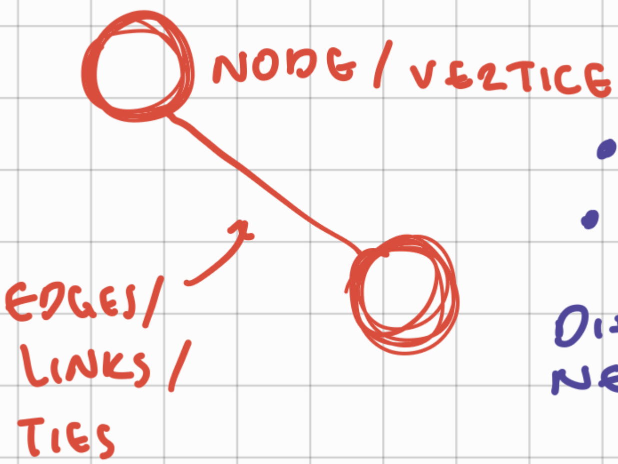
INFORMATION NETWORKS

HAVE A LOOK AT NETWORKS

z.B.: NETWORK OF WIKIPEDIA ARTICLES ABOUT CLIMATE CHANGE AND THEIR CLUSTERING. [EMAPS].

A FOODWEB IS ALSO A NETWORK.

A NETWORK (OR A GRAPH) IS A REPRESENTATION OF CONNECTIONS AMONG DIFFERENT ITEMS.



- SYMMETRIC ⟷
- ASYMMETRIC ⟶

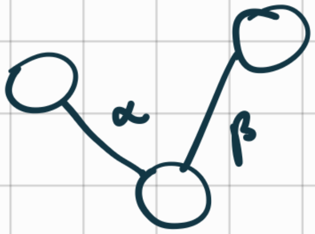
DIFFERENT TYPES OF NETWORK.

PYTHON: networkx as nx

UNDIRECTED NETWORK
```
g = nx.Graph( )
g.add_edge( . , . )
```

DIRECTED NETWORKS
```
d = nx.DiGraph( )
d.add_edge( .→, . )
```

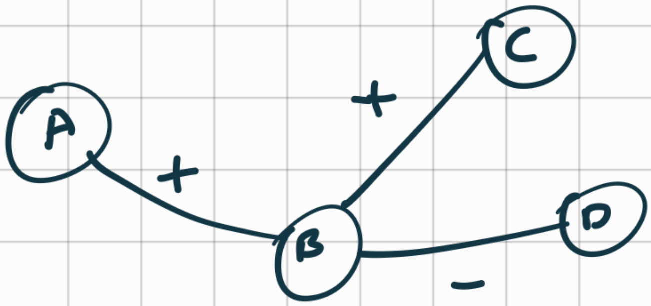
WEIGHTED NETWORKS: w = nx.Graph( )



w.add_edge(·, ·, weight = α)

SIGNED NETWORKS: It's possible to study networks whose links carry a positive or negative connotation.

s. = nx.Graph( )



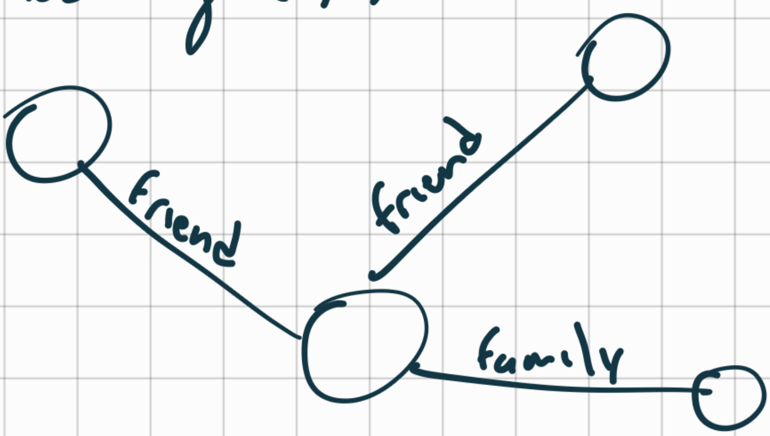
s.add_edge(·, ·, sign = "+")
= "-")

{BOOLEAN}

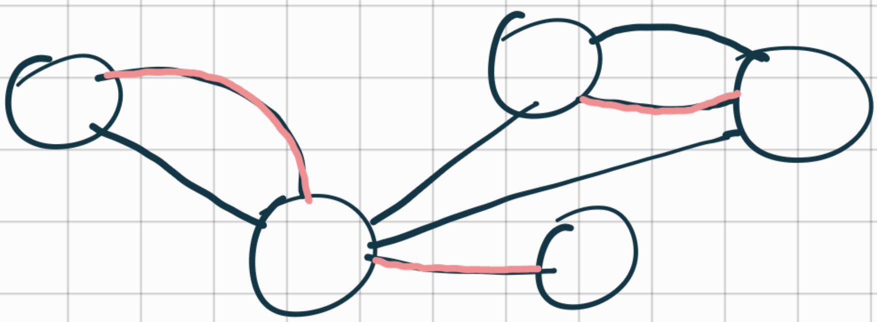
OTHER ATTRIBUTES

g = nx. Graph( )

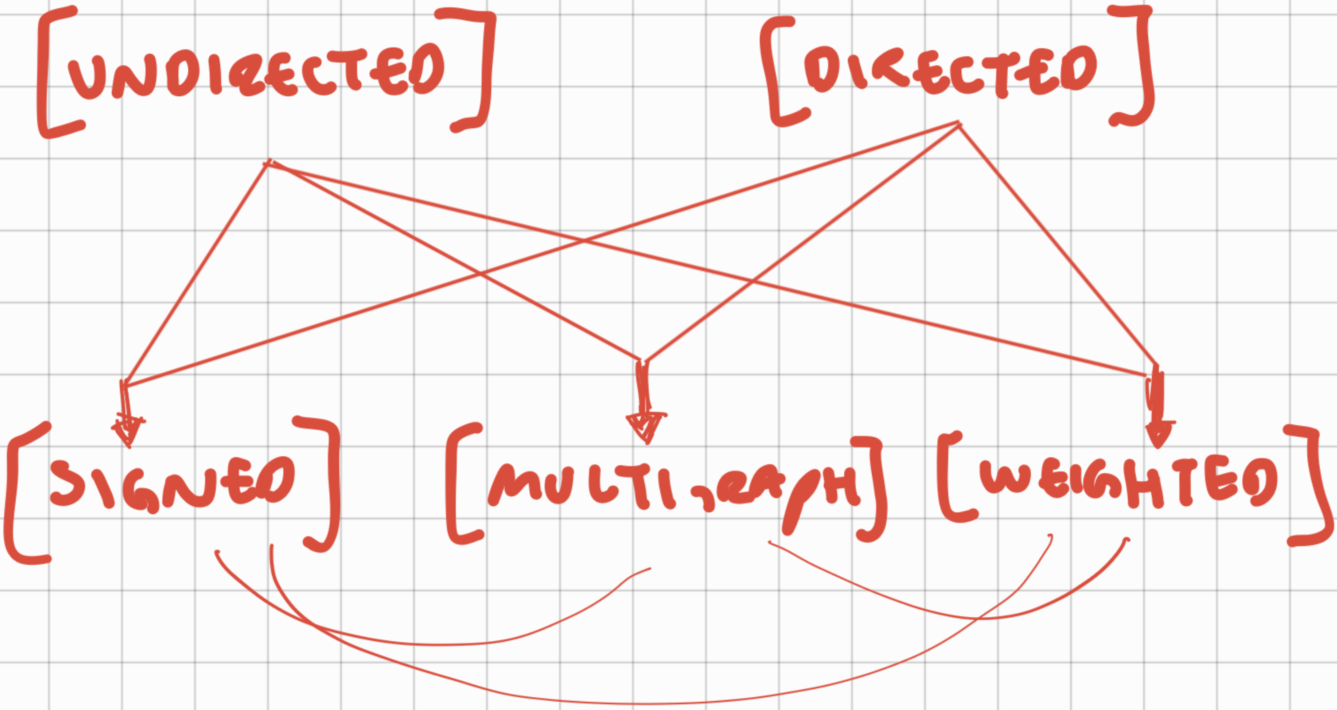
g.add_edge (·, ·, relation = "<attribute>")



MULTIGRAPHS: A pair of nodes that can have different types of relationships simultaneously.



SUMMARY



PYTHON TOOLS

```
#LIST OF EDGES
  g.edges()

# LIST OF EDGES WITH ALL ATTRIBUTES
  g.edges(data=True)
```

SPECIFIC ATTRIBUTES; z.B.: "relation"

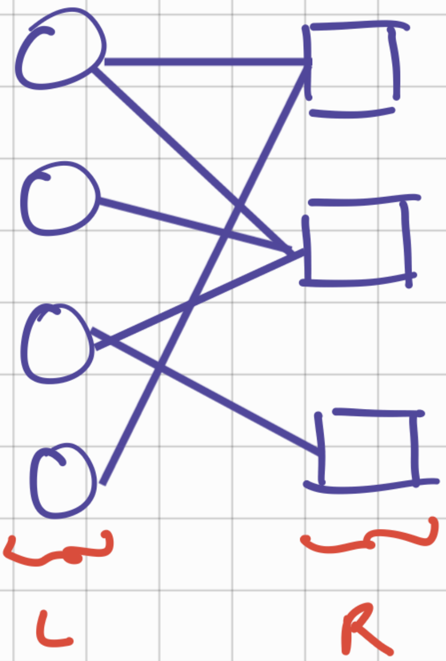
g.edges(data="relation")

ACCESSING A SPECIFIC EDGE

g.edge[•][•] # will print all attributes.

BIPARTITE GRAPHS

WHEN A GRAPH CAN BE SPLIT INTO TWO SETS, L AND R, AND EVERY EDGE CONNECTS A NODE IN L WITH A NODE IN R.



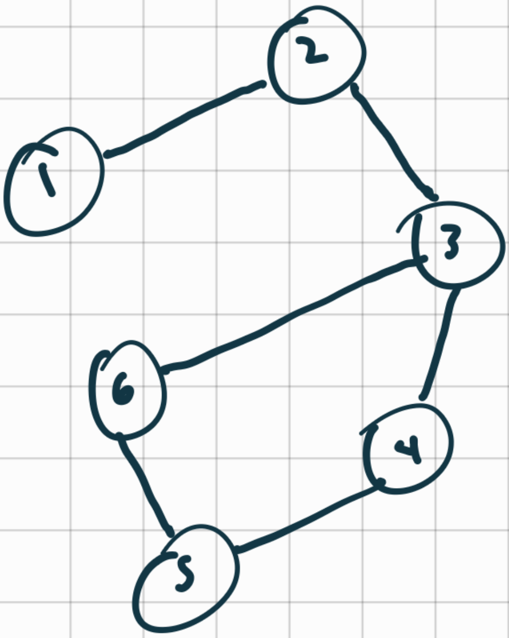
from networkx.algorithms \
import bipartite

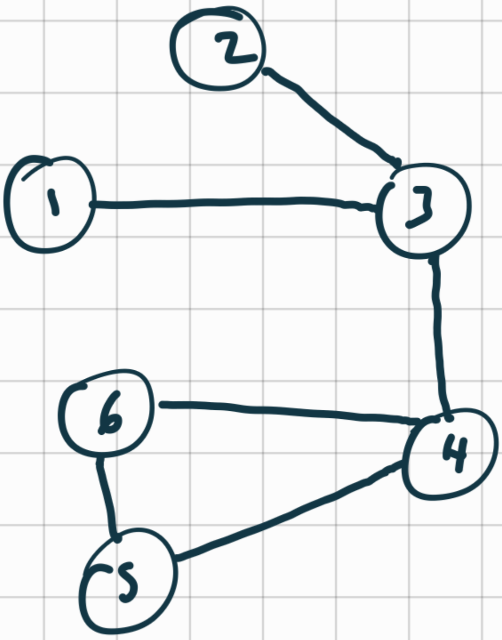
{ VERY EASY TO CREATE :) }

MODULE 7, VIDEO
t = 3:40

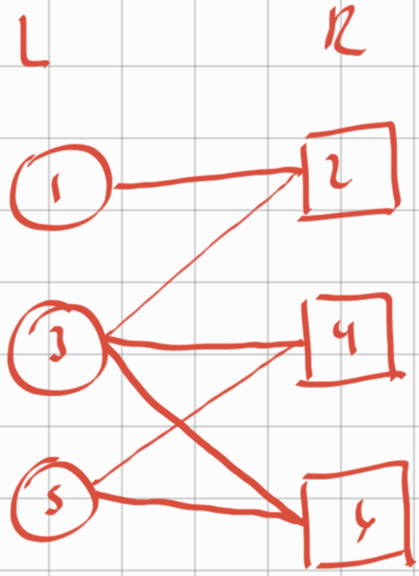
REMARKS OF A BIPARTITE GRAPH:

- CANNOT CONTAIN A CYCLE OF AN ODD NUMBER OF NODES; z.B.:

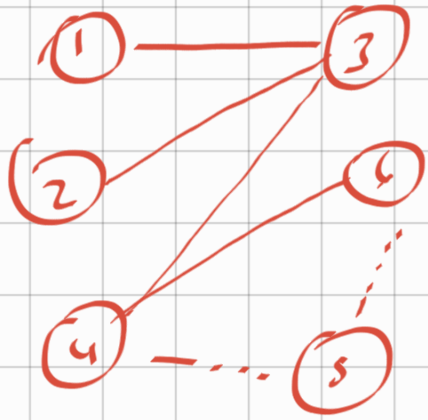




{1, 3, 5} ; {2, 4, 6}



NOT BIPARTITE.



IN PYTHON, IT'S POSSIBLE TO CHECK IF A GRAPH IS BIPARTITE:

`g.is_bipartite(g)` } For a specific node.

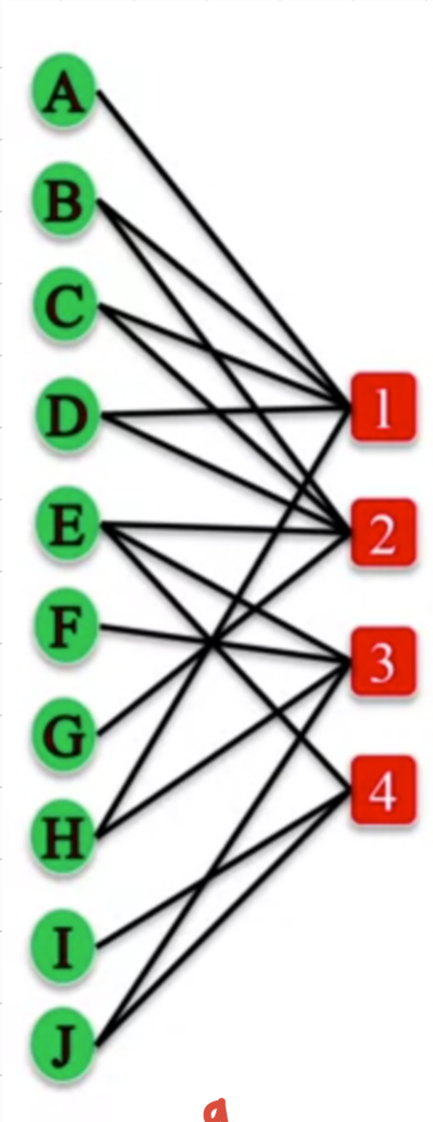
`g.is_bipartite_node_set(g, <>)` } For a specific set of nodes.

IT'S POSSIBLE TO GET THE SET OF NODES THAT BIPARTITE.

```
bipartite.sets(g)
```

L-BIPARTITE GRAPH PROJECTION

Network of nodes in group Γ, where a pair of nodes is connected if they have a common neighbour in Λ in the bipartite graph. z.B.:

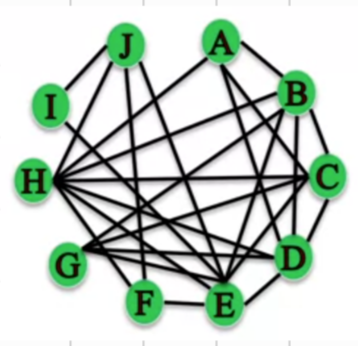


g

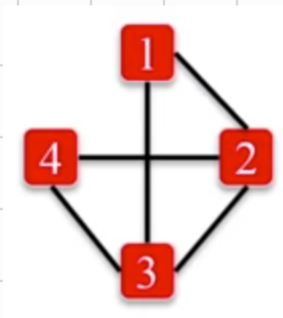
$\Lambda = \{1, 2, 3, 4\}$

$\Gamma = \{A, B, C, D, E, F, G, H, I, J\}$

```
#Projection on Γ
γ = bipartite.projected_graph(g, Γ)
#Projection on Λ
λ = bipartite.projected_graph(g, Λ)
```

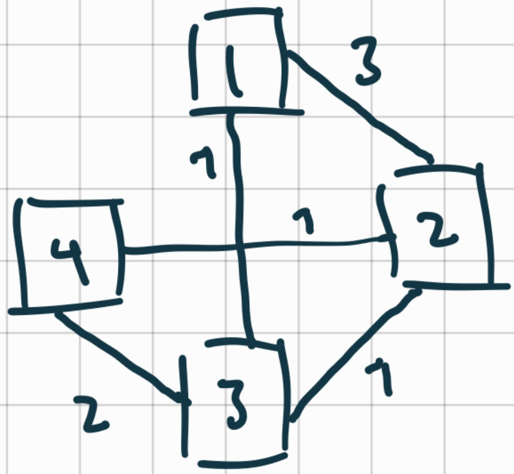


γ



λ

You can see that there's a bit of loss of information on λ when reducing g since there are more than one repetition for the connection of certain nodes. To preserve this information, one can consider the creation of a L-Bipartite weighted graph projection; i.e.: here:



BASED ON MY LIKINGS

EXAMPLE OF USAGE: RECOMMENDATION OF ARTISTS BASED ON USERS THAT LIKE SIMILAR ARTISTS.

(1) SANTA FE KLAN

NEW USER

(2) LEFTY SM

NEXT RECOMMENDATION BASED ON λ IS

(3) STILO

THEN

(4) DOBLE ONE FLOW LEGAL.

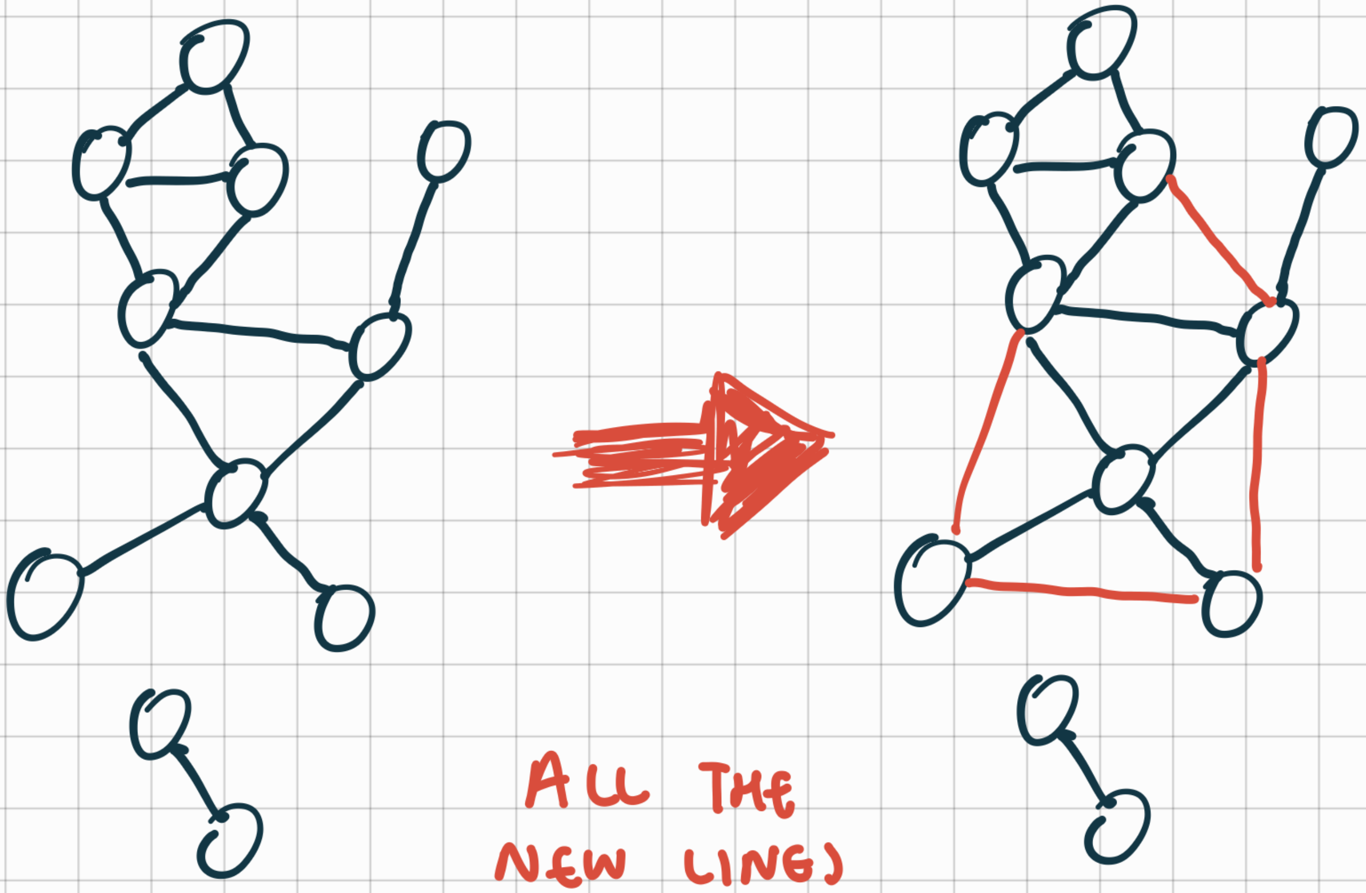
THEN

MODULE 2

TRIADIC CLOSURE:

THE TENDENCY FOR NODES WHO SHARE CONNECTIONS IN A NETWORK TO BECOME CONNECTED.

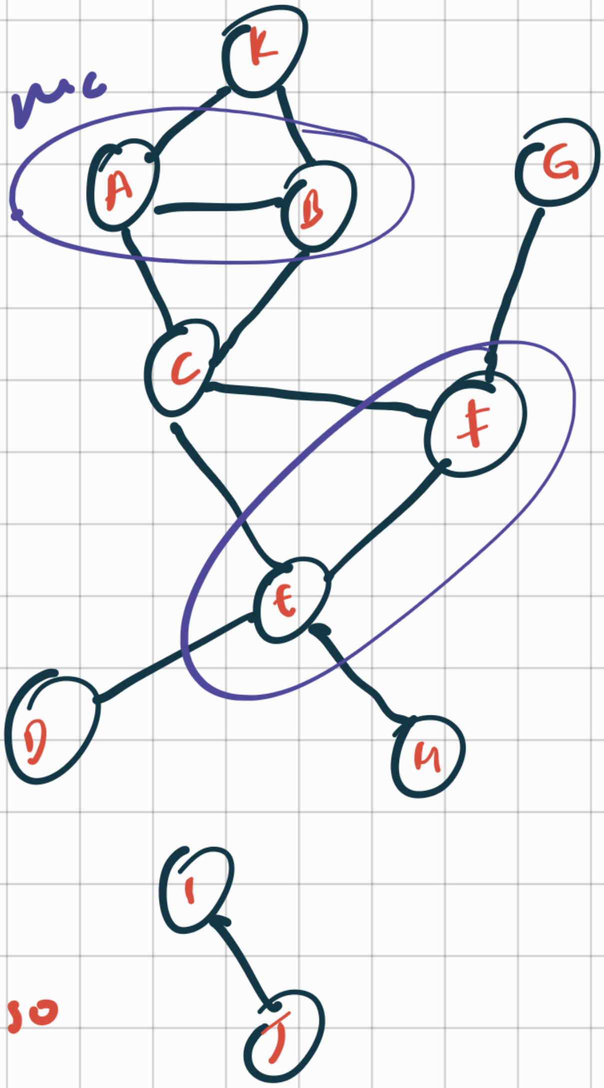
FOR INSTANCE



ALL THE NEW LINES ARE LIKELY TO OCCUR AS THEY FOR A TRIANGLE

LOCAL CLUSTERING COEFFICIENTS OF A NODE $\mathcal{K}$

FRACTION OF PAIRS OF THE NODE'S CONNECTION THAT ARE CONNECTED WITH EACH OTHER.



z.B.: $\Lambda \rightarrow C$

$$\mathcal{K}_\Lambda = \frac{m_\Lambda}{n_\Lambda}$$

$\Lambda \equiv$ NODE TAG

$\mathcal{K}_\Lambda \equiv$ LOCAL CLUSTERING COEFFICIENT OF NODE Λ

$m_\Lambda \equiv$ # OF PAIRS OF Λ's CONNECTIONS THAT ARE ALSO CONNECTED

$d_\Lambda \equiv$ # OF DIRECT CONNECTIONS TO Λ

$n_\Lambda \equiv$ # OF PAIRS OF Λ's CONNECTIONS

$$n_\Lambda \hat{=} \frac{d_\Lambda(d_\Lambda - 1)}{2}$$

$$\kappa_c = \frac{m_c}{n_c} = \frac{2}{\left[\frac{4(4-1)}{2}\right]} = \frac{4}{4(4-3)}$$

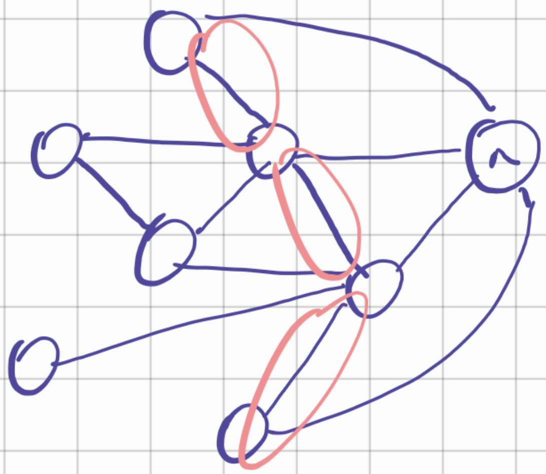
$$\kappa_c = \frac{1}{3}$$

↳ LOCAL CLUSTERING COEFFICIENT FOR NODE C IN THE GIVEN NETWORK IS 1/3. THIS MEANS THAT ONE-THIRD OF ALL THE POSSIBLE PAIRS OF CONNECTIONS TO C, WHICH COULD BECOME CONNECTED, ARE ACTUALLY CONNECTED.

REMARKS. IF YOU WERE TO GET THE LOCAL CLUSTERING COEFF OF J, YOU WOULD GET A MATHEMATICAL INCONSISTENCY. THEN ONE SETS:

$$\text{IF} \quad n_\lambda = 0 \rightarrow \kappa_\lambda \overset{!}{=} 0$$

QUICK EXAMPLE



$$\kappa_\lambda = \frac{2 \cdot m_\lambda}{d_\lambda(d_\lambda - 1)}$$

$$d_\lambda = 4; \; m_\lambda = 3$$

$$\therefore \kappa_\lambda = \frac{2 \cdot \cancel{3}}{4(\cancel{3})} = \frac{1}{2}$$

IT'S POSSIBLE TO GET THESE COEFFICIENTS IN NETWORK X.

GLOBAL CLUSTERING COEFF.

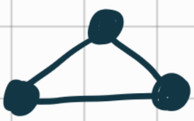
↳ ONE CAN TAKE THE AVERAGE LOCAL CLUSTERING COEFF. OF ALL NODES WITHIN THE NETWORK.

OR

→ COMPUTE THE TRANSITIVITY OF A NETWORK

WHAT IS TRANSITIVITY?

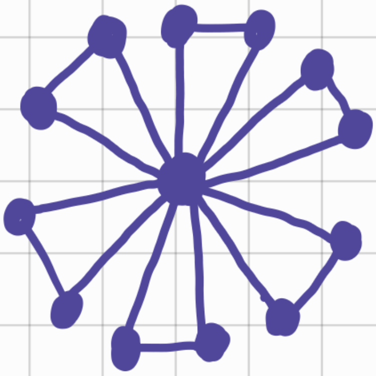
↳ PERCEIVE AS THE PERCENTAGE OF OPEN TRIADS THAT ARE TRIANGLES IN A NETWORK.

TRIANGLES 

OPEN TRIADS 

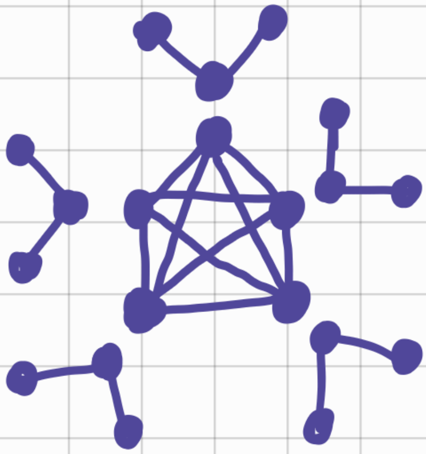
$$\text{TRANSITIVITY} = \frac{3 * \#\text{ OF CLOSED TRIADS}}{\#\text{ OF OPEN TRIADS}}$$

BOTH CONCEPTS CAN BE USED TO GAIN AN IDEA OF HOW THE GLOBAL CLUSTERING LOOKS LIKE. HOWEVER, HAVE A LOOK AT THE FOLLOWING TWO CASES.



MOST NODES HAVE HIGH LOCAL CLUSTERING COEFFICIENT BUT THE HIGH-DEGREE NODE HAS A LOW LOCAL CLUSTERING COEFFICIENT.

- AVE. CLUSTERING COEFF = 0.93
- TRANSITIVITY = 0.23



MOST NODES HAVE LOW LCC BUT THE HIGH DEGREE NODES HAVE HIGH LCC.

- AVE. CLUSTERING COEFF = 0.93
- TRANSITIVITY = 0.86

- THE AVERAGE LOCAL CLUSTERING COEFFICIENT OF ALL NODES IN NETWORKX
  ↳ nx.average_clustering(g)

- TRANSITIVITY IN NETWORKX
  ↳ nx.transitivity(g)

DISTANCE