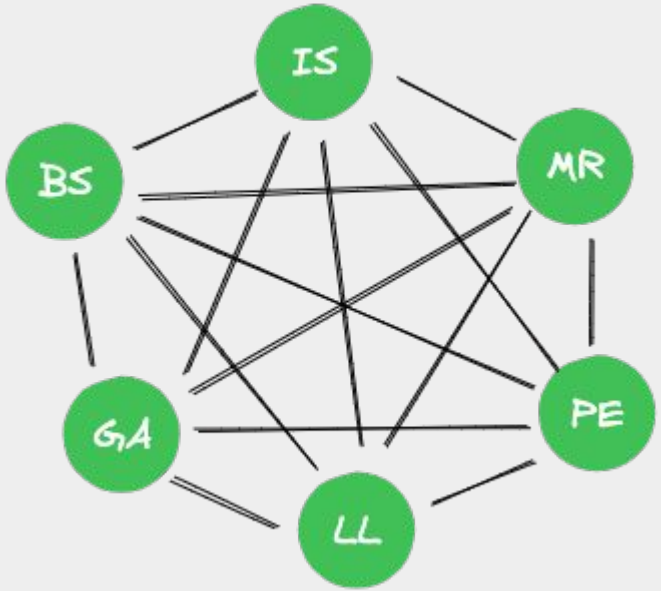
Article history:
Received 7 July 2020
Revised 6 August 2020
Accepted 12 August 2020
Available online 19 August 2020

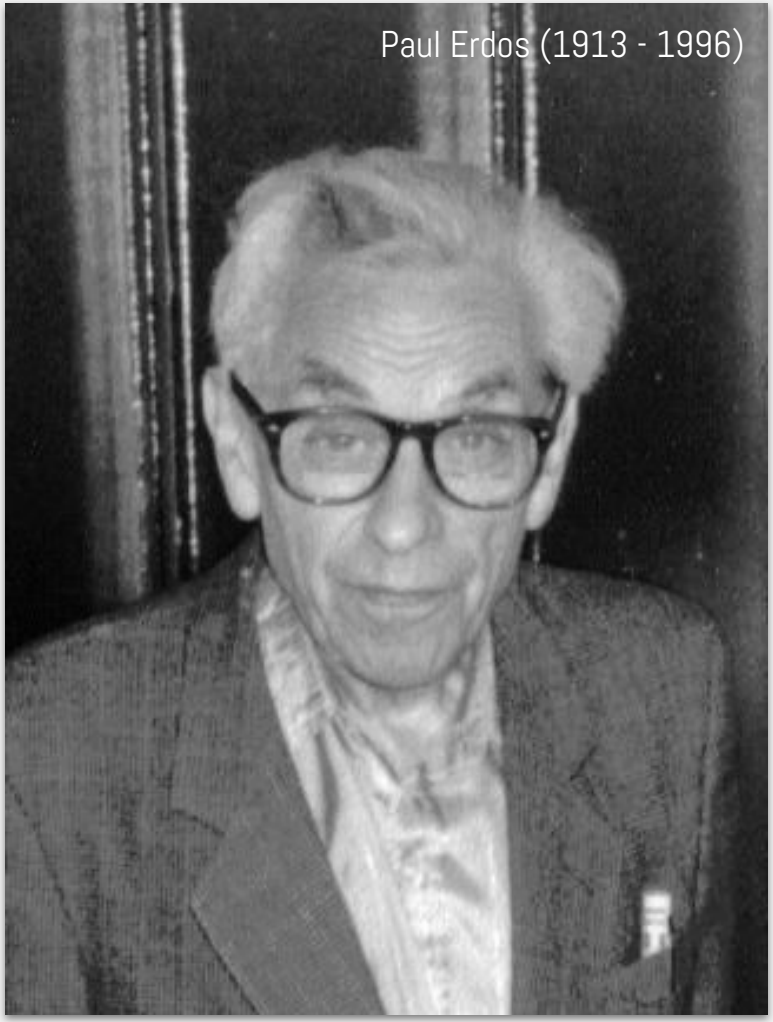
Keywords:
COVID-19
SARS-CoV-2
Pandemic
Data Science
Bibliometrics
Scientometrics

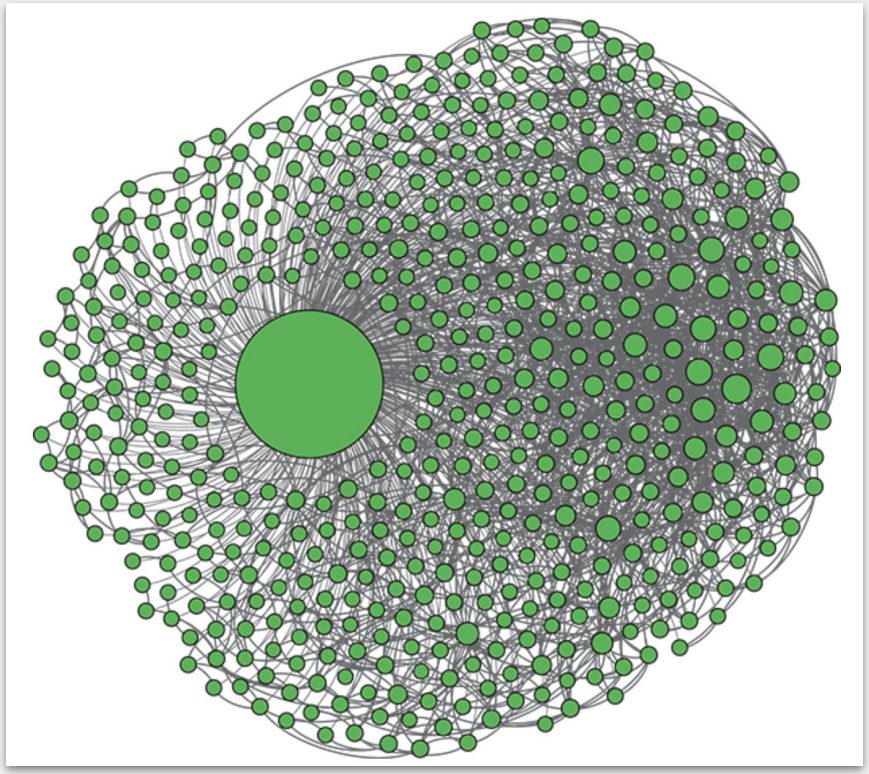
ABSTRACT

COVID-2019 has been recognized as a global threat, and several studies are being conducted in order to contribute to the fight and prevention of this pandemic. This work presents a scholarly production dataset focused on COVID-19, providing an overview of scientific research activities, making it possible to identify countries, scientists and research groups most active in this task force to combat the coronavirus disease. The dataset is composed of 40,212 records of articles' metadata collected from Scopus, PubMed, arXiv and bioRxiv databases from January 2019 to July 2020. Those data were extracted by using the techniques of Python Web Scraping and preprocessed with Pandas Data Wrangling. In addition, the pipeline to preprocess and generate the dataset are versioned with the Data Version Control tool (DVC) and are thus easily reproducible and auditable.

Coauthorship networks

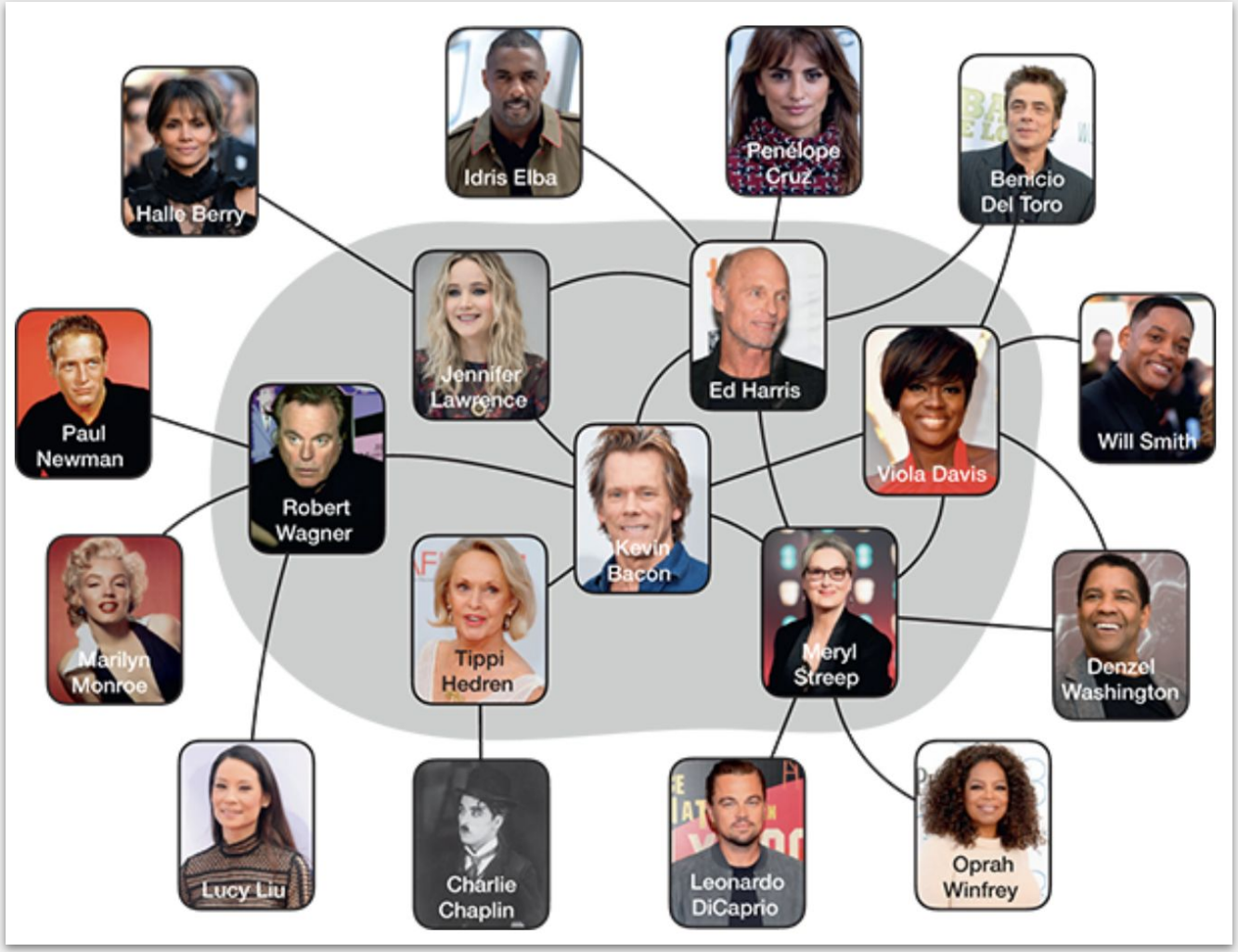






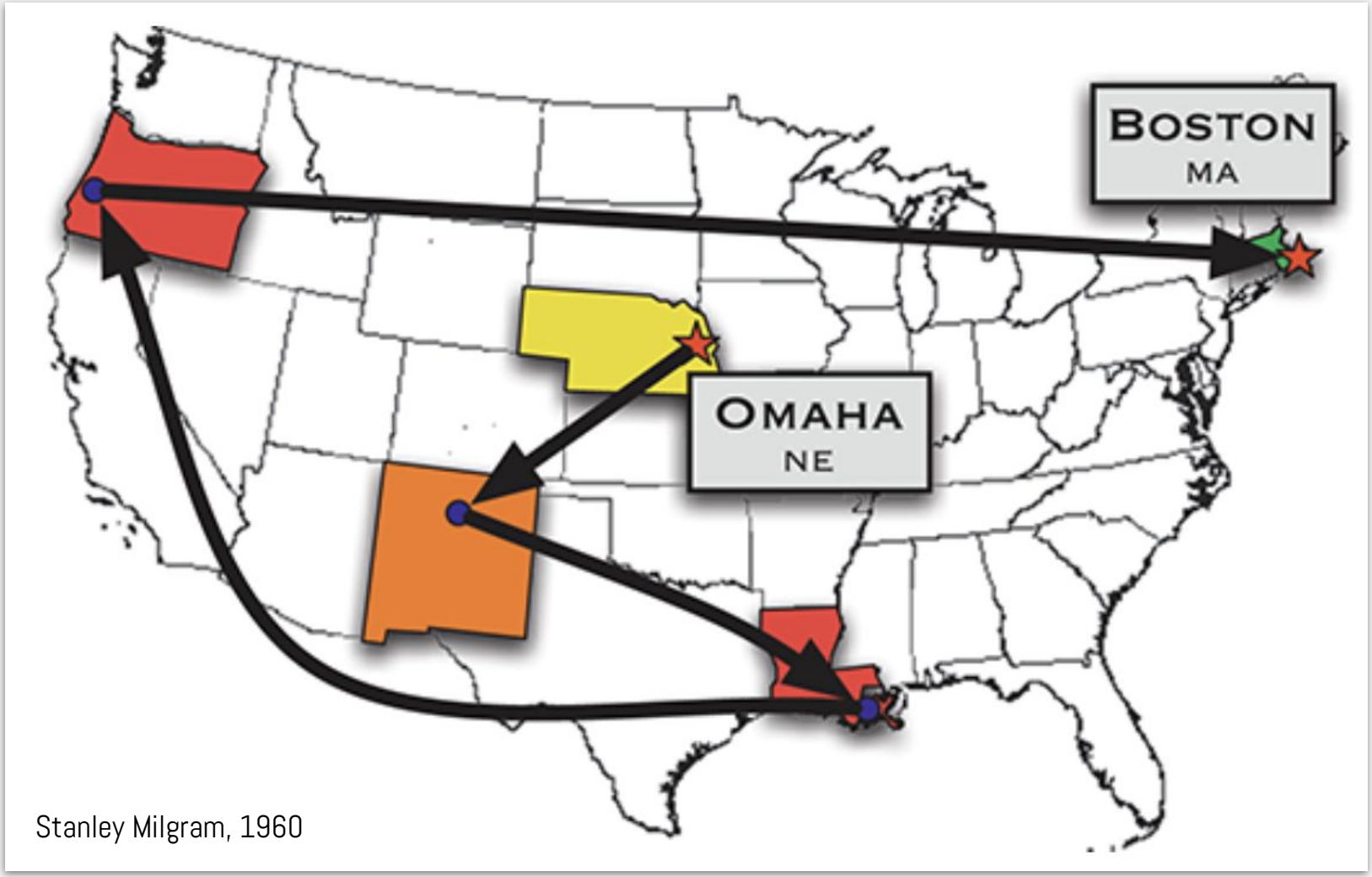
Ego Network

Paul Erdős was a famous mathematician who made critical contributions to network science. Mathematicians are fond of studying their distance in the coauthorship network from the particular node corresponding to Erdős. They call this distance their **Erdős number**.



The Oracle of Bacon

http://oracleofbacon.org/



Stanley Milgram, 1960

CURRENT ROUND

START
Walmart

GOAL
Hebrew language

PLAY NOW!
21s

ROUND RESULTS | DAY LEADERS | WEEK LEADERS | ALL-TIME LEADERS

YOUR WINS ARE NOT SAVED! CREATE ACCOUNT! | LOGIN

Boy Scouts of America → Sodium chloride
(4 minutes ago) ← →

#1.**TealDeer46**(2000pts)

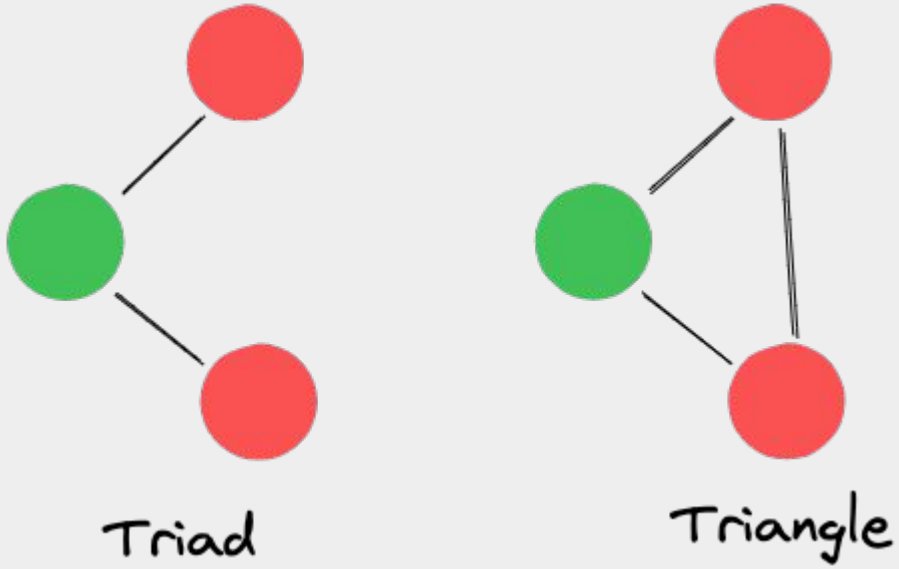
WIN #1
Boy Scouts of America → New Mexico → Mining → Sodium chloride
(800PTS · 4CLICKS · 19SECS)

WIN #2
Boy Scouts of America → New Mexico → Economy of New Mexico → Manganese → Chlorine → Sodium chloride
(1200PTS · 6CLICKS · 82SECS)

#2.**CrimsonChinchilla37**(600pts)

Feedback?

Friend of a Friend



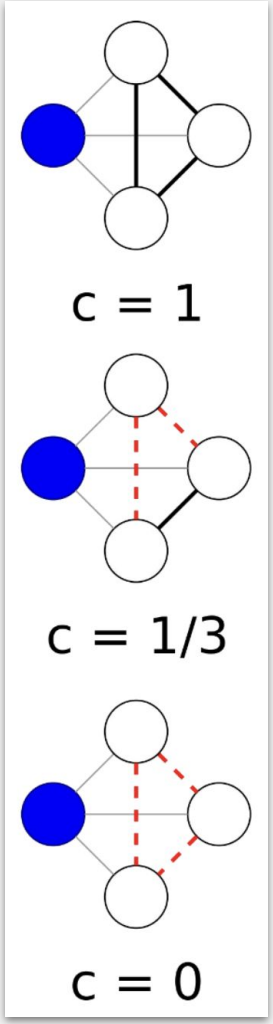
Clustering Coefficient

Some social theories consider triads essential units of social network analysis.

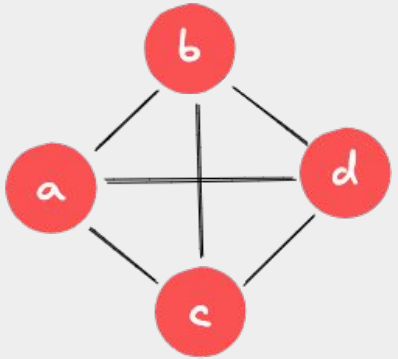
- The **clustering coefficient** is the fraction of possible triangles that contain the ego
- Think of the clustering coefficient as a measure of the **"stardom"**

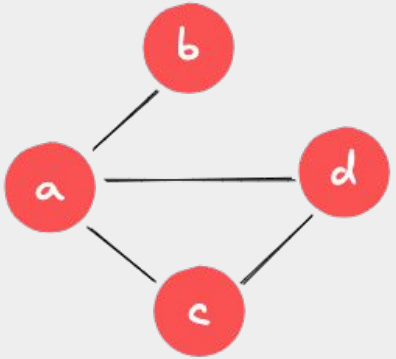
$$C(i) = \frac{\tau(i)}{\tau_{max}(i)} = \frac{\tau(i)}{\binom{k_i}{2}} = \frac{2\tau(i)}{k_i(k_i-1)}$$

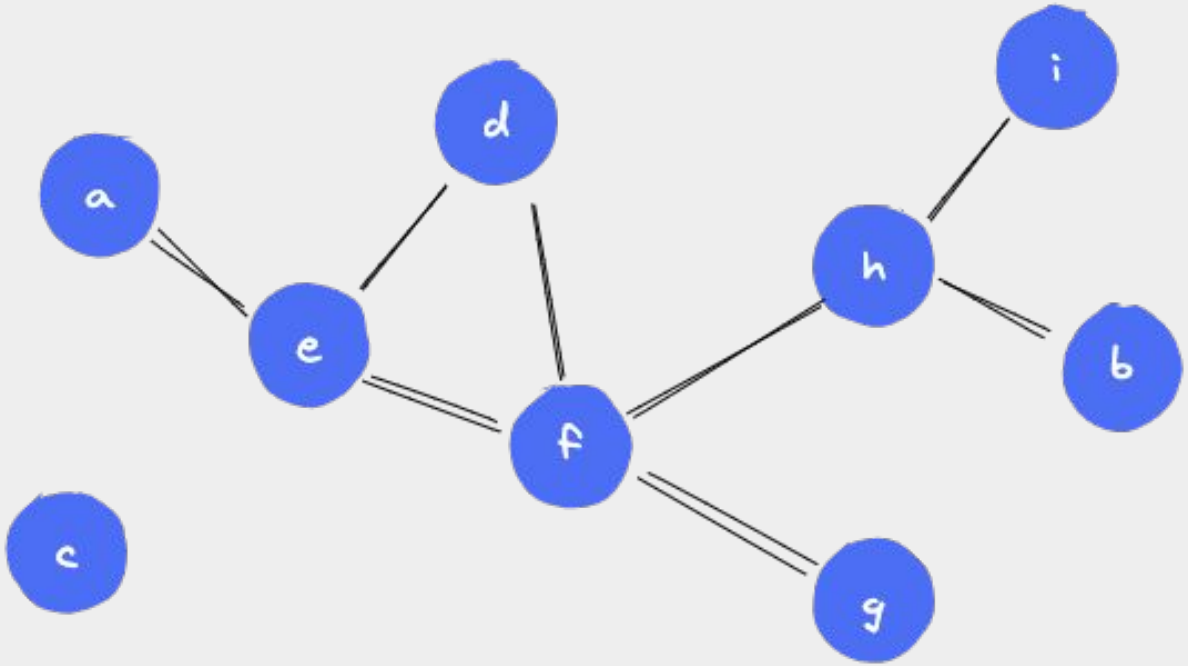
$$C = \frac{\sum_{i;k_i>1} C(i)}{N_{k>1}}$$



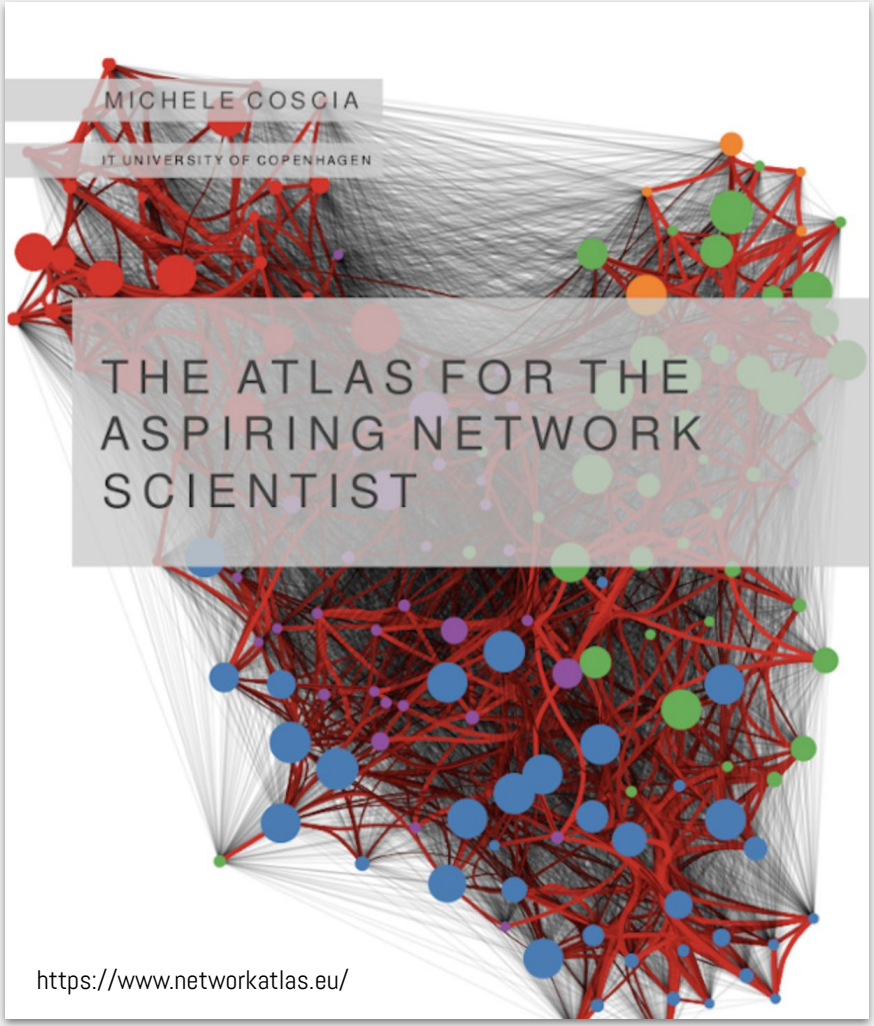
```
nx.triangles(G)
{'a': 3, 'b': 3, 'c': 3, 'd': 3}

nx.clustering(G,"a")
1.0

nx.clustering(G)
{'a': 1.0, 'b': 1.0, 'c': 1.0, 'd': 1.0}

nx.average_clustering(G)
1

G.remove_edge("b","c")
G.remove_edge("b","d")

nx.clustering(G)
{'a': 0.3333333333333333, 'b': 0, 'c': 1.0, 'd': 1.0}

nx.average_clustering(G)
0.5833333333333333
```








```
nx.average_clustering(G)
0.16666666666666666
```



Further Reading

Chapter 07 Paths & Walks
Chapter 09 Density
Chapter 10 Shortest Paths