Rensselaer

Introduction to Algorithms

Class 2: Graph Algorithm 1

Jianxi Gao

Department of Computer Science
Rensselaer Polytechnic Institute

www.gaojianxi.com

Big-O notation

We can categorize these functions (i.e., algorithms) into:

O(1)	constant time (not related to input size)
O(log n)	logarithmic time (e.g. binary search)
O(n)	linear time (e.g. finding an element in an unordered list)
O(n log n)	linearithmic time (e.g. quicksort average runtime)
O(n^2)	polynomial time (e.g. bubblesort)

Reasonable (in term of runtime)

Not reasonable (in term of runtime)

O(2^n)	Exponential time (OUCH!)
O($n!$)	Fractorial time (FORGET IT!)

we aim to use heuristics to GREATLY reduce the solution space
we may need to settle for near-optimal or near-correct solutions

Big-O common rules

1. Multiplicative constant can be omitted: $cn^2 = O(n^2)$

2. Given a>b, n^a **dominates** n^b

3. Any exponential **dominates** any polynomial

4. Any polynomial **dominates** any logarithm.

Big-Omega and Big-Theta(g)

1. f = O(g) "f grows no faster than g" f <= g

2. f = big-Omega(g) "f grows no slower than g" f >= g
   g = big-Omega(f) "g grows no slower than f" g >= f

3. f = big-Theta(g) "f grows at the same rate as g" f == g

4. if f = O(g) and g = O(f), then f = big-Theta(g)

   f<=g g<=f g==f

5. if f = O(g) and f = big-Omega(g), then f = big-Theta(g)

A test about Big-O, Big-Omega, and Big-Theta (NOT RELATED TO YOUR GRADING)

- Question 1:

  $f(n) = n - 100$ $g(n) = n - 200$ C

- Question 2:

  $f(n) = 100n + \log n$ $g(n) = n + \log^2 n$ C

- Question 3:

  $f(n) = n2^n$ $g(n) = 3^n$ A

- Question 4:

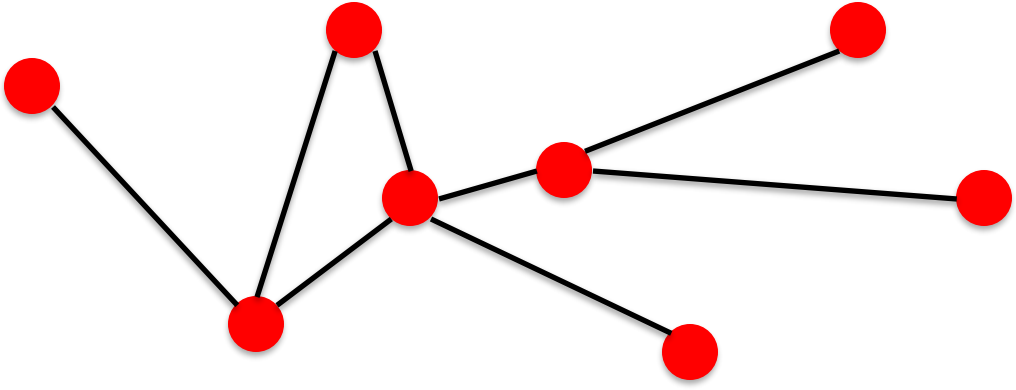
  $f(n) = \sqrt{n}$ $g(n) = \log^3 n$ B

- Question 5:

- $f(n) = n!$ $g(n) = 2^n$ B

1. Multiplicative constant can be omitted: $cn^2 = O(n^2)$
2. Given a>b, n^a dominates n^b
3. Any exponential dominates any polynomial
4. Any polynomial dominates any logarithm.
5. Factorial dominates exponential

Definition of Graphs



- **components**: nodes, vertices N
- **interactions**: links, edges L
- **system**: network, graph (N,L)

A **graph** is a mathematical structure for representing relationships.

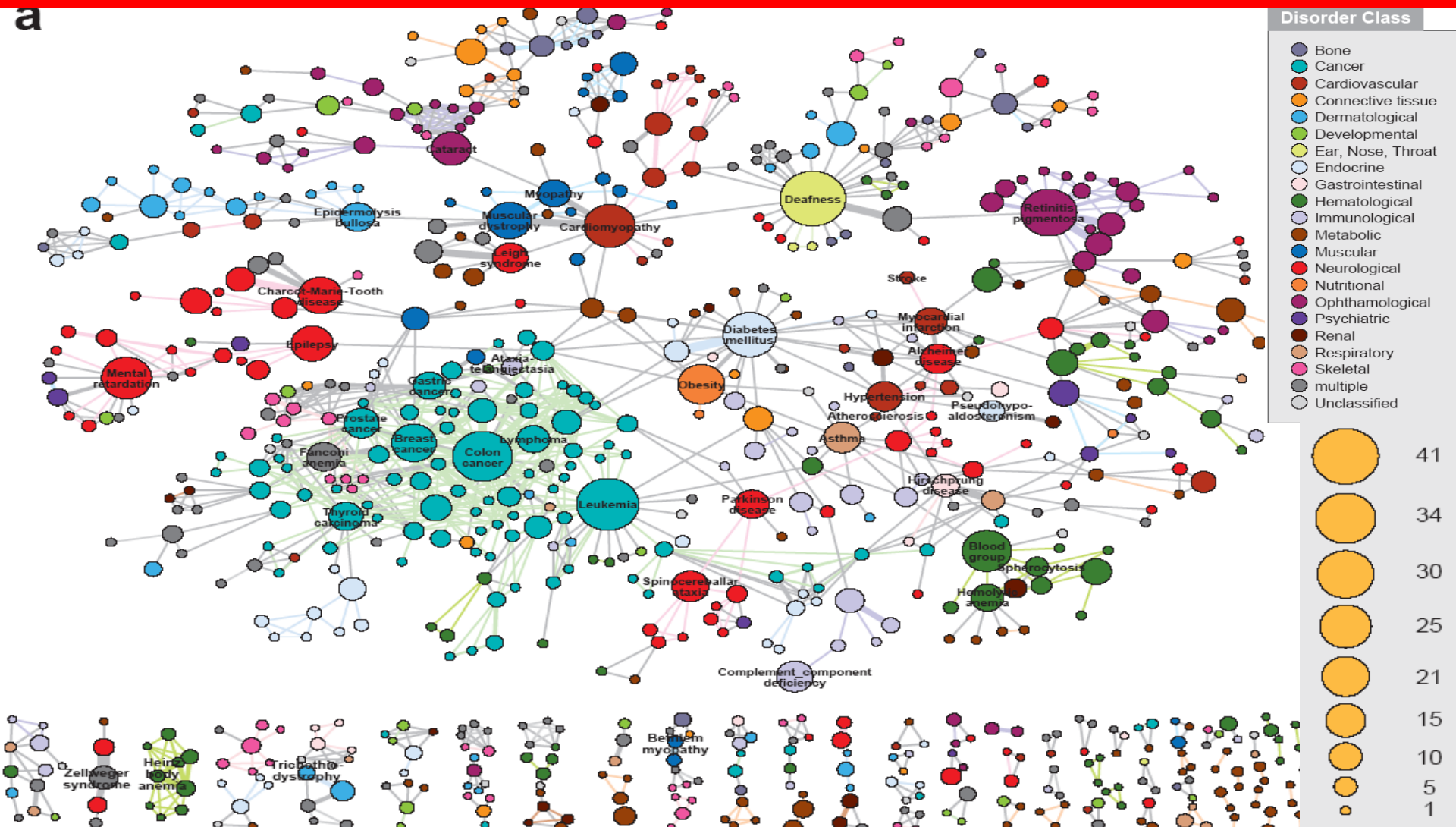

"I think the next century
will be the century
of complexity."
Stephen Hawking
January 23, 2000`

NETWORKS ARE AT THE HEART OF COMPLEX SYSTEMS

- Behind each complex system there is a **network**, that defines the interactions between the components.
- We will never understand complex system unless we map out and understand the networks behind them.

Examples of graphs

HUMAN DISEASE NETWORK



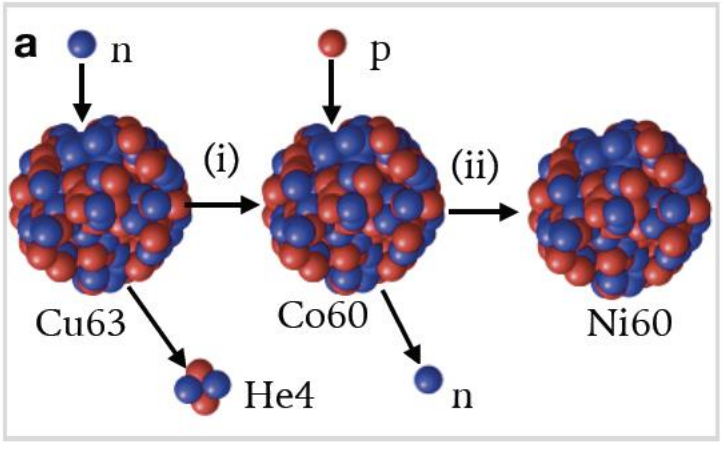
Examples of Networks: Facebook

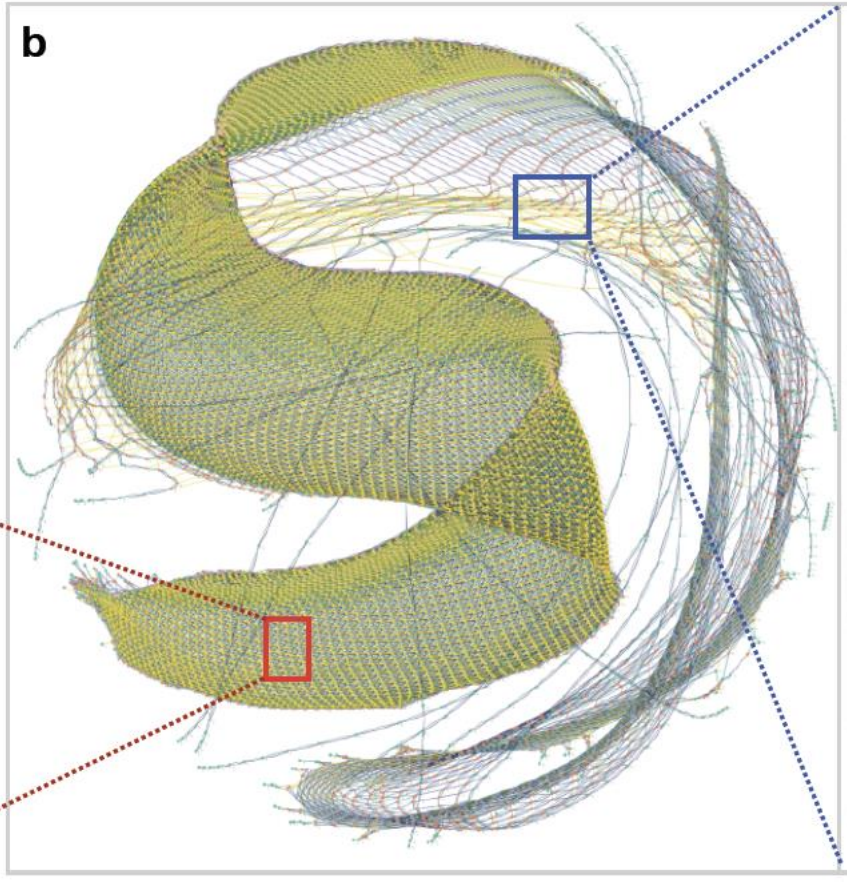
Fake News and True News

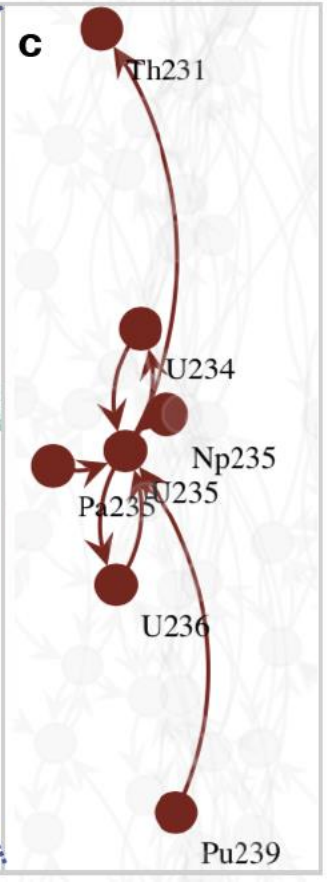
The “Social Graph” behind Facebook.

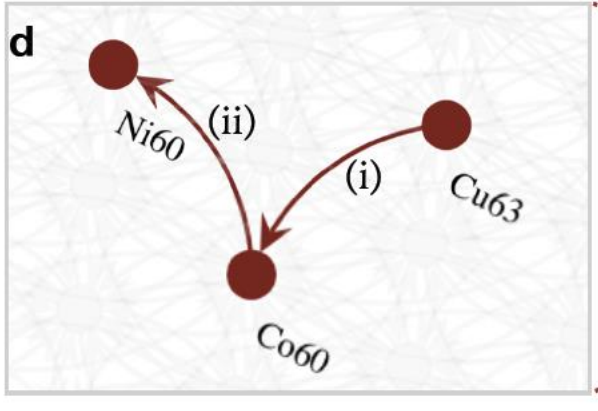
- Networks are at the heart of some of the most revolutionary technologies of the 21st century, empowering everything from Google to Facebook, CISCO, and Twitter.
- If you were to understand the searchability, **hopeless without invoking the Web’s topology.**

Examples of Networks: Nuclear Reaction Network

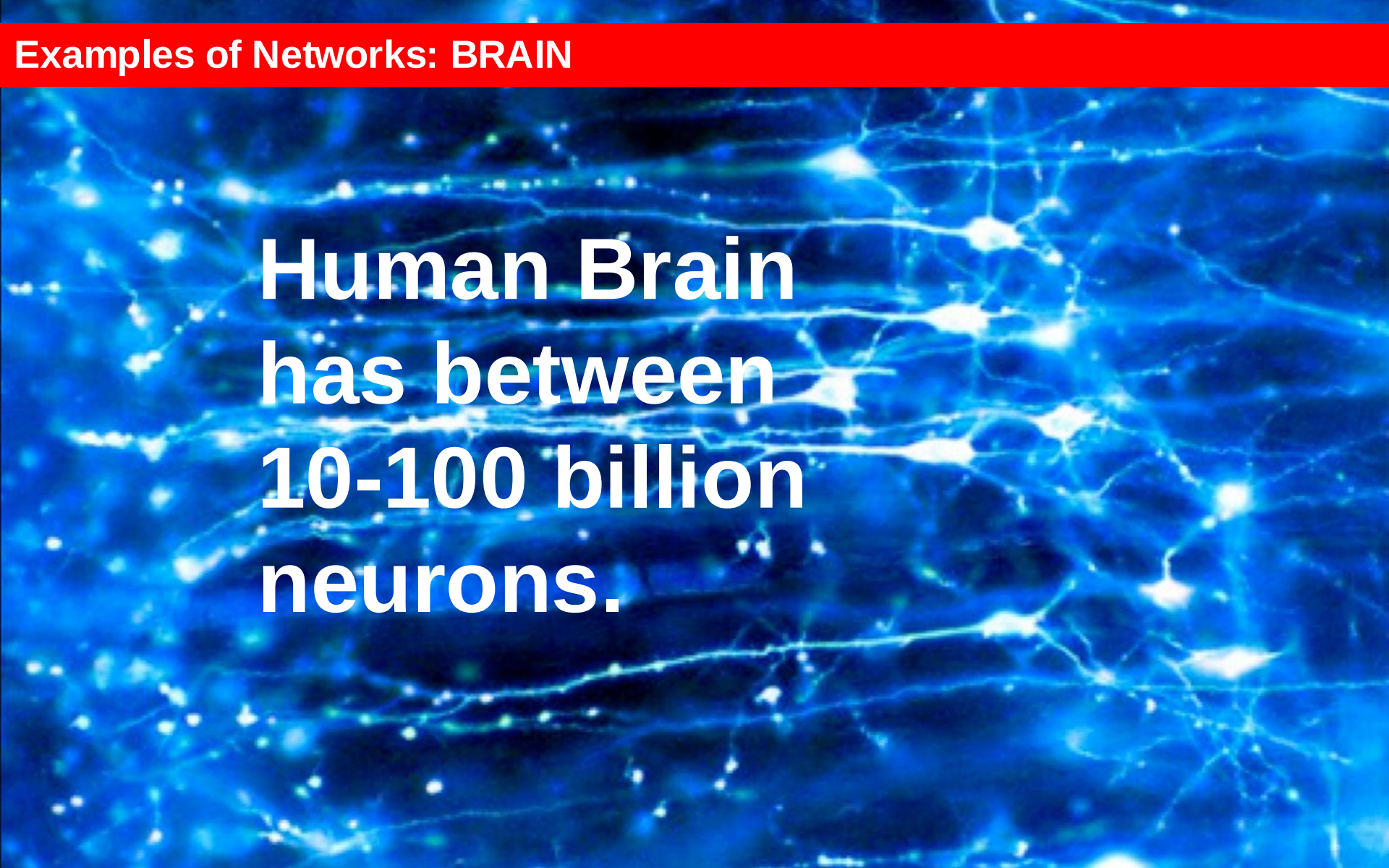




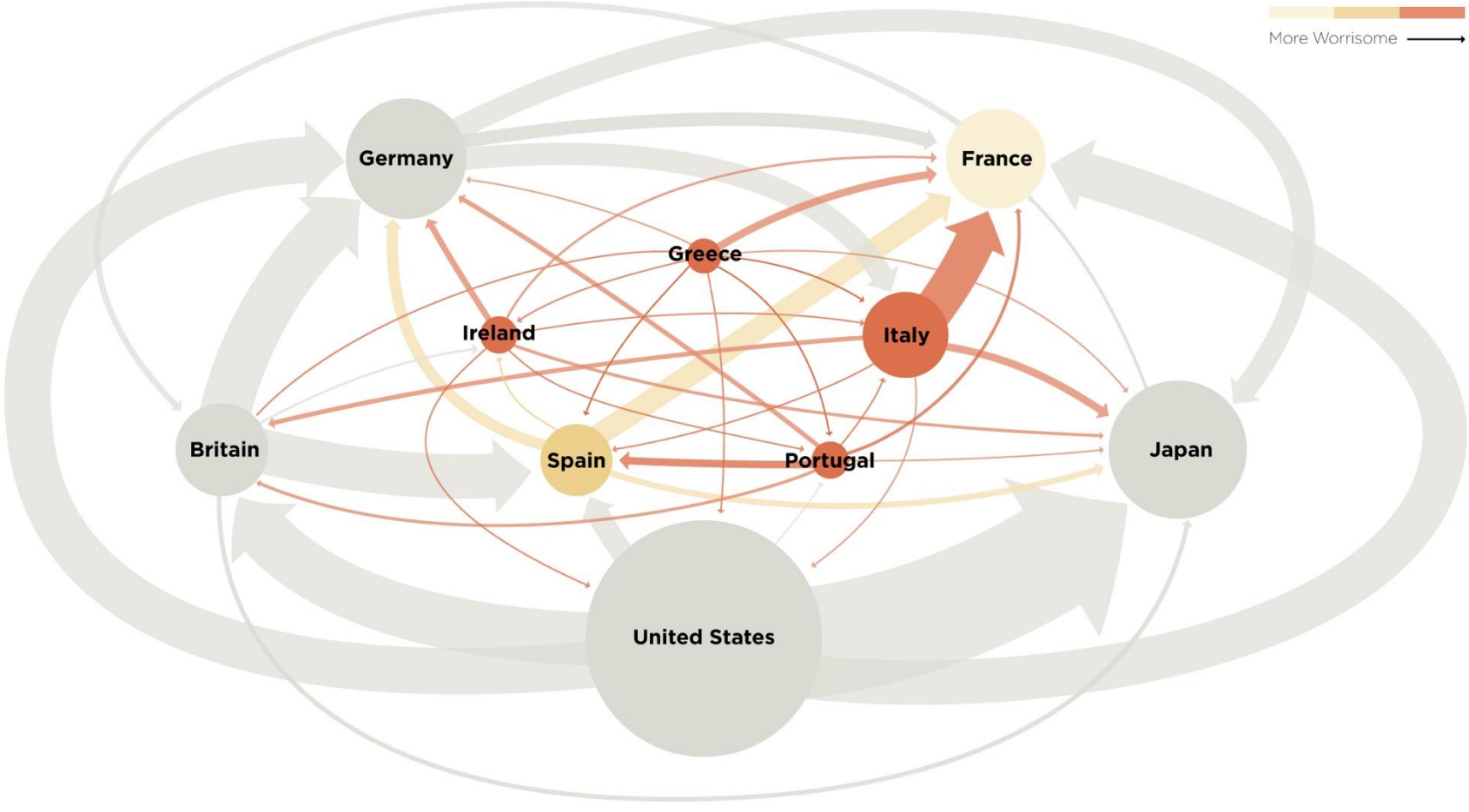




Examples of Networks: BRAIN



Examples of Networks: Financial network in 2011



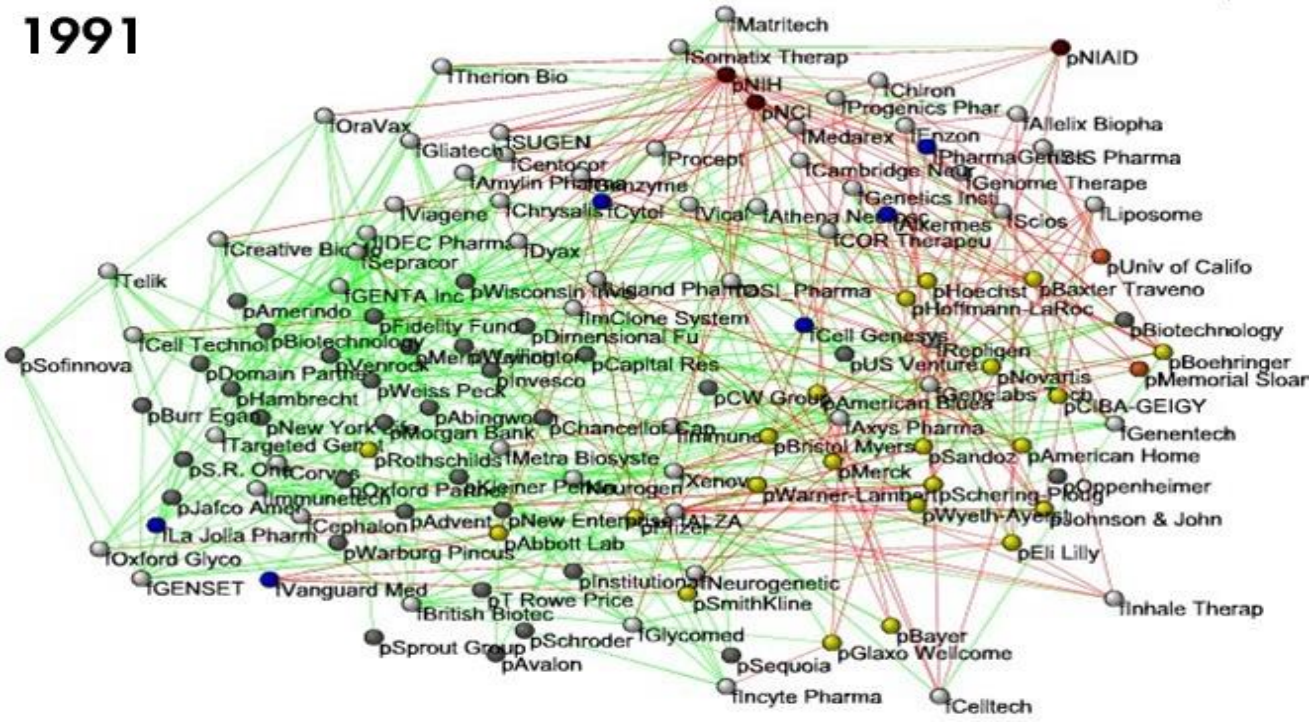
Examples of Networks: BUSINESS TIES IN US BIOTECH-INDUSTRY

Nodes:

- Companies
- Investment
- Pharma
- Research Labs
- Public
- Biotechnology

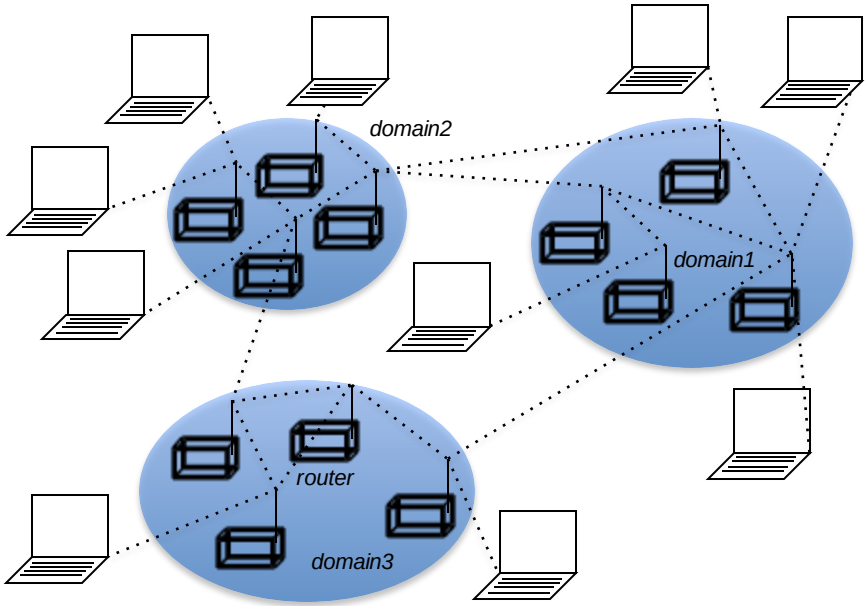
Links:

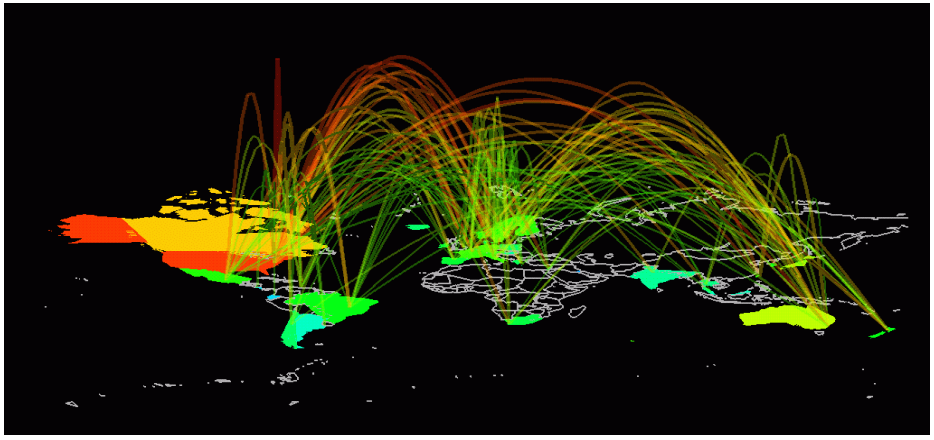
- Collaborations
- Financial
- R&D

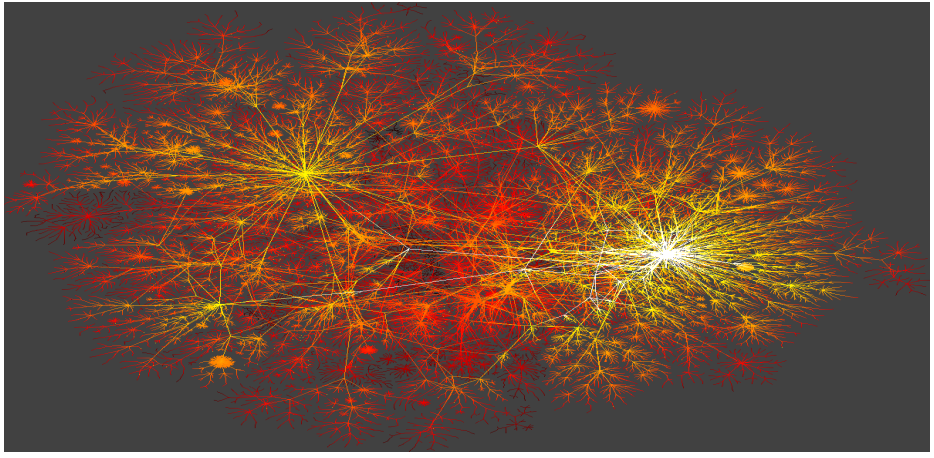


http://ecclectic.ss.uci.edu/~drwhite/Movie

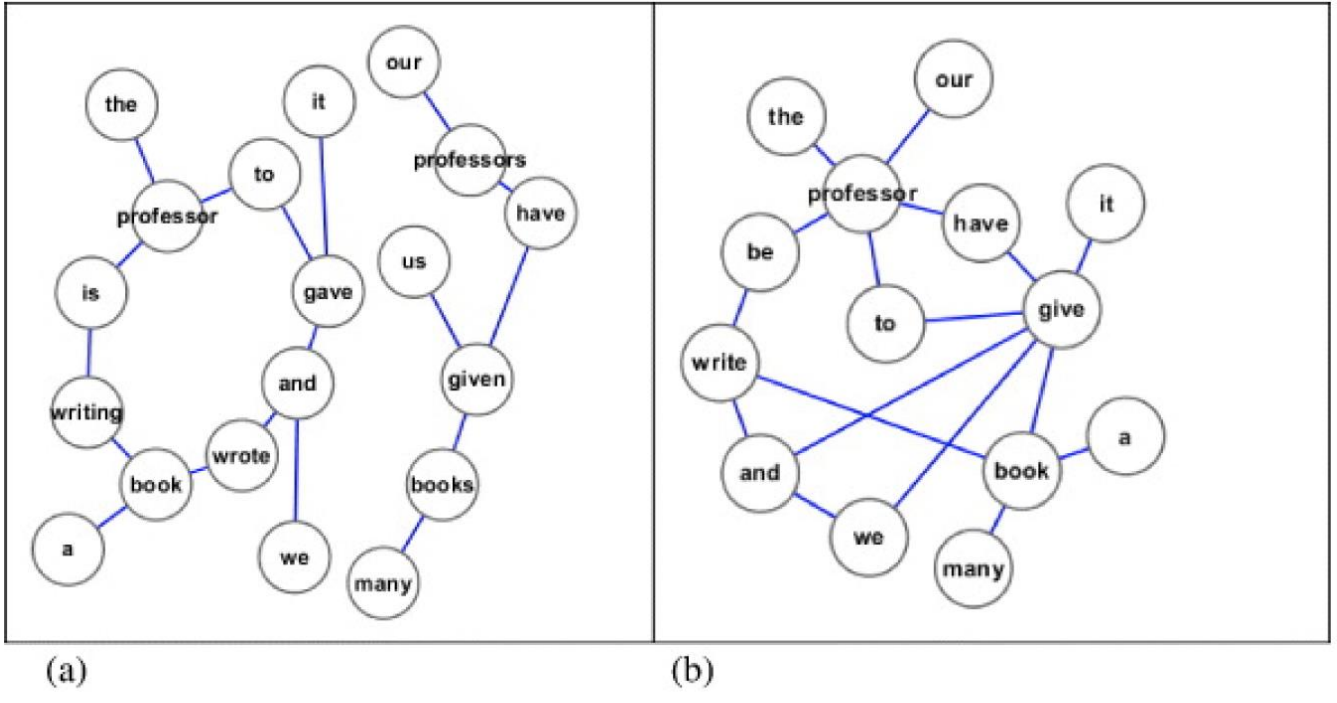
Examples of Networks: INTERNET







Examples of Networks: Network of words



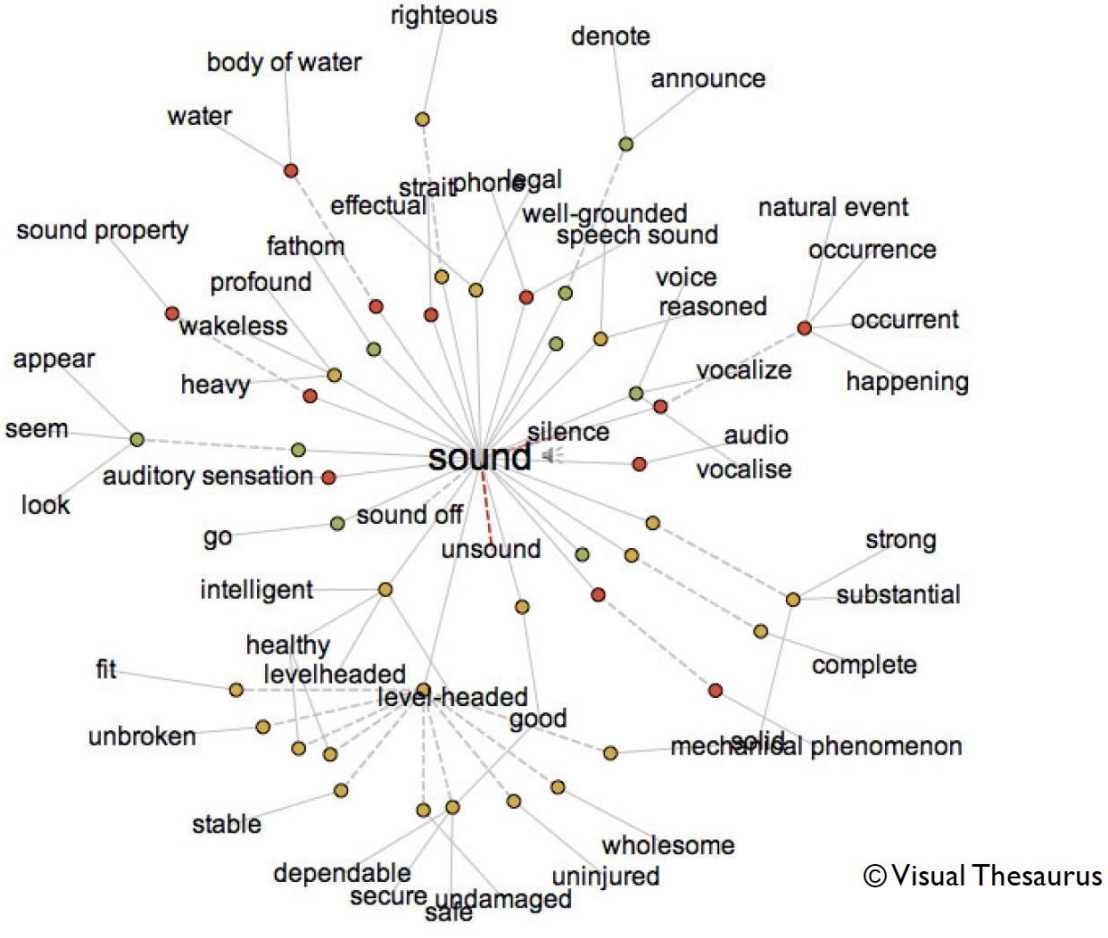
From Liu & Xu 2011; networks were generated from the following three sentences:

This professor is writing a book.

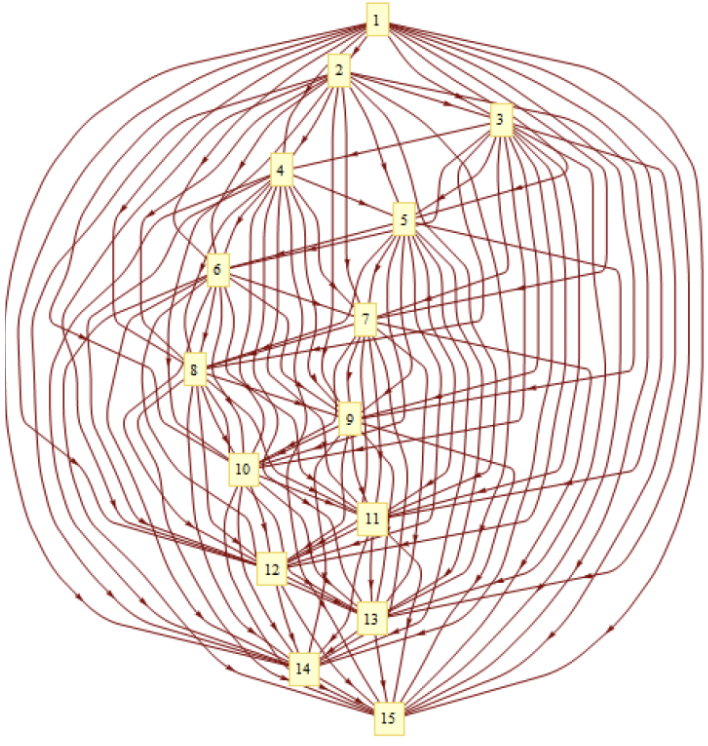
Our professors have given us many books.

We wrote a book and gave it to the professor.

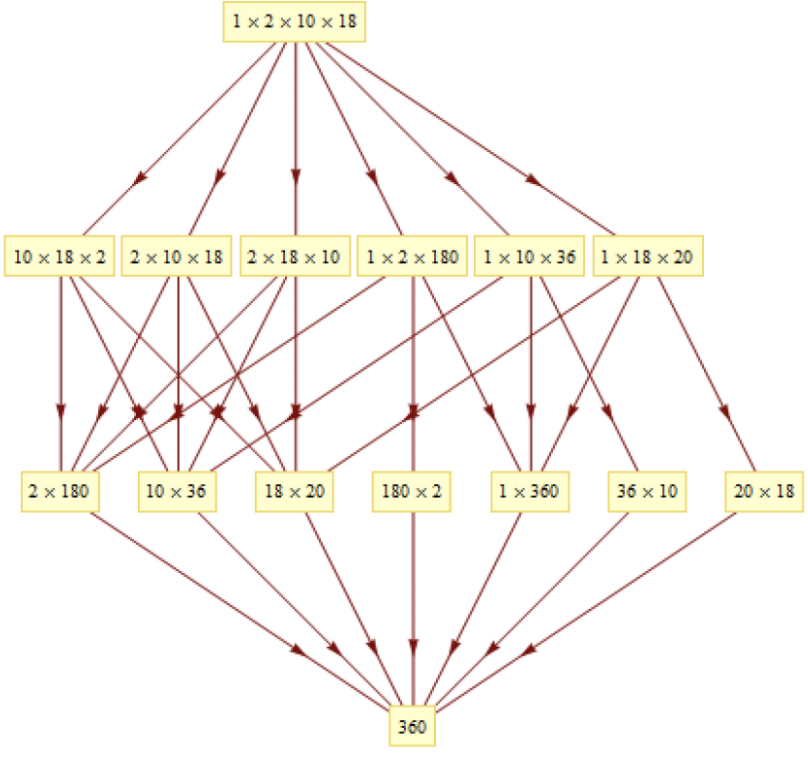
Examples of Networks: Network of words



Examples of Networks: Network of Numbers (1)

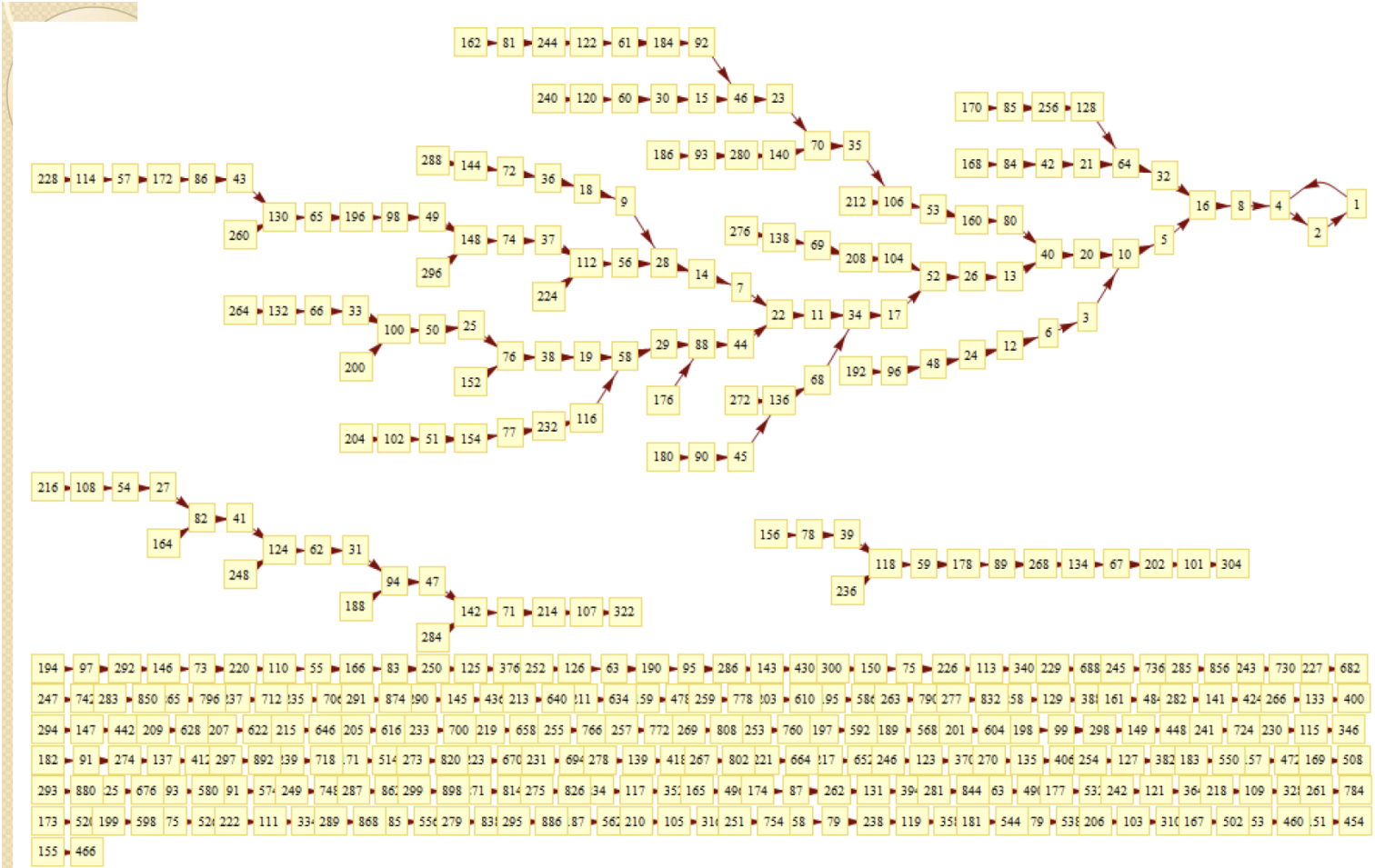


Transitivity network ($i \longrightarrow j$ if and only if $i < j$)



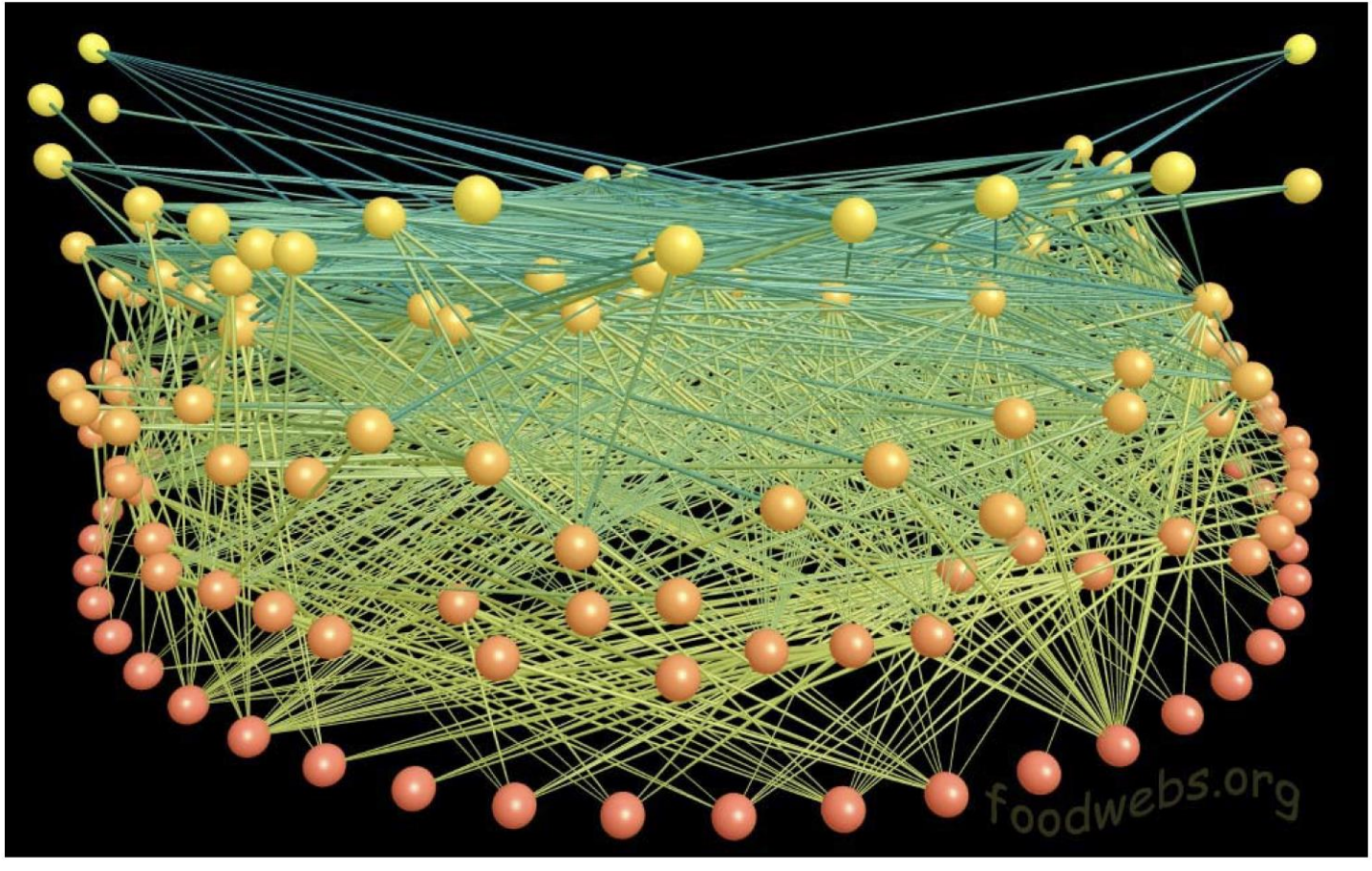
Associativity network (about multiplication)

Examples of Networks: Network of Numbers (2)



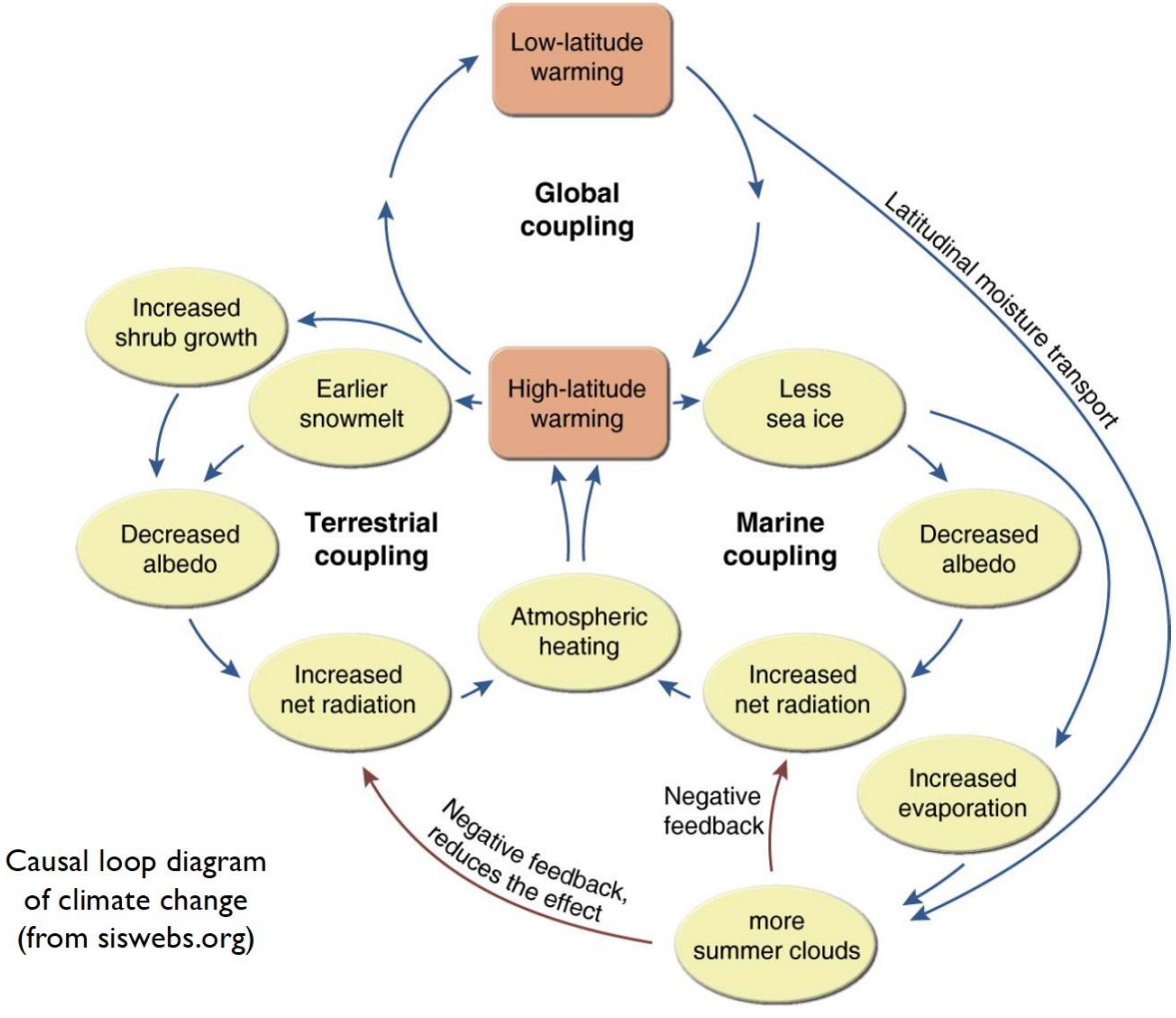
Collatz sequence ($x \rightarrow y$; $y = x/2$ if x is even, or $3x+1$ otherwise)

Examples of Networks: Food Webs



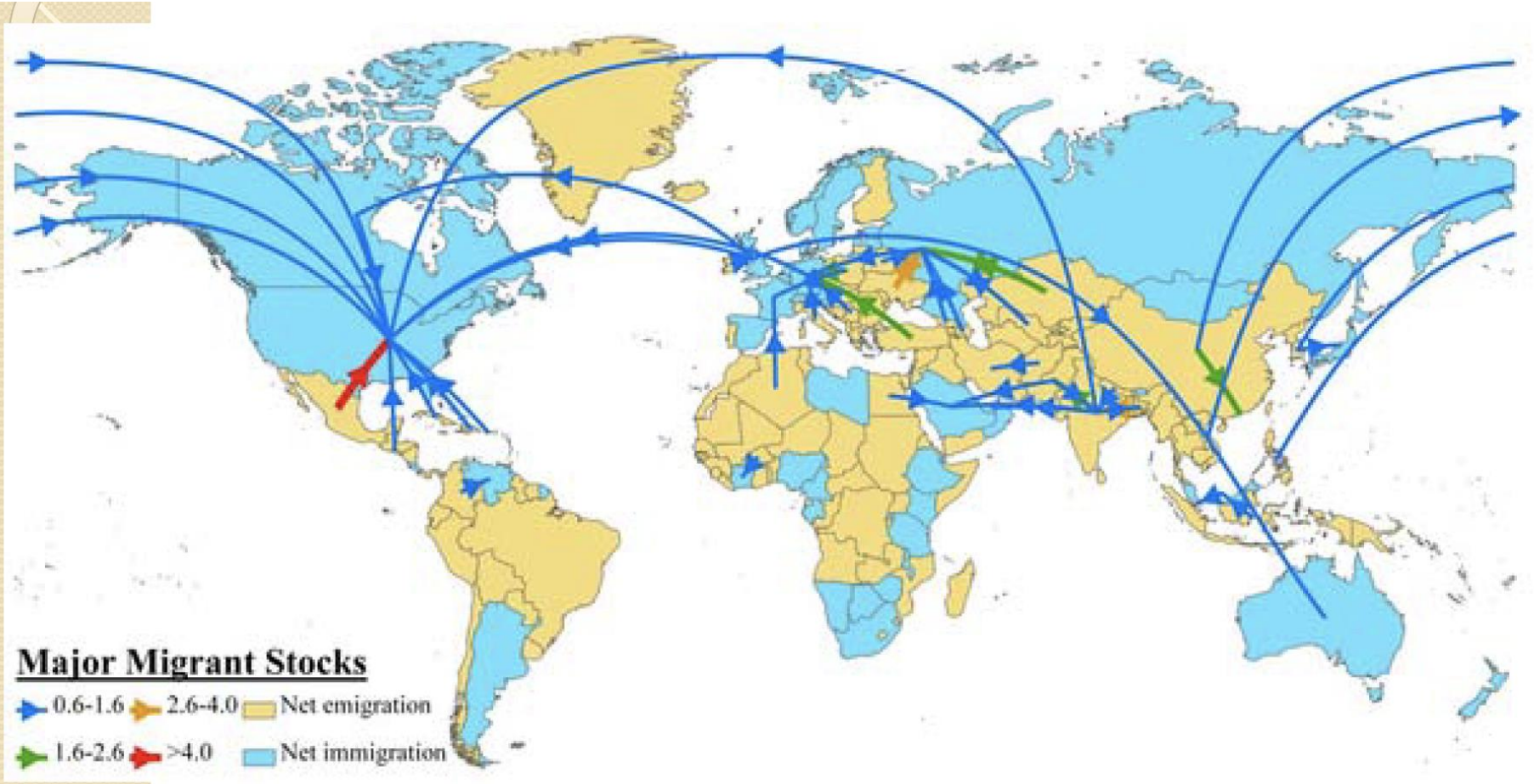
Food web in El Verde Rainforest, Puerto Rico by J. Dunne (from foodwebs.org)

Examples of Networks: Causal Loops



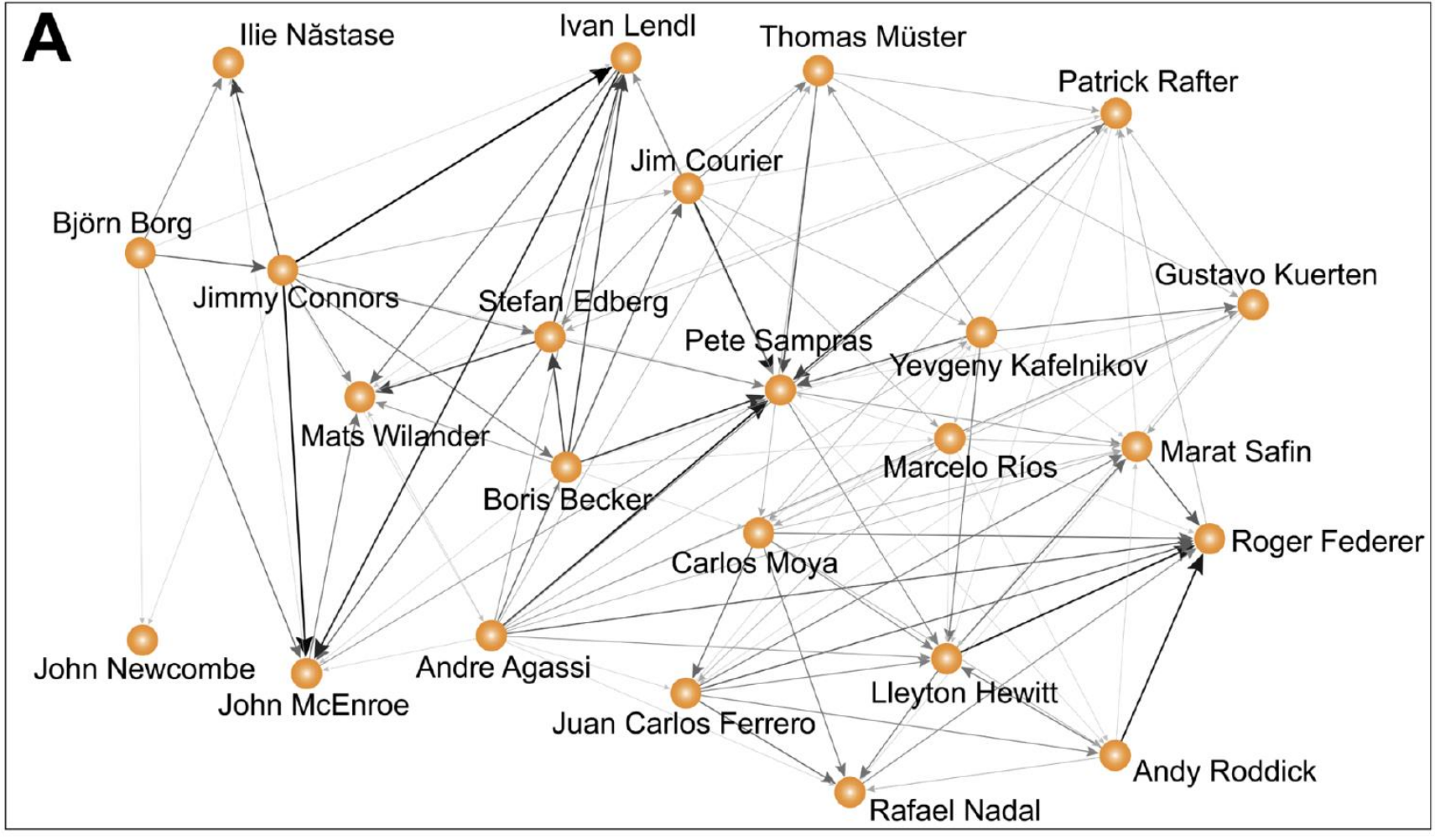
Causal loop diagram of climate change (from siswebs.org)

Examples of Networks: Network of Human Migration

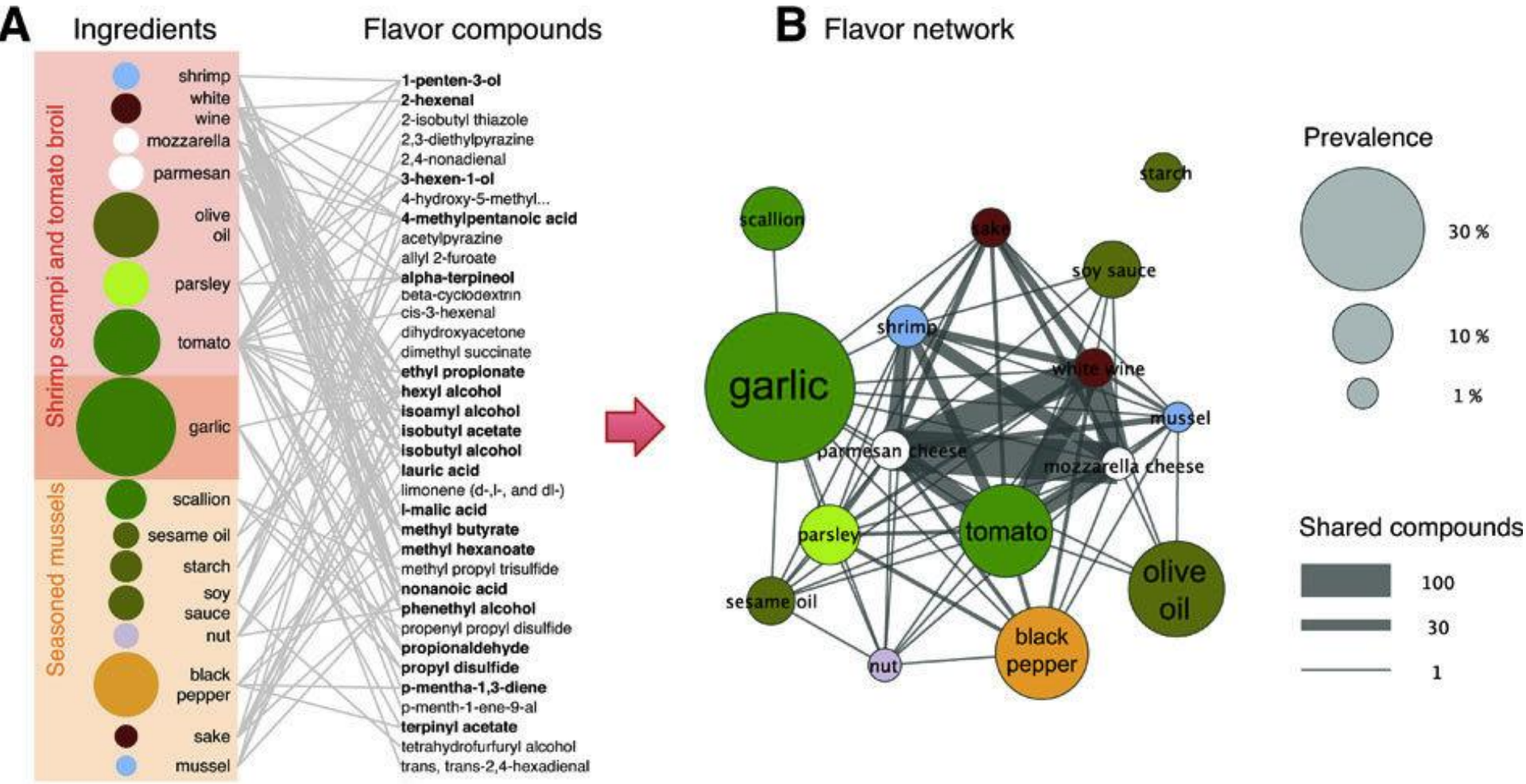


Davis et al., 2013

Examples of Networks: Network of Top Tennis players

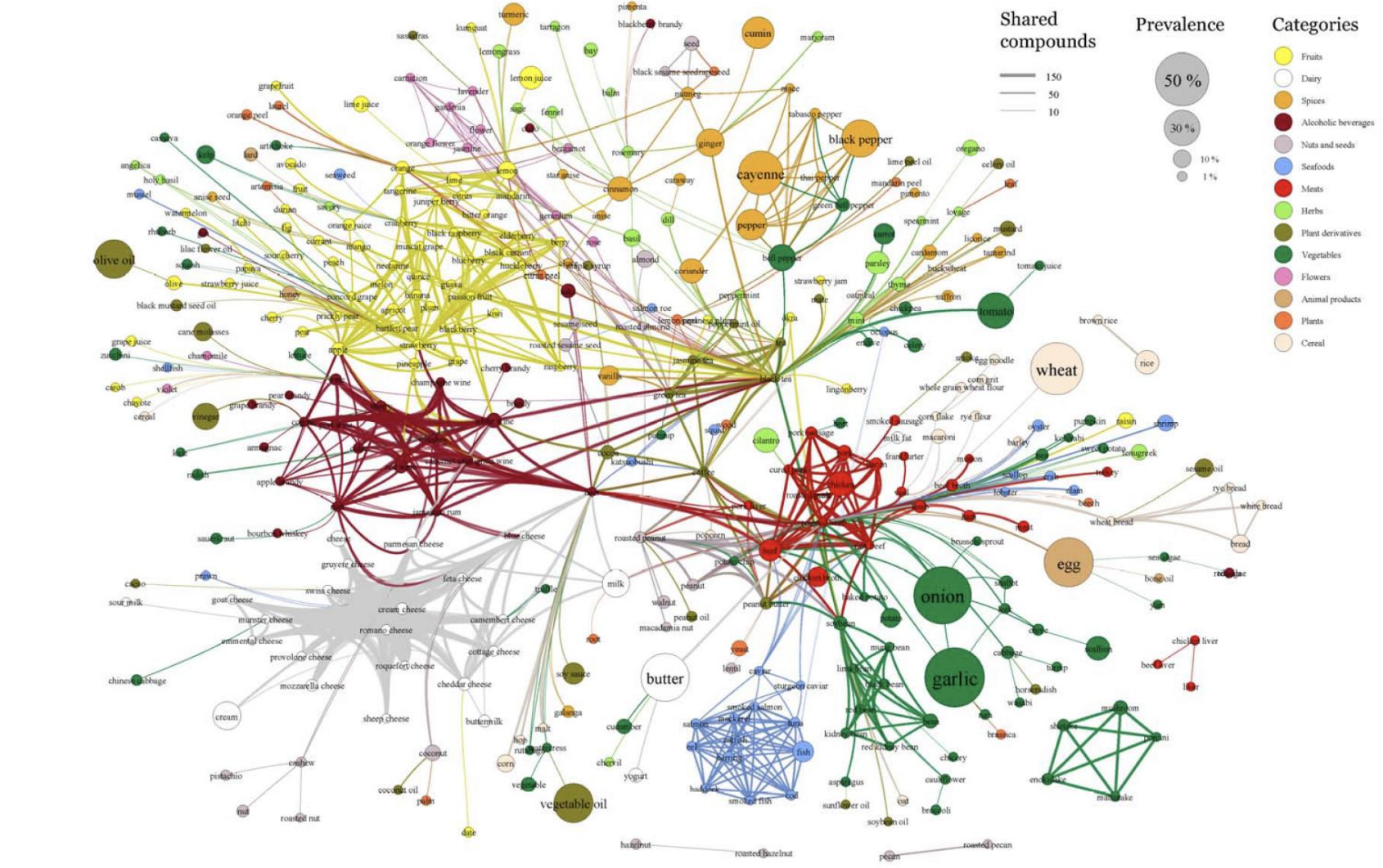


Ingredient-Flavor Bipartite Network

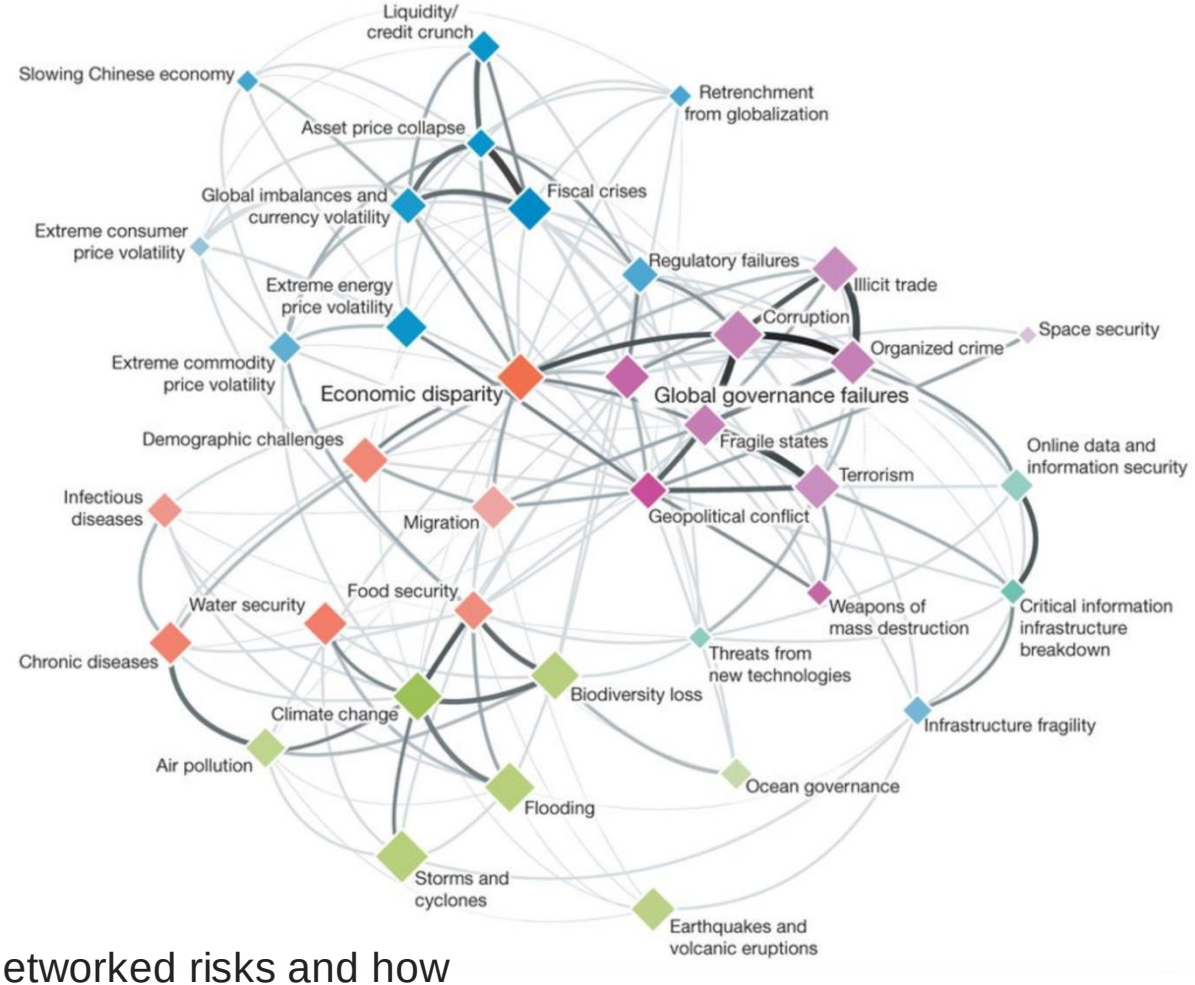


Y.-Y. Ahn, S. E. Ahnert, J. P. Bagrow, A.-L. Barabási [SEP] Flavor network and the principles of food pairing , Scientific Reports 196, (2011).

Ingredient-Flavor Bipartite Network

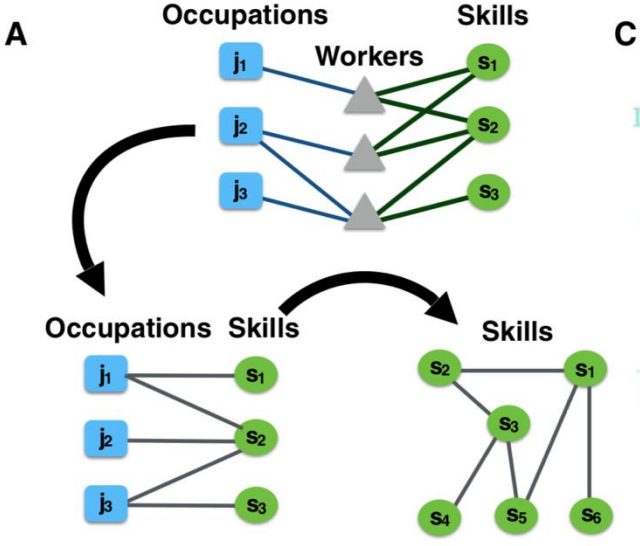


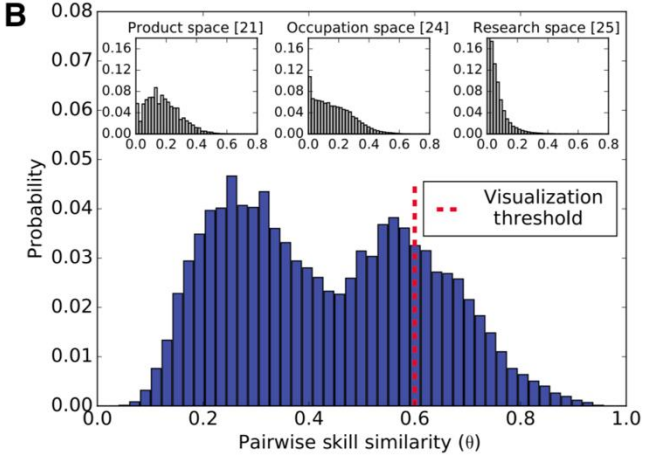
Global Risk Network

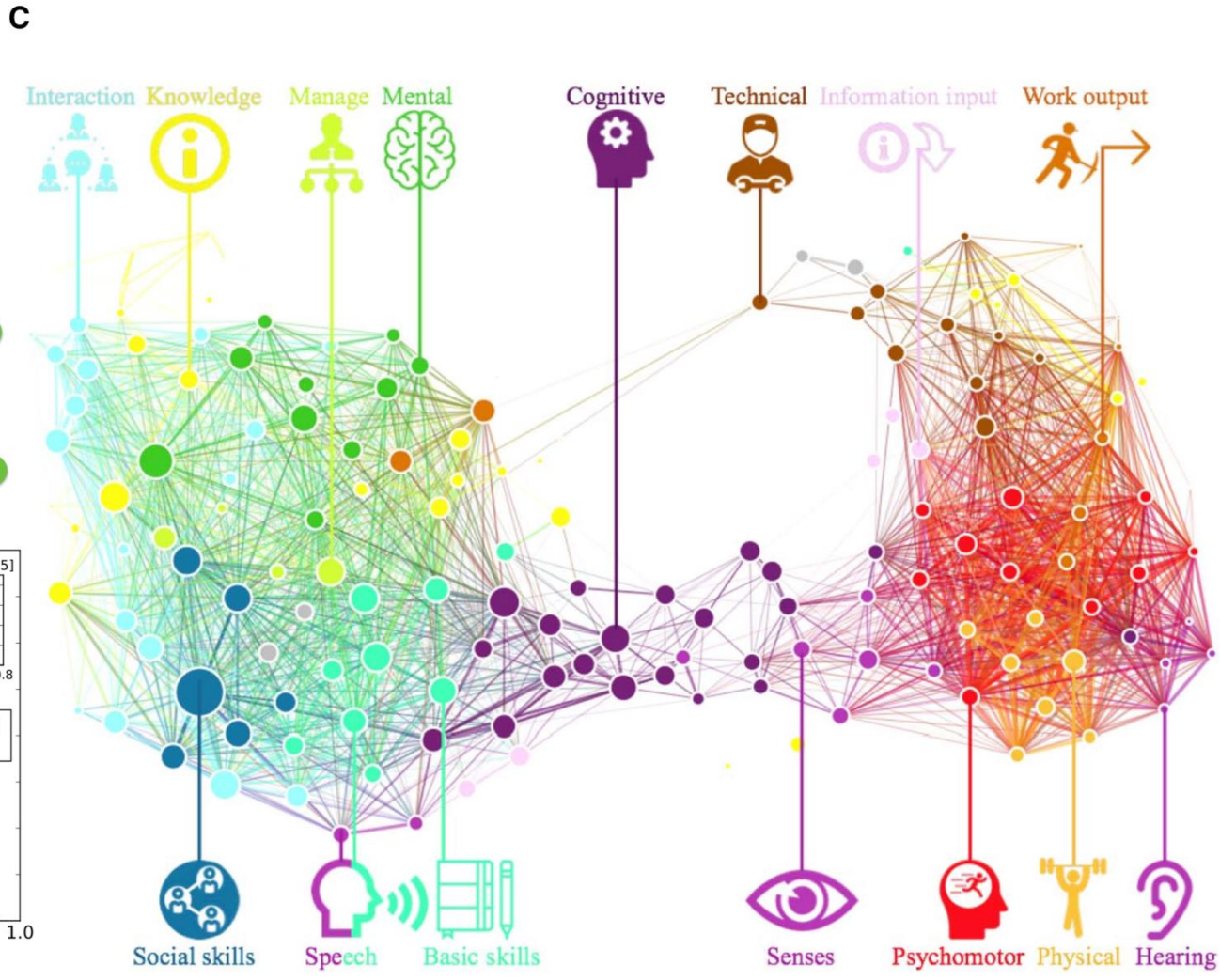


Helbing, Dirk. "Globally networked risks and how to respond." *Nature* 497.7447 (2013): 51.

Job graph


A
Occupations
Skills
Workers
j1
j2
j3
s1
s2
s3
Occupations
Skills
Skills
s4
s5
s6


B
Product space [21]
Occupation space [24]
Research space [25]
Probability
Visualization threshold
Pairwise skill similarity (θ)

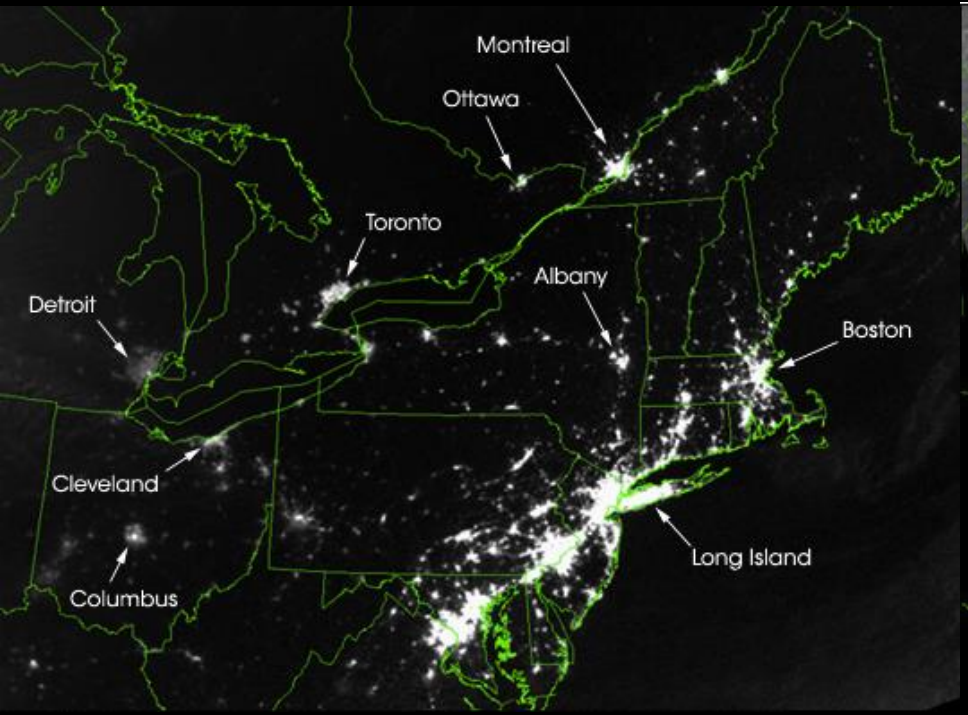

C
Interaction
Knowledge
Manage
Mental
Cognitive
Technical
Information input
Work output
Social skills
Speech
Basic skills
Senses
Psychomotor
Physical
Hearing

Take-Home message

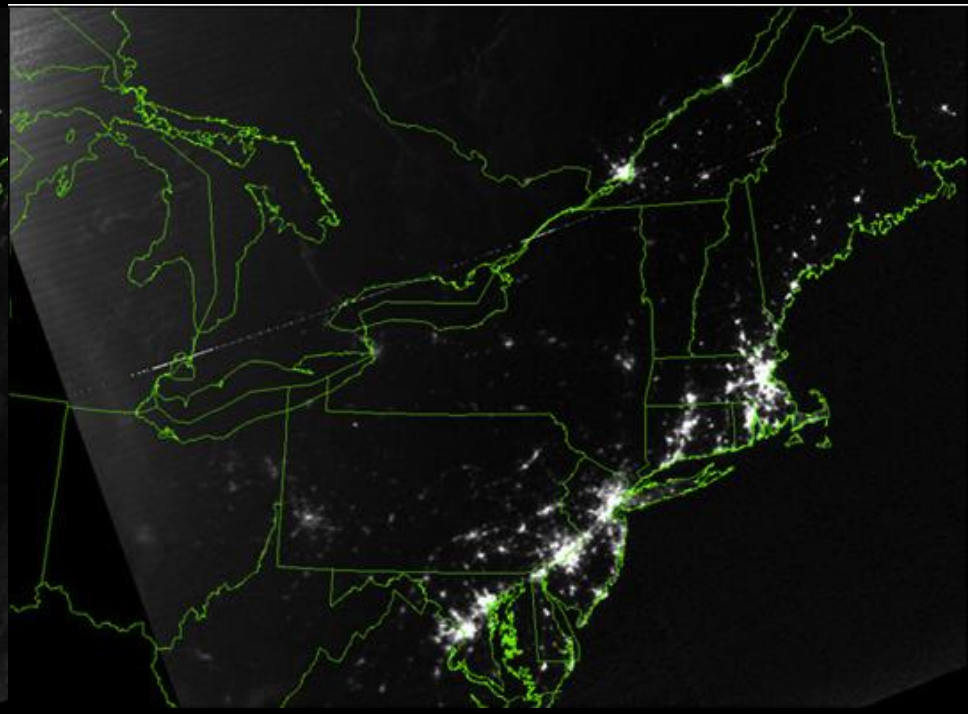
- Anything can be understood as a Graph if you pay attention to “connections” between the components.
- Can you give some examples of networks I did not mention?

Importance of graphs

A SIMPLE STORY (2): August 15, 2003 blackout.



August 14, 2003: 9:29pm EDT
20 hours before



August 15, 2003: 9:14pm EDT
7 hours after

A SIMPLE STORY (2): August 15, 2003 blackout.

An important theme of this class:

→ we must understand how network structure affects the robustness of a complex system.

→ develop quantitative tools to assess the interplay between network structure and the dynamical processes on the networks, and their impact on failures.

→We will learn that failures reality failures follow reproducible laws, that can be quantified and even predicted using the tools of network science.

MOST IMPORTANT | Networks Really Matter

If you were to understand the spread of diseases, **can you do it without networks?**

If you were to understand the WWW structure, searchability, etc, **hopeless without invoking the Web's topology.**

If you want to understand human diseases, **it is hopeless without considering the wiring diagram of the cell**.

Network analysis and visualization tools

Gephi

Cytoscape

NetworkX

Infomap

Igraph

Statnet

Network Workbench

Pajek network visualization (Windows)

Jung network analysis

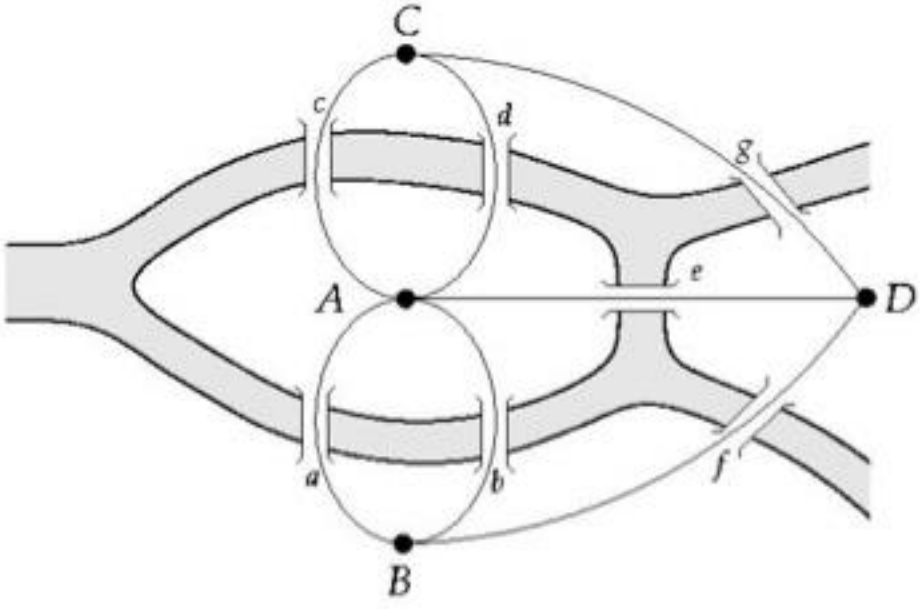
GraphViz

Matlab's Networks toolbox

THE BRIDGES OF KONIGSBERG

Can one walk across the seven bridges and never cross the same bridge twice?

THE BRIDGES OF KONIGSBERG



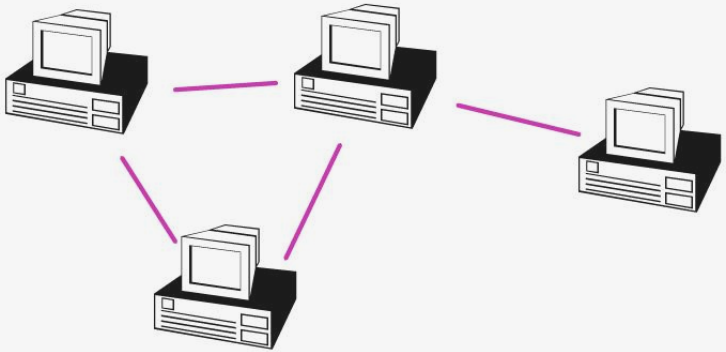
Can one walk across the seven bridges and never cross the same bridge twice?

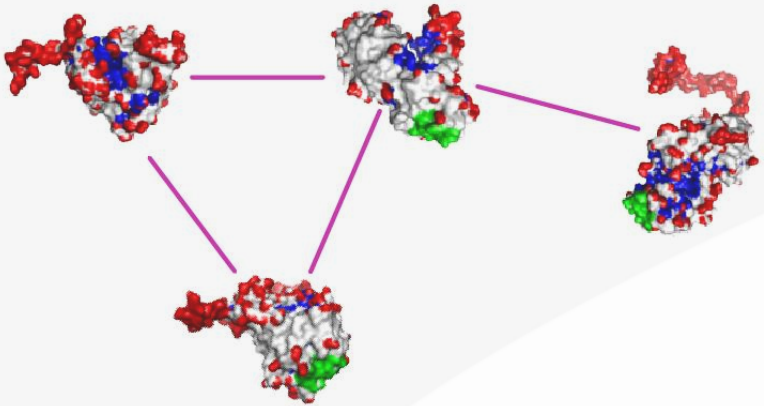
1735: **Euler's theorem:**

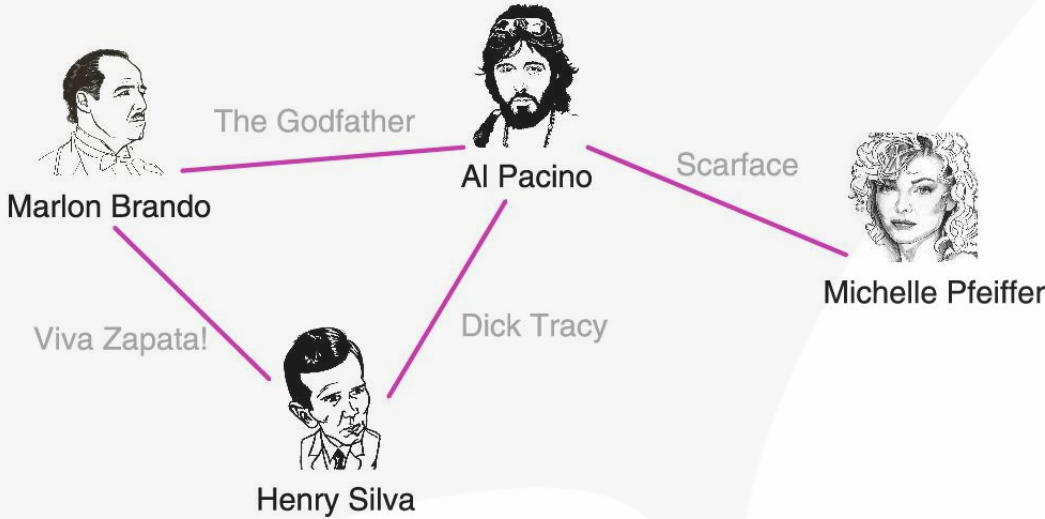
(a) If a graph has more than two nodes of odd degree, there is no path.

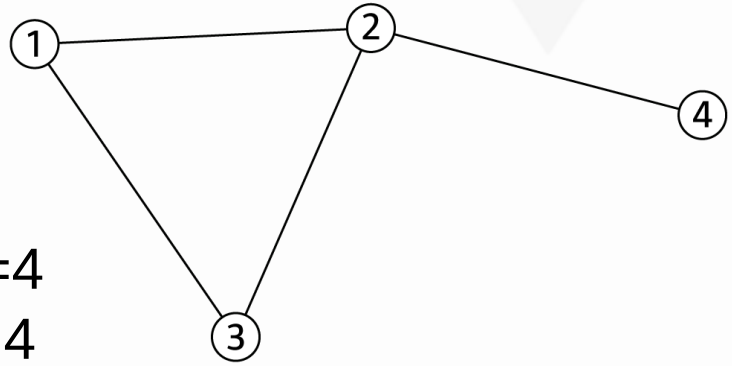
(b) If a graph is connected and has no odd degree nodes, it has at least one path.

A COMMON LANGUAGE









N=4
L=4

Algorithms

UNDIRECTED NETWORKS

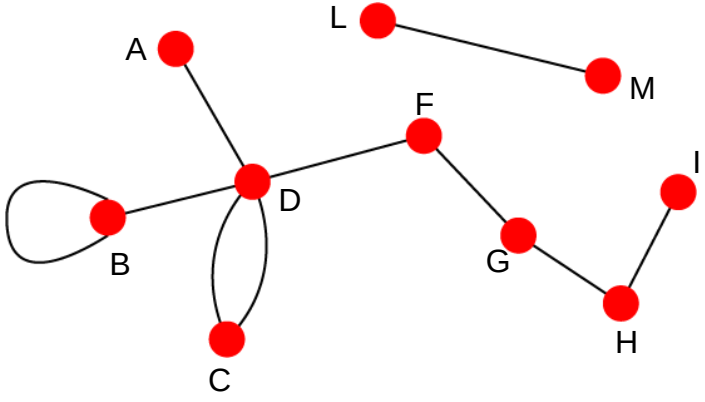
Notation. $G = (V, E)$

V = nodes (or vertices).

E = edges (or arcs) between pairs of nodes

Captures pairwise relationship between objects.

Network size parameters: $n = |V|$, $m = |E|$.



V = {A,B,C,D,E,F,G,H,I, L,M}

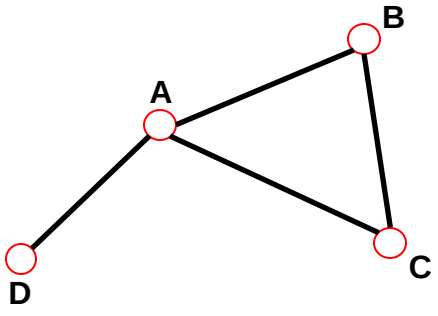
E {A-D,B-B,B-D,C-D,C-D,D-F,F-G,G-H,H-I,L-M}

A-D is the same as D-A

n = 10, m = 10.

DISTANCE IN A GRAPH

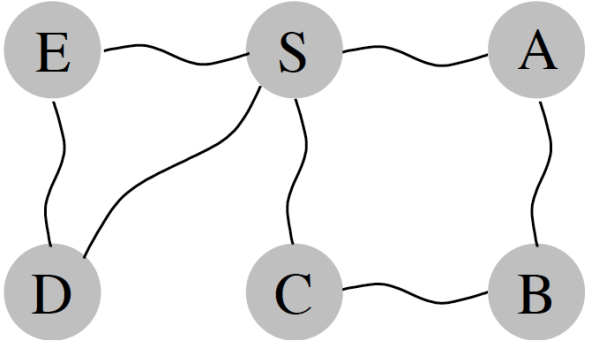
Shortest Path

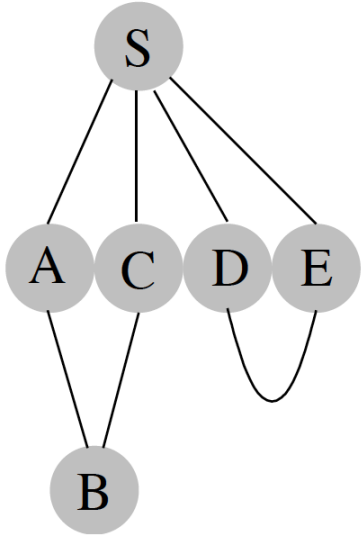


The *distance (shortest path, geodesic path)* between two nodes is defined as the number of edges along the shortest path connecting them.

*If the two nodes are disconnected, the distance is infinity.

What are the distance from S to other nodes?





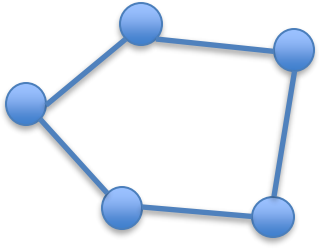
What are the distance from D to other nodes?

DISTANCE IN A GRAPH

Shortest Path

For a given graph $N = |V|$, how many edges can a graph have?

Path



Cycle

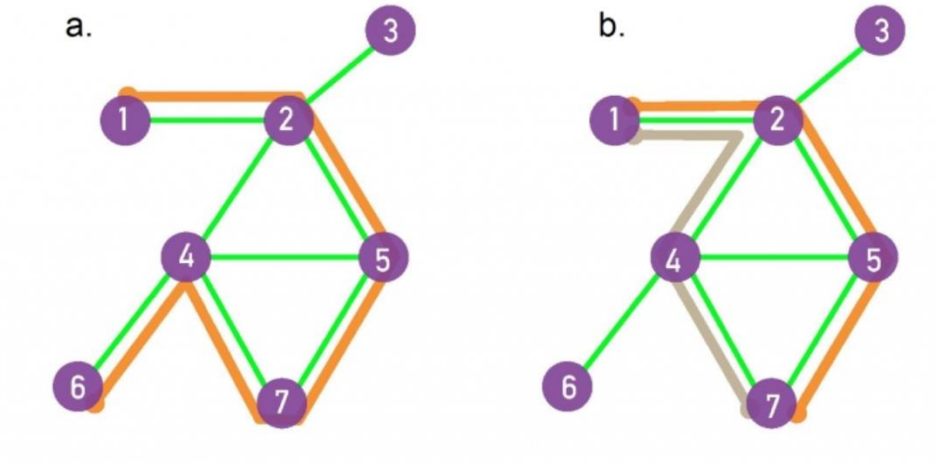
A (simple) path is a graph with not cycles.

PATHS

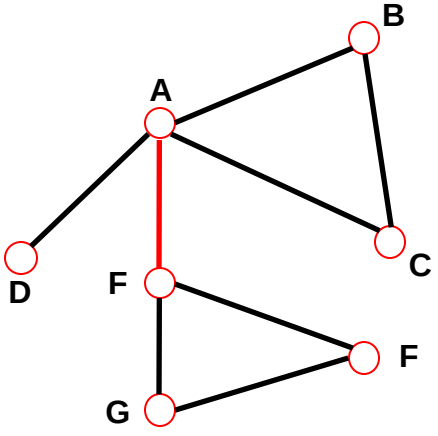
A *path is* a sequence of nodes in which each node is adjacent to the next one.

$P_{i0,in}$ of length *n* between nodes i_0 and i_n is an ordered collection of *n+1* nodes and *n* links

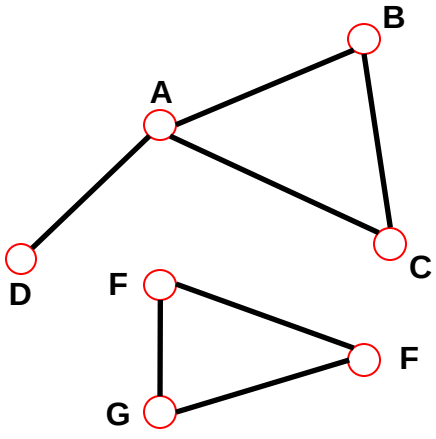
$$P_n = \{(i_0, i_1), (i_1, i_2), (i_2, i_3), \ldots, (i_{n-1}, i_n)\}$$



Connected graph



Connected



Not connected

Telephone
Internet
transportation

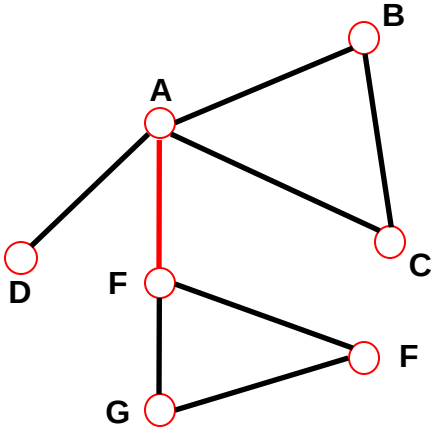
Bridge: A-F

Def 1: Nodes *u* and *v* are connected, if ∃ a path connection *u* and *v*.

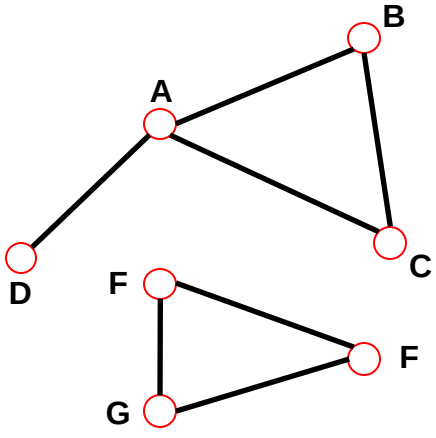
Def 2: A graph is connected if all nodes are connected to each other.

A disconnected graph is made up by two or more connected components.

Connected graph



Connected



Not connected

Fundamental questions: Given Graph G=(V,E), nodes s and t:

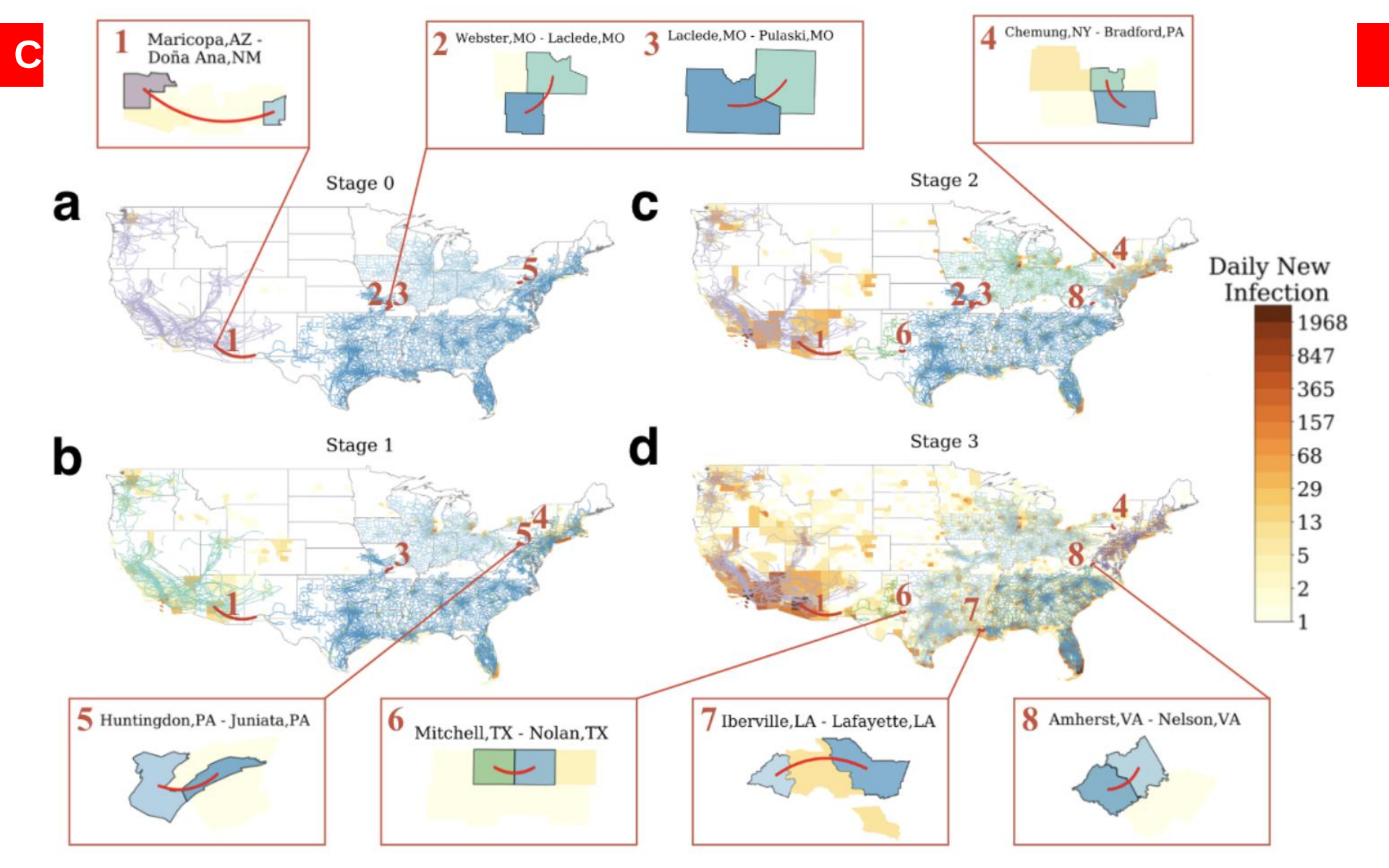
- Is there a path from s to t?
- How to find the shortest path from s to t?

In fact, we will learn an efficient algorithm:

- Find all nodes s is connected to (within the connected component).
- Find shortest paths from s to all the other nodes. →find distance from s to all other nodes.

Question: If I find the distance, should I know all the paths?

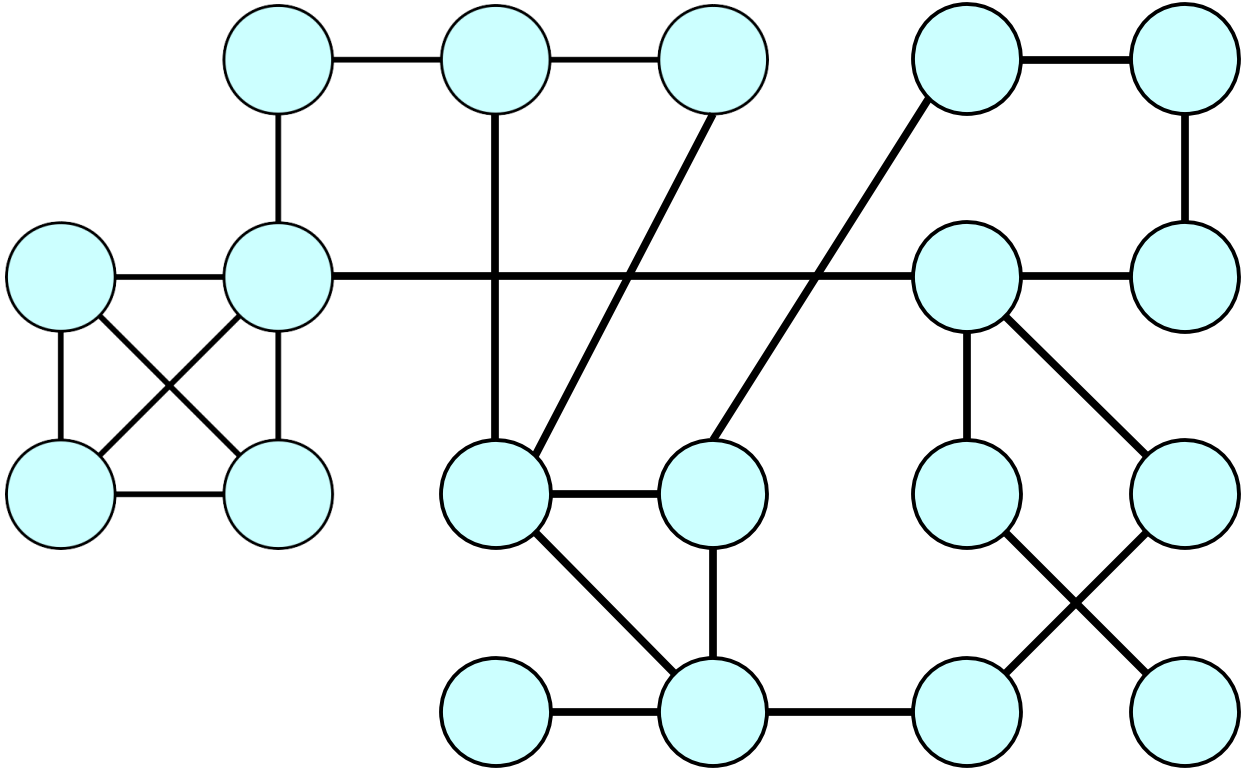
Connected graph


1 Maricopa,AZ - Doña Ana,NM
2 Webster,MO - Laclede,MO
3 Laclede,MO - Pulaski,MO
4 Chemung,NY - Bradford,PA
a
Stage 0
b
Stage 1
c
Stage 2
d
Stage 3
Daily New Infection
1968
847
365
157
68
29
13
5
2
1
5 Huntingdon,PA - Juniata,PA
6 Mitchell,TX - Nolan,TX
7 Iberville,LA - Lafayette,LA
8 Amherst,VA - Nelson,VA

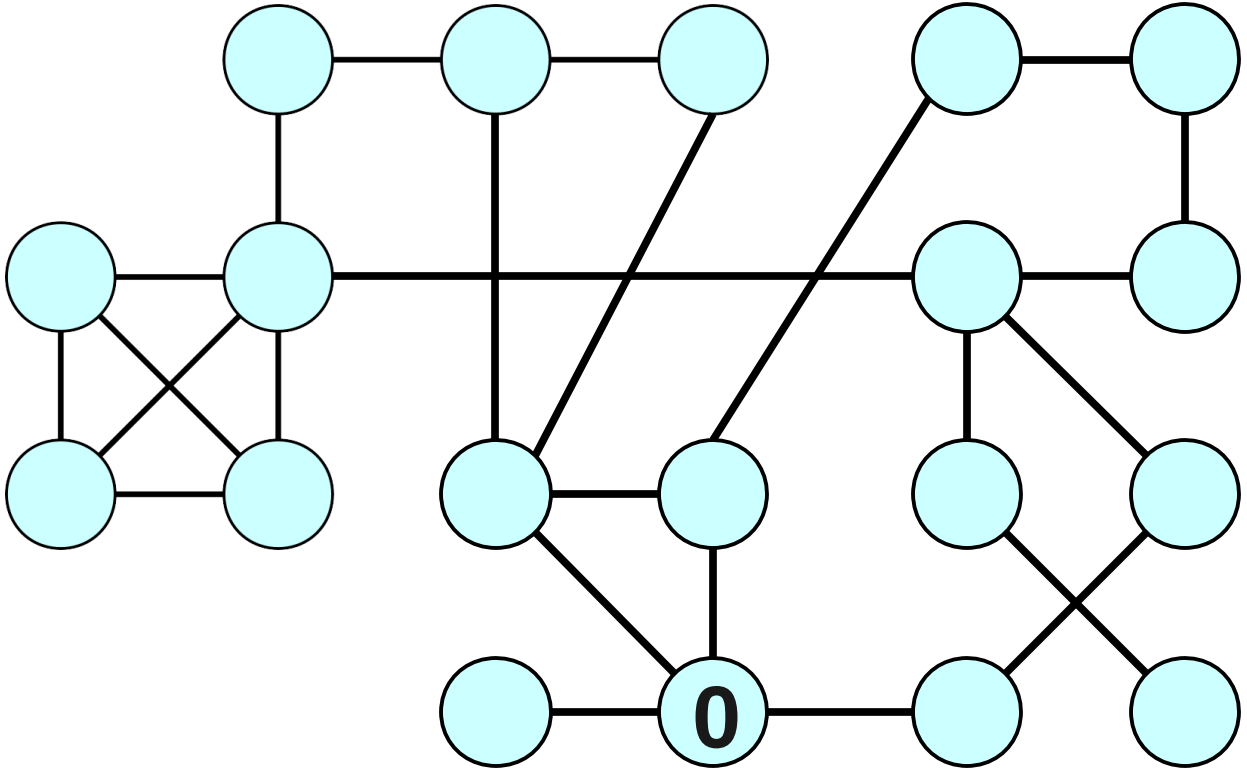
Six Degrees of Separation

Six Degree of Separation

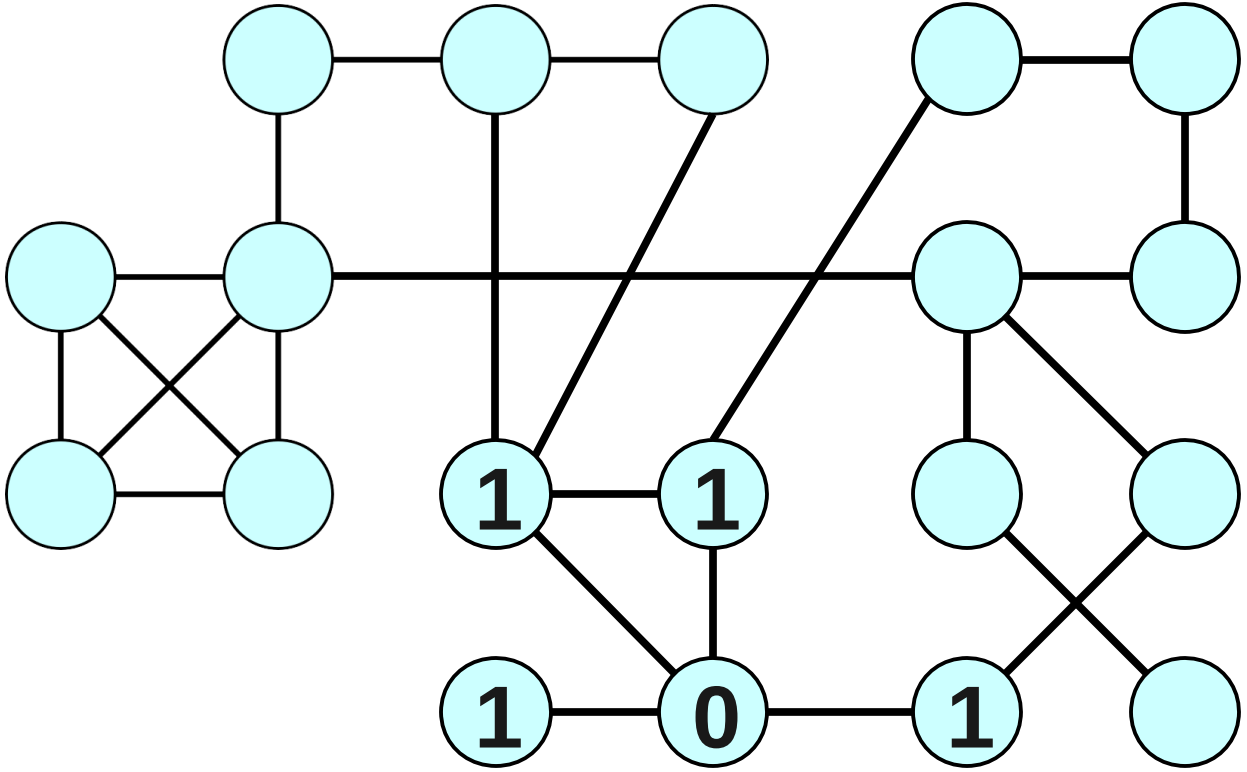
A Social Network



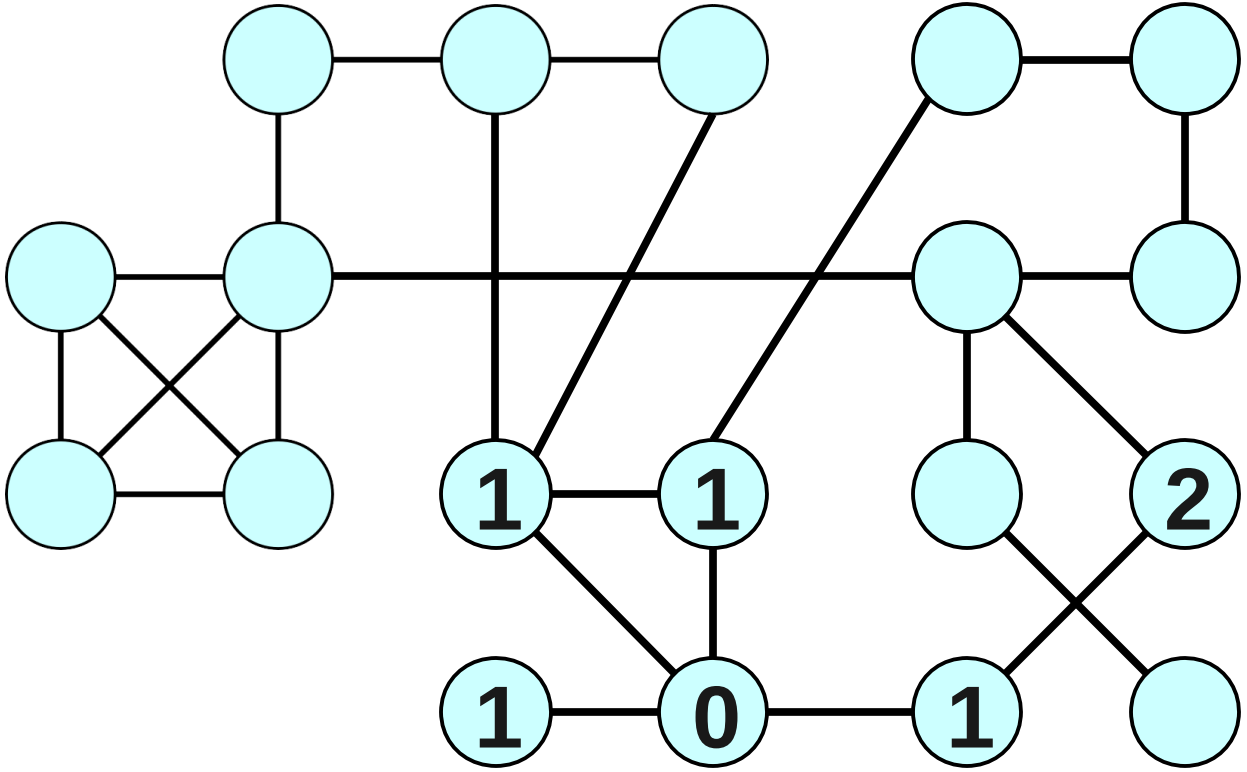
A Social Network



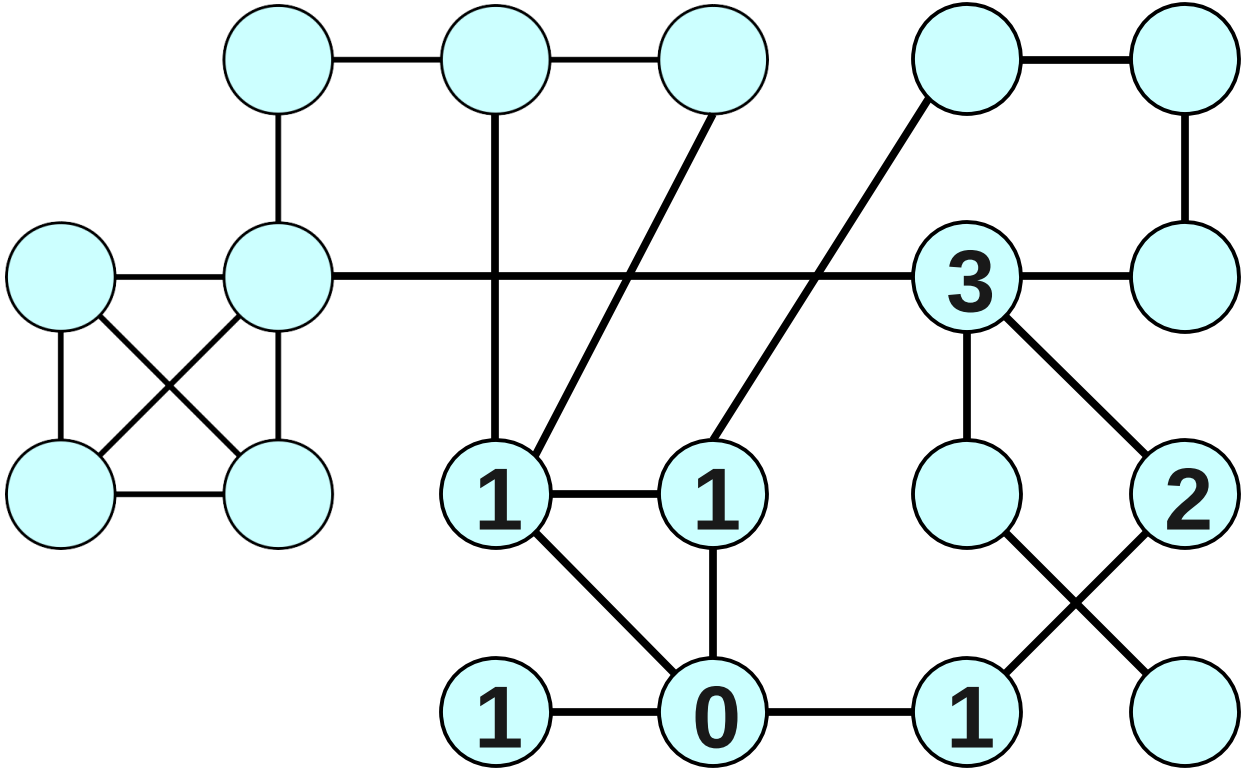
A Social Network



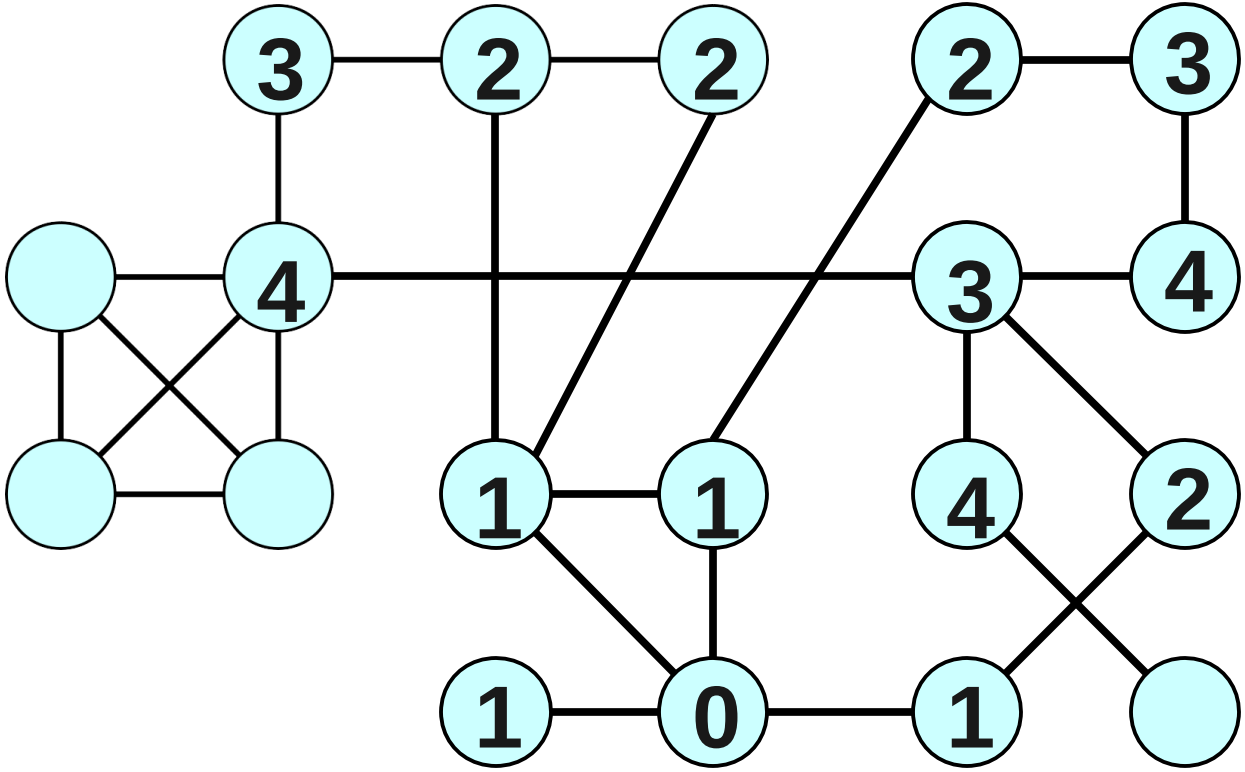
A Social Network



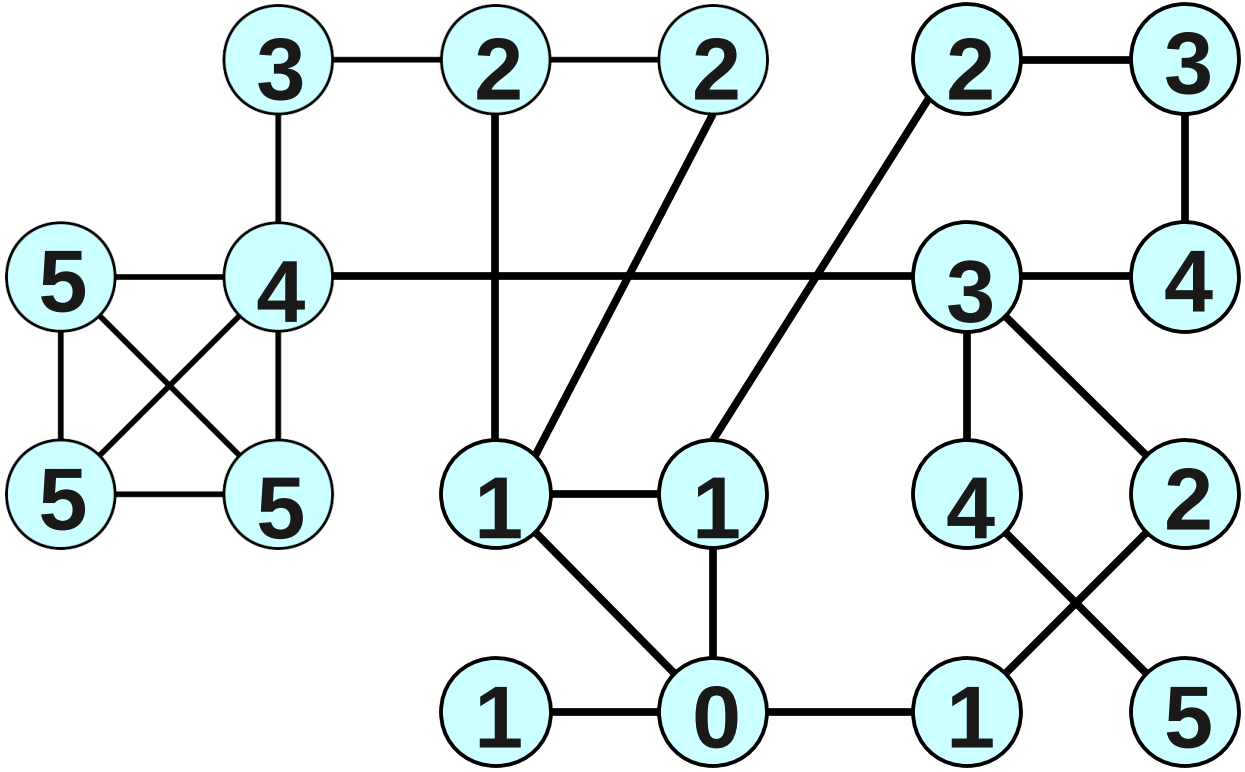
A Social Network



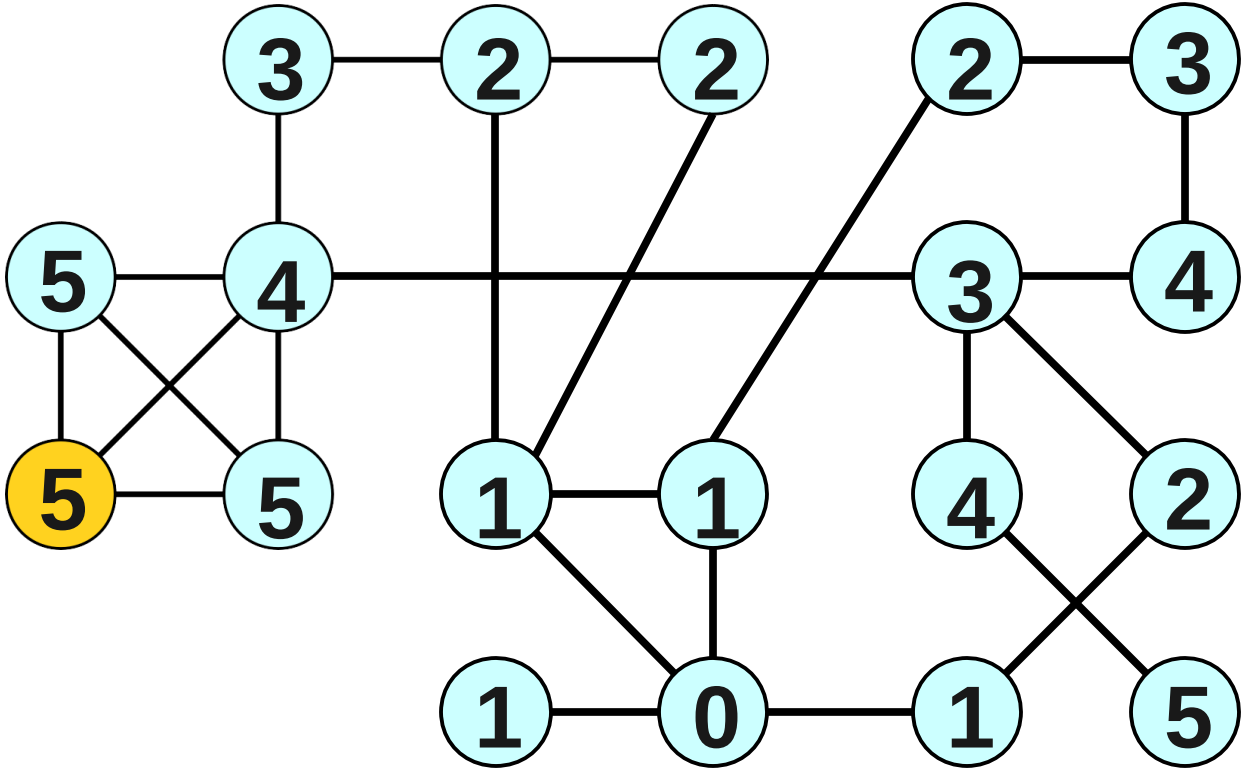
A Social Network



A Social Network

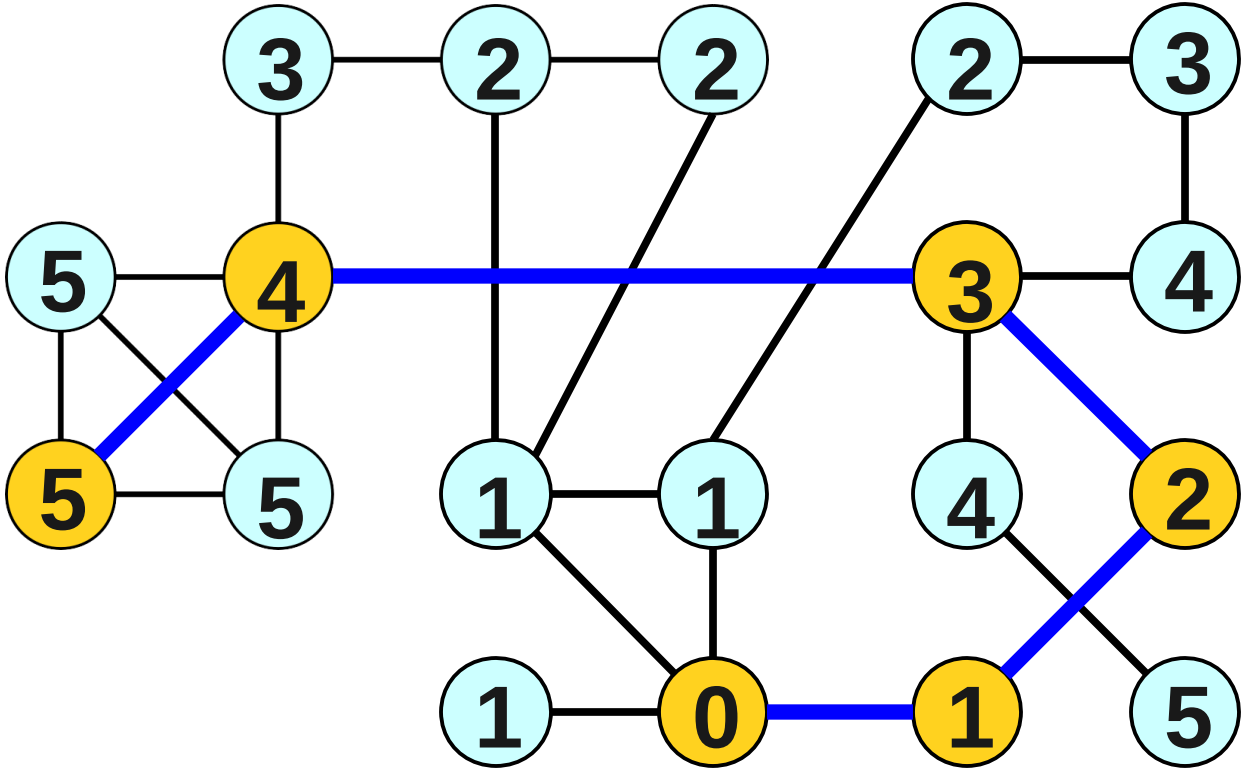


A Social Network

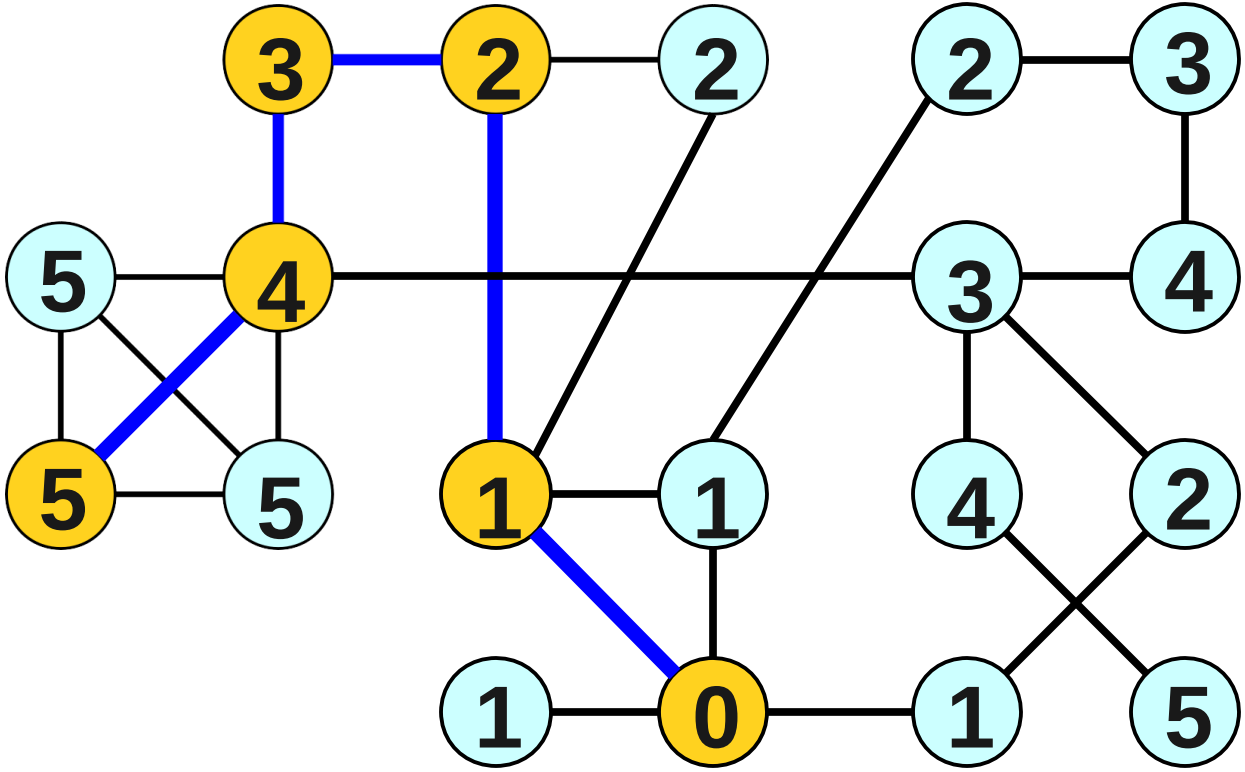


Can you find a path from this node (5) to the original node?

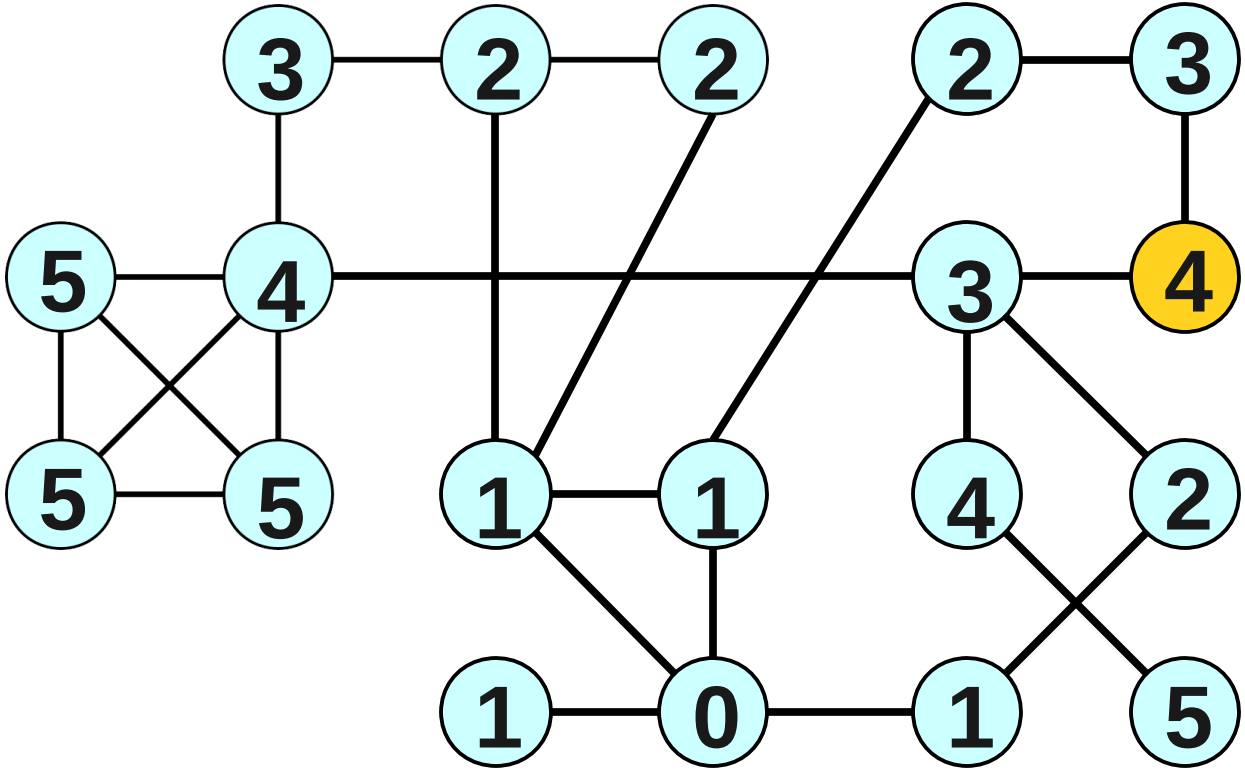
A Social Network



A Social Network

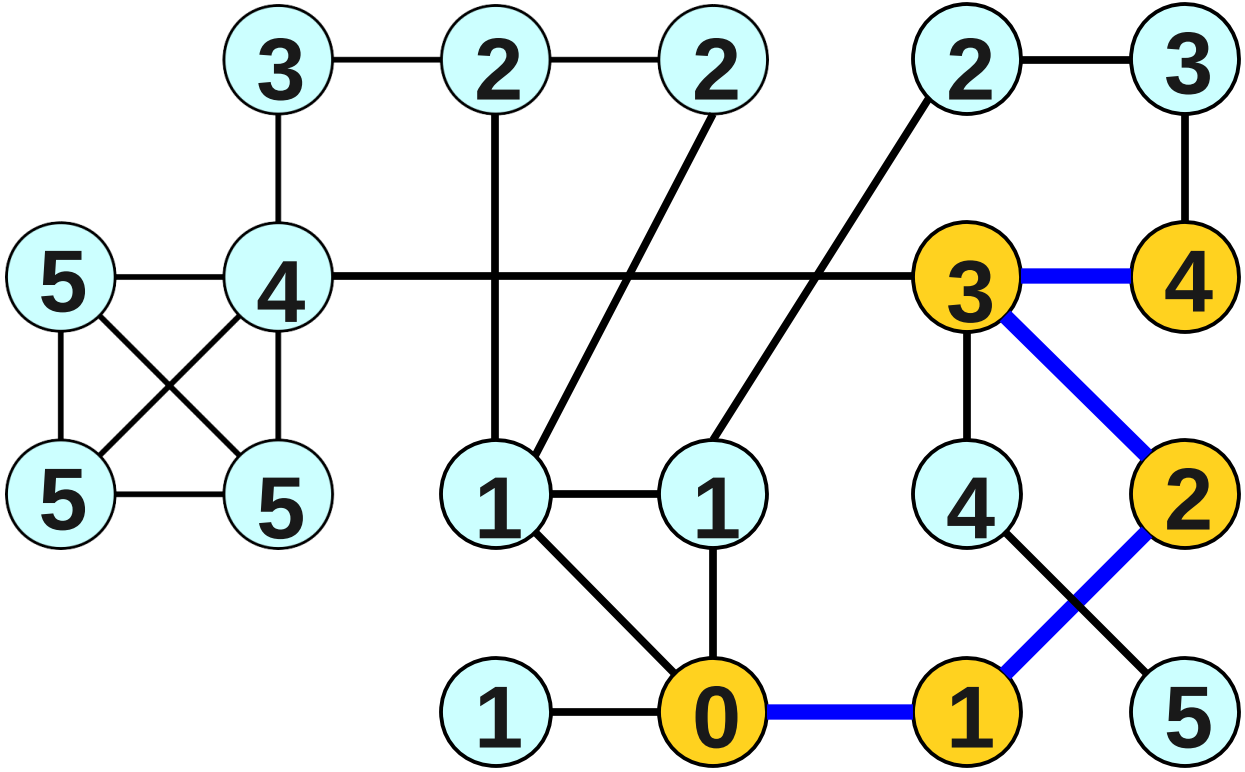


A Social Network

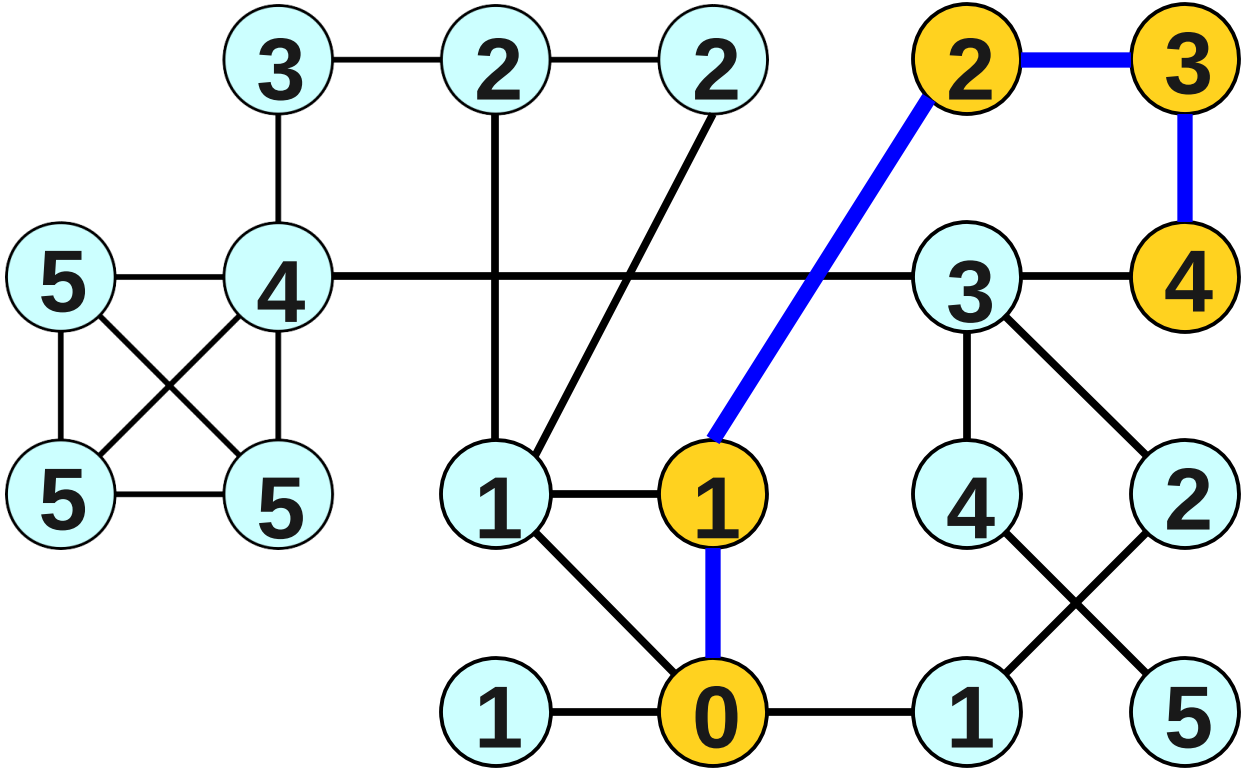


Can you find a path from this node (4) to the original node?

A Social Network



A Social Network



The Shortest Path Problem

- **Input:**
  - An undirected graph $G = (V, E)$.

    A start node $s \in V$.

- **Output:**
  - Distances from s to all nodes v.

The Shortest Path Problem

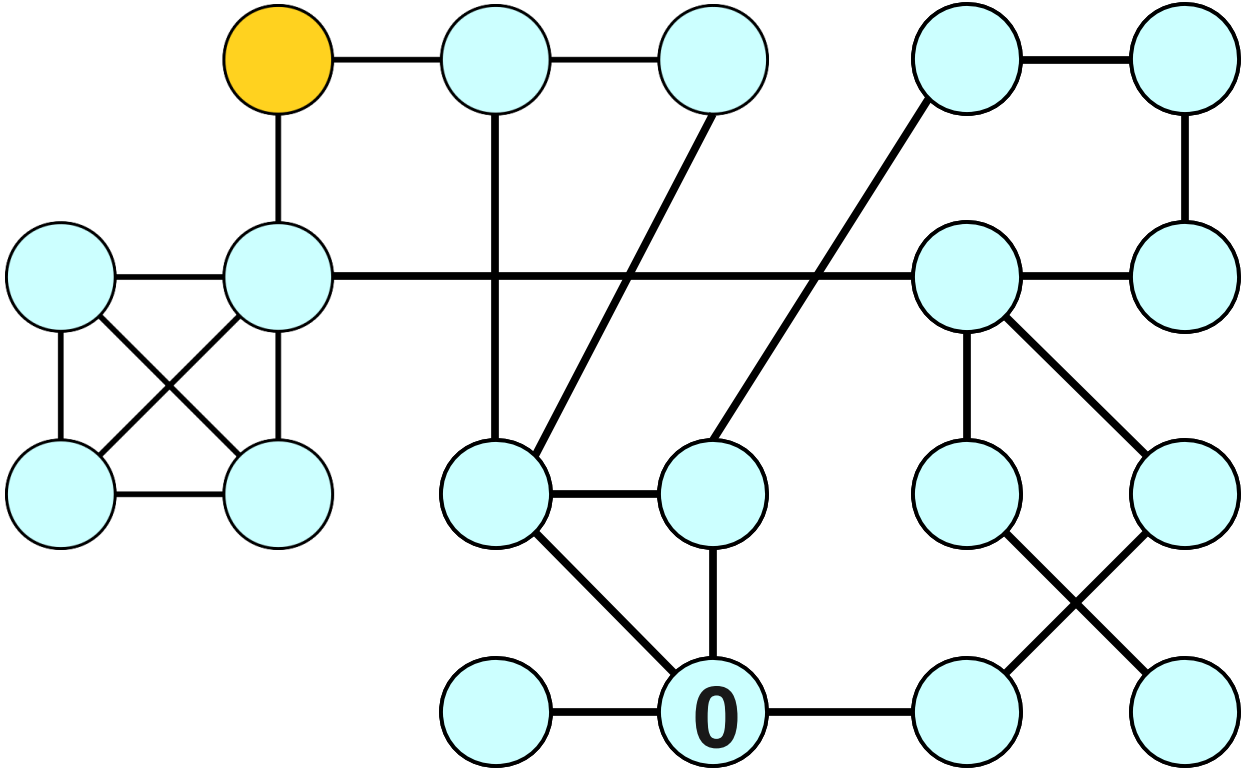
- **Input:**
  - An undirected graph $G = (V, E)$.

    A start node $s \in V$.

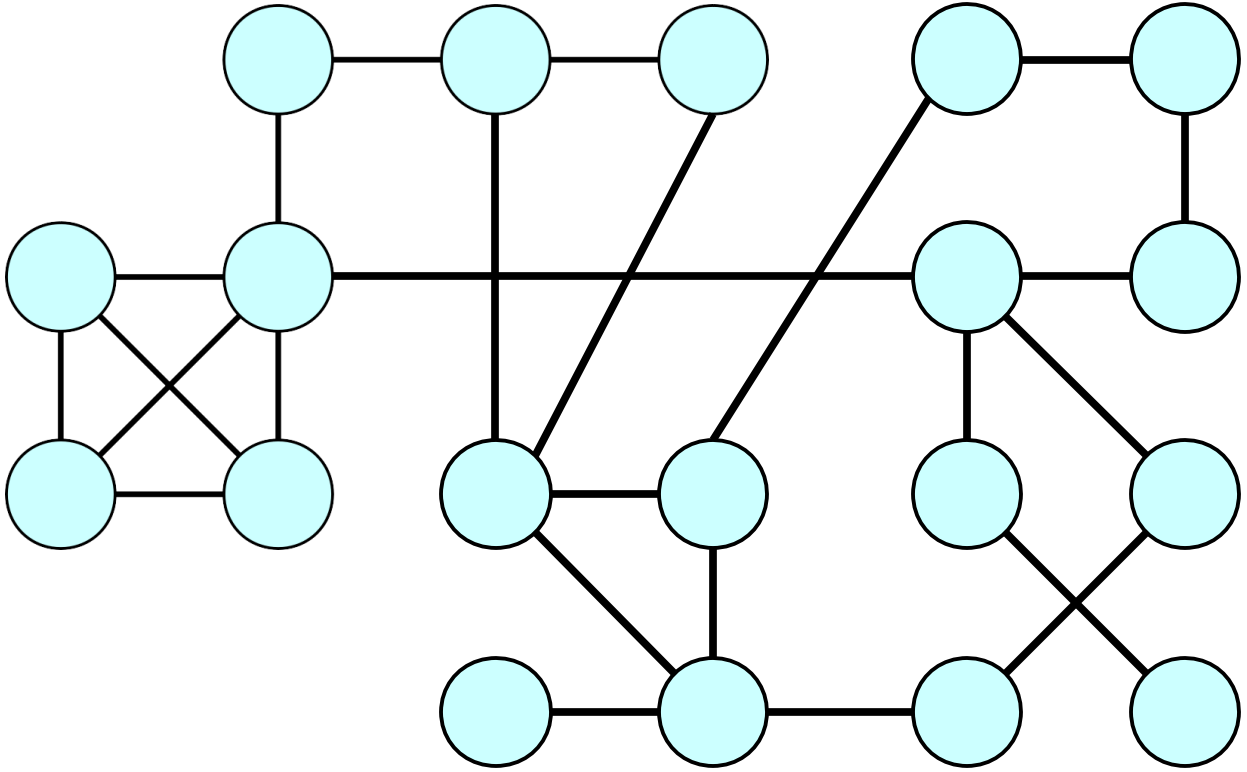
- **Output:**
  - Distances from s to all nodes v.

Allows us to find shortest paths
Allows us to find connected components
Allows us to find the distance to all other nodes in the connected component

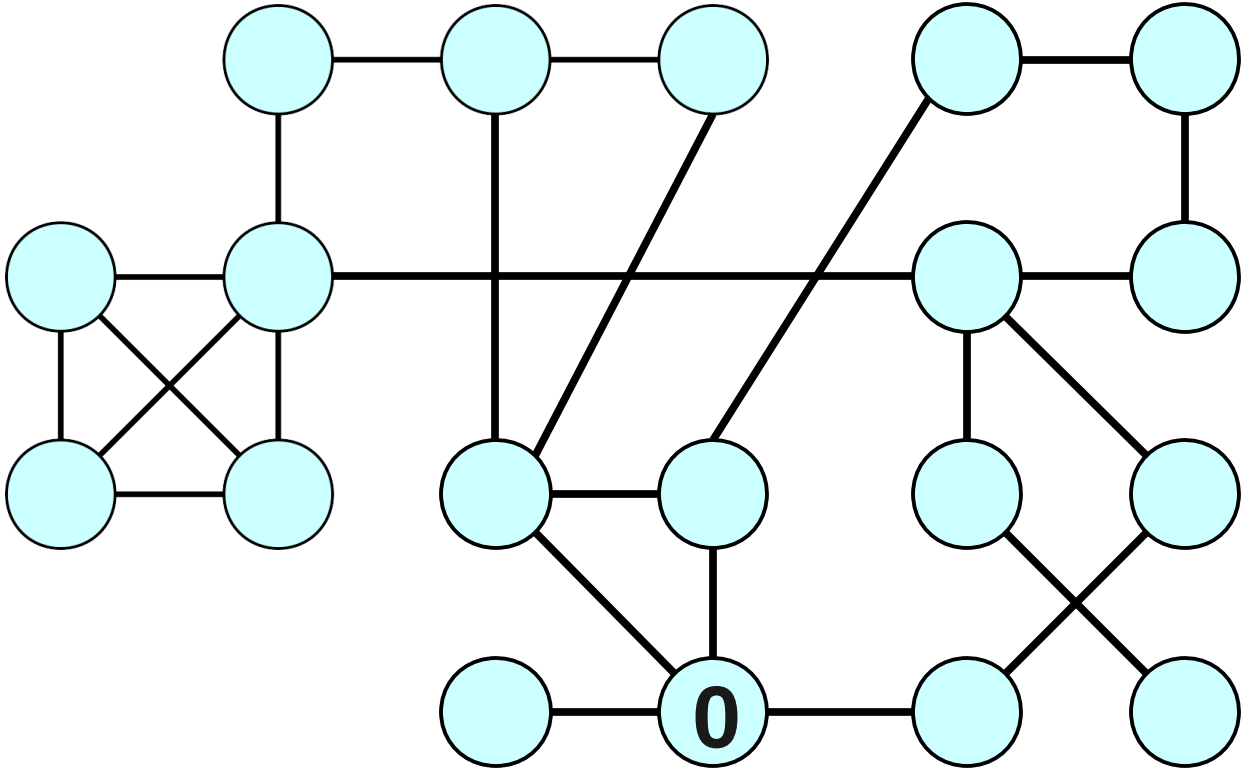
An Inefficient Algorithm



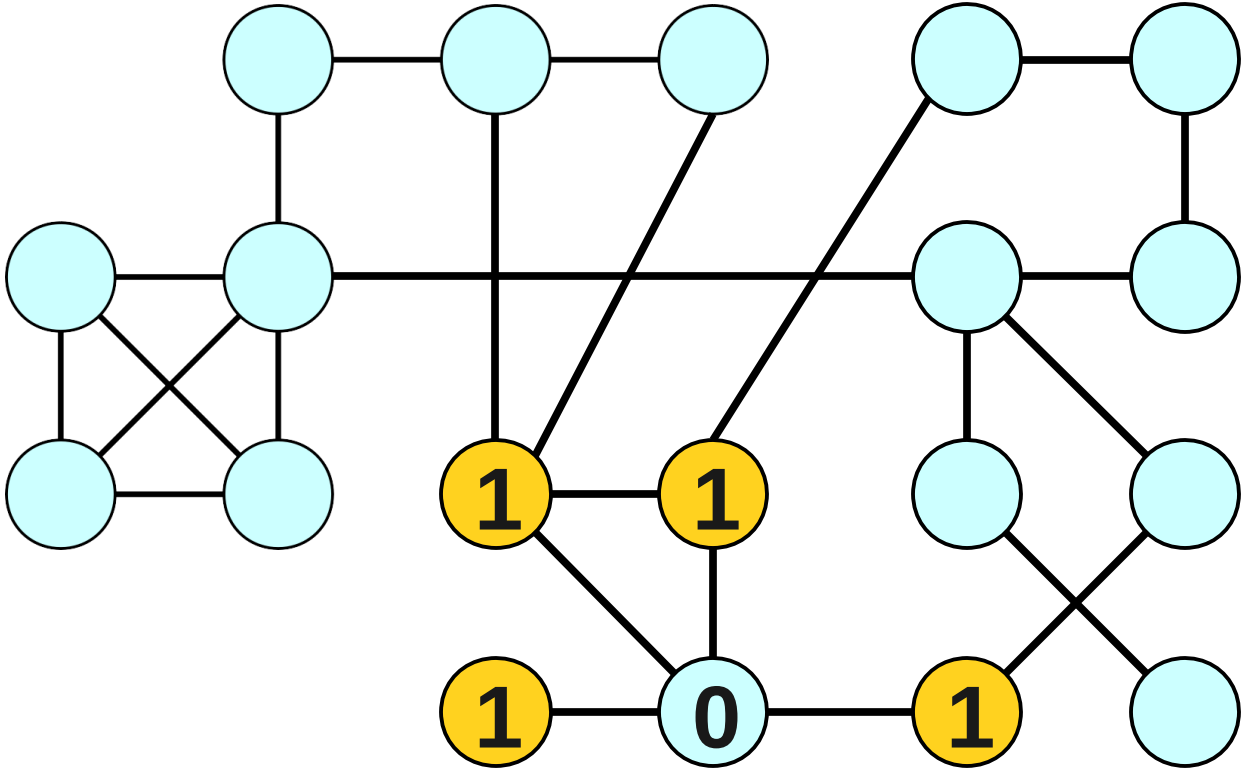
FINDING DISTANCES: A Better Algorithm



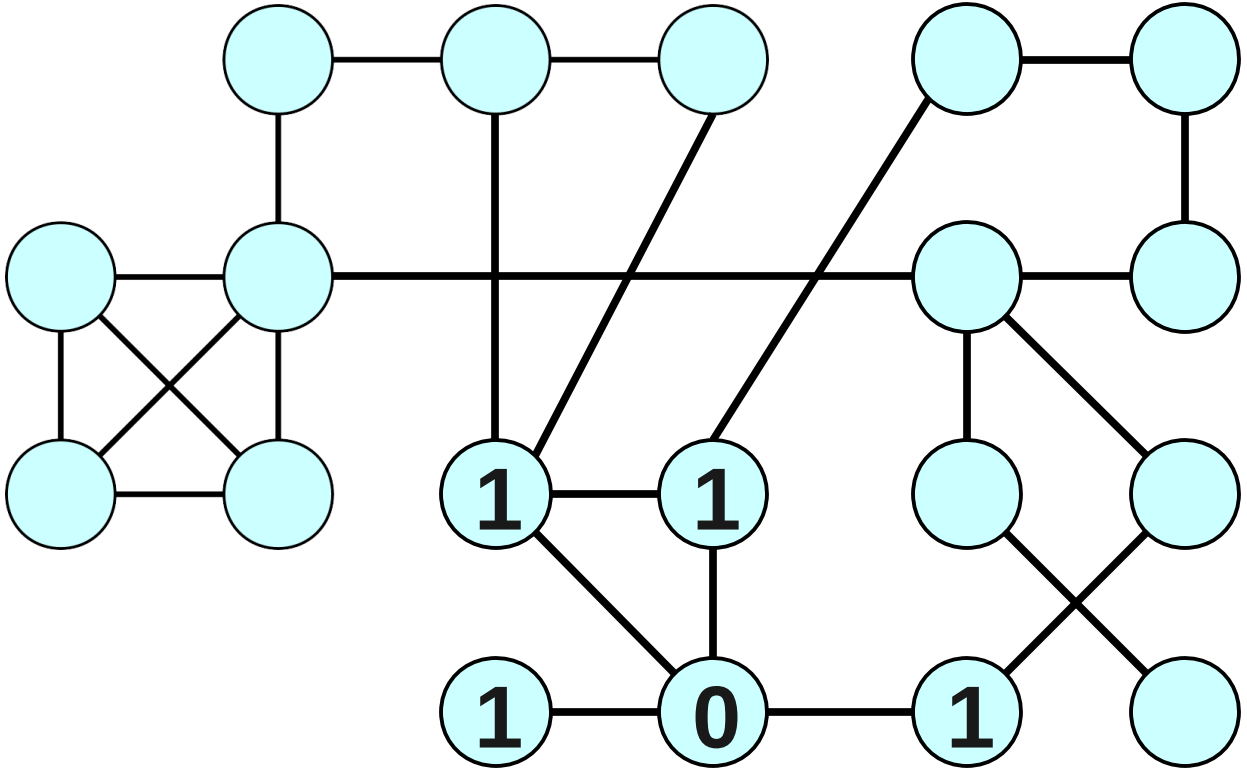
FINDING DISTANCES: A Better Algorithm



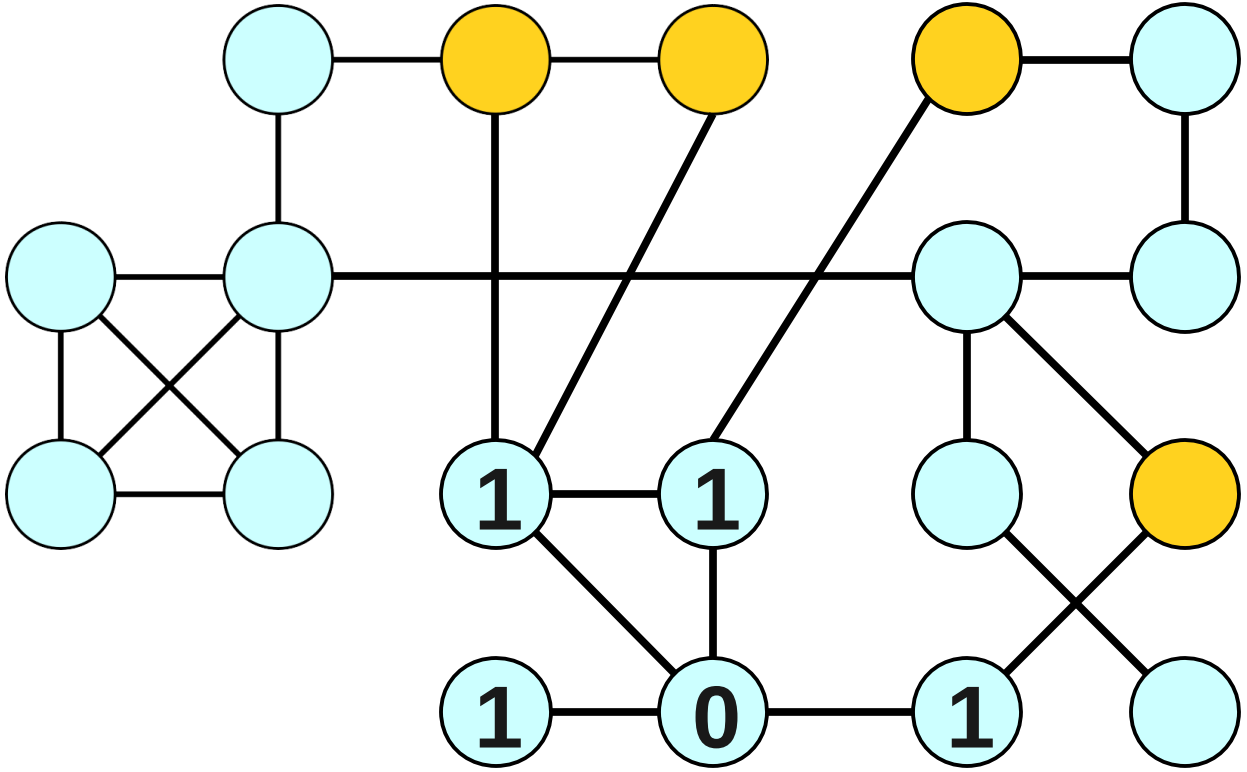
FINDING DISTANCES: A Better Algorithm



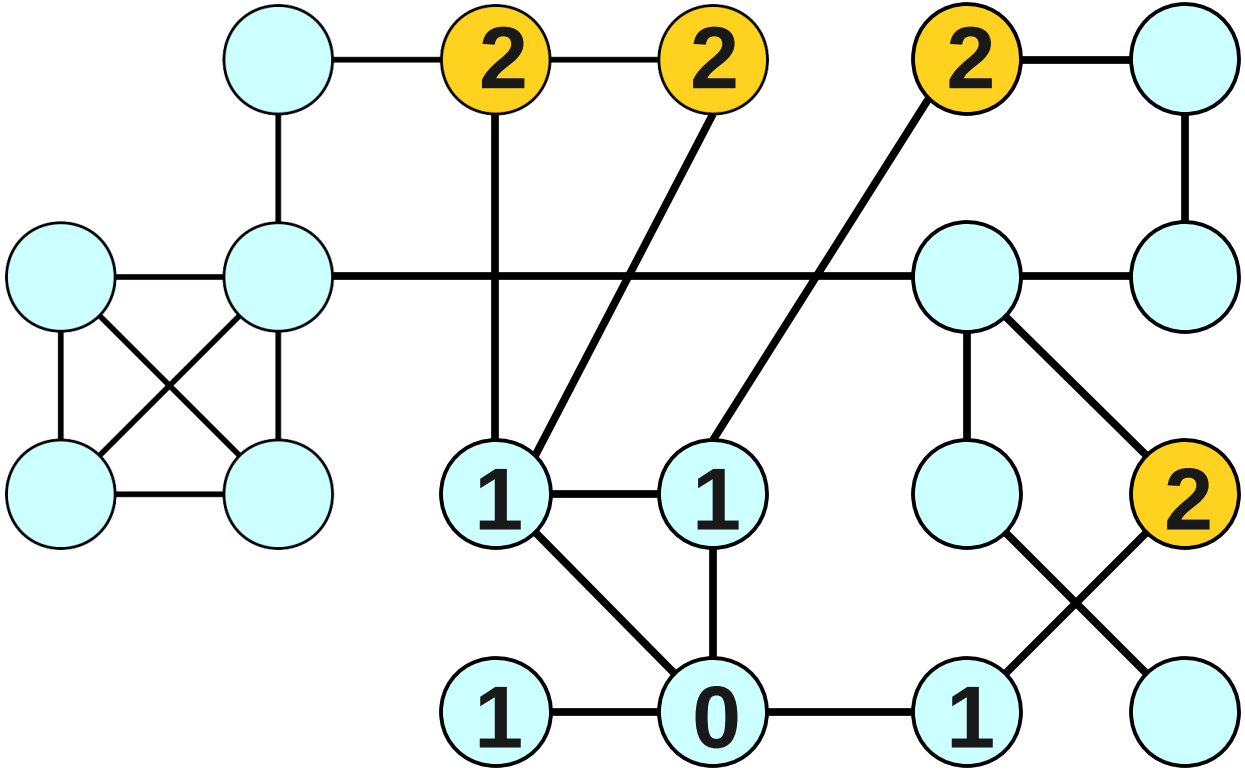
FINDING DISTANCES: A Better Algorithm



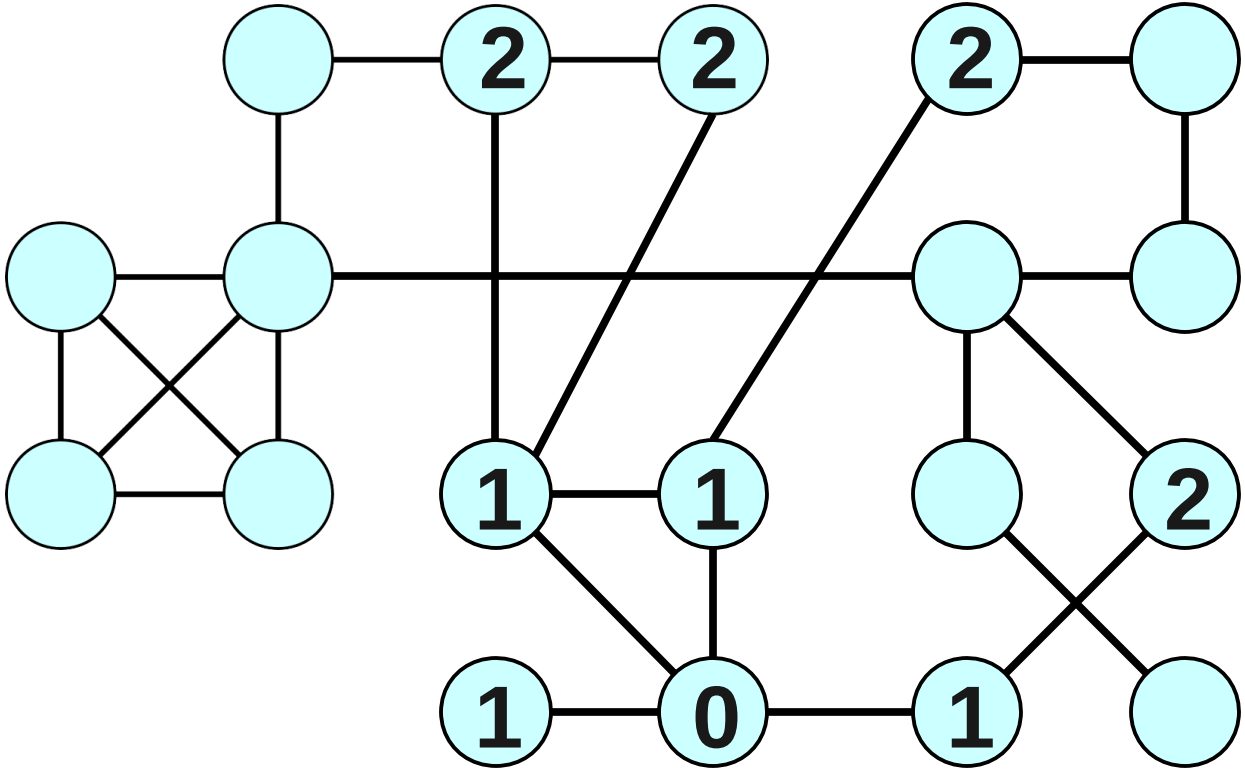
FINDING DISTANCES: A Better Algorithm



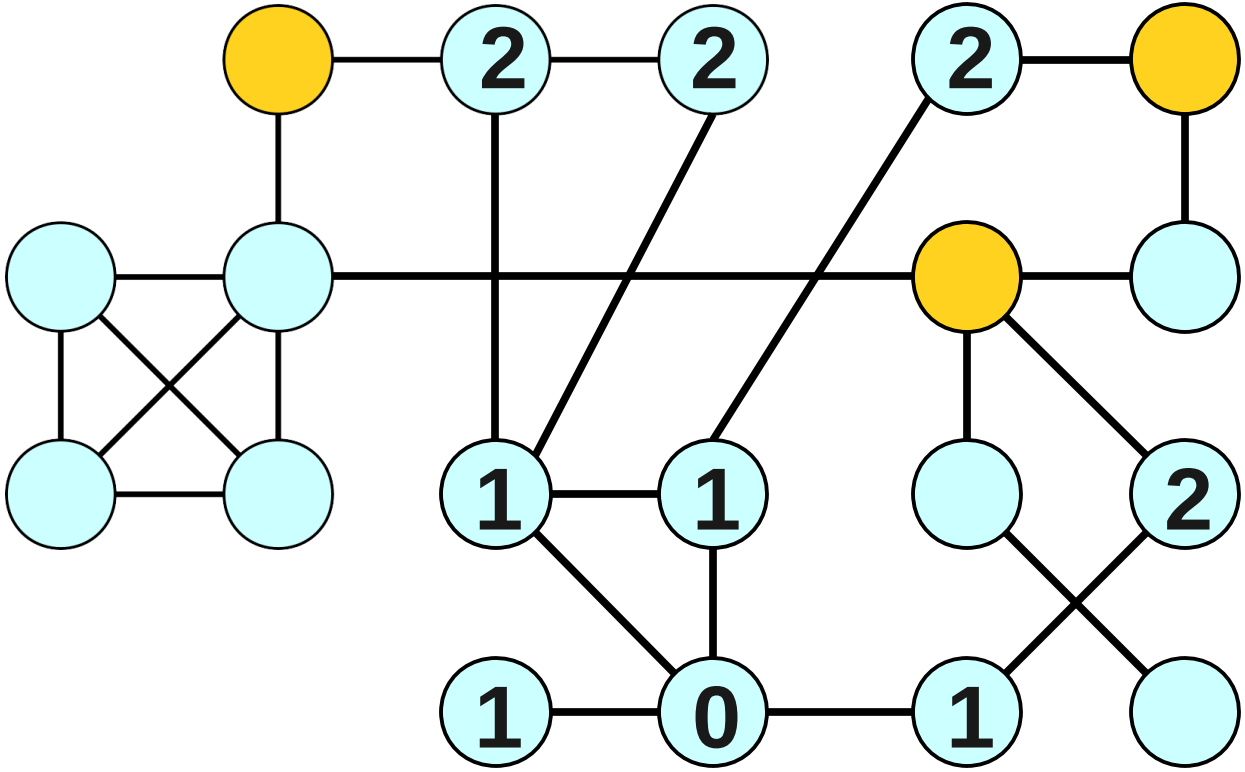
FINDING DISTANCES: A Better Algorithm



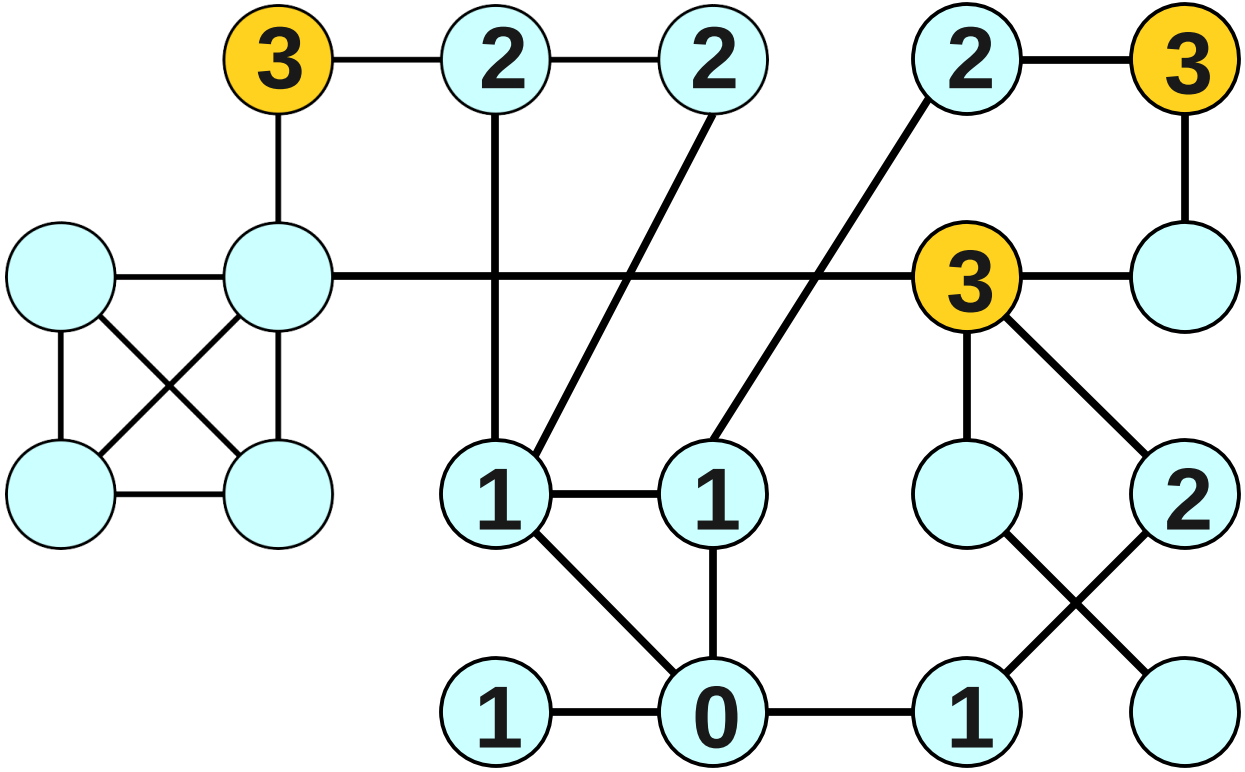
FINDING DISTANCES: A Better Algorithm



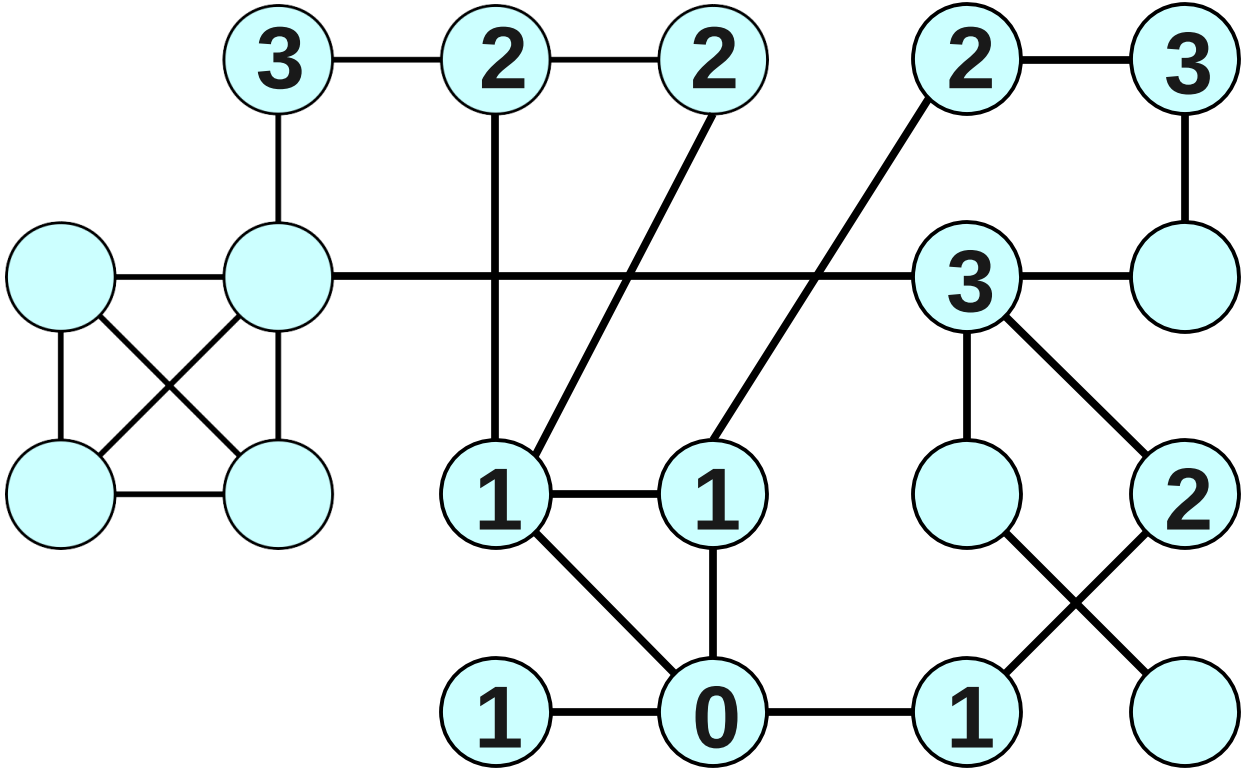
FINDING DISTANCES: A Better Algorithm



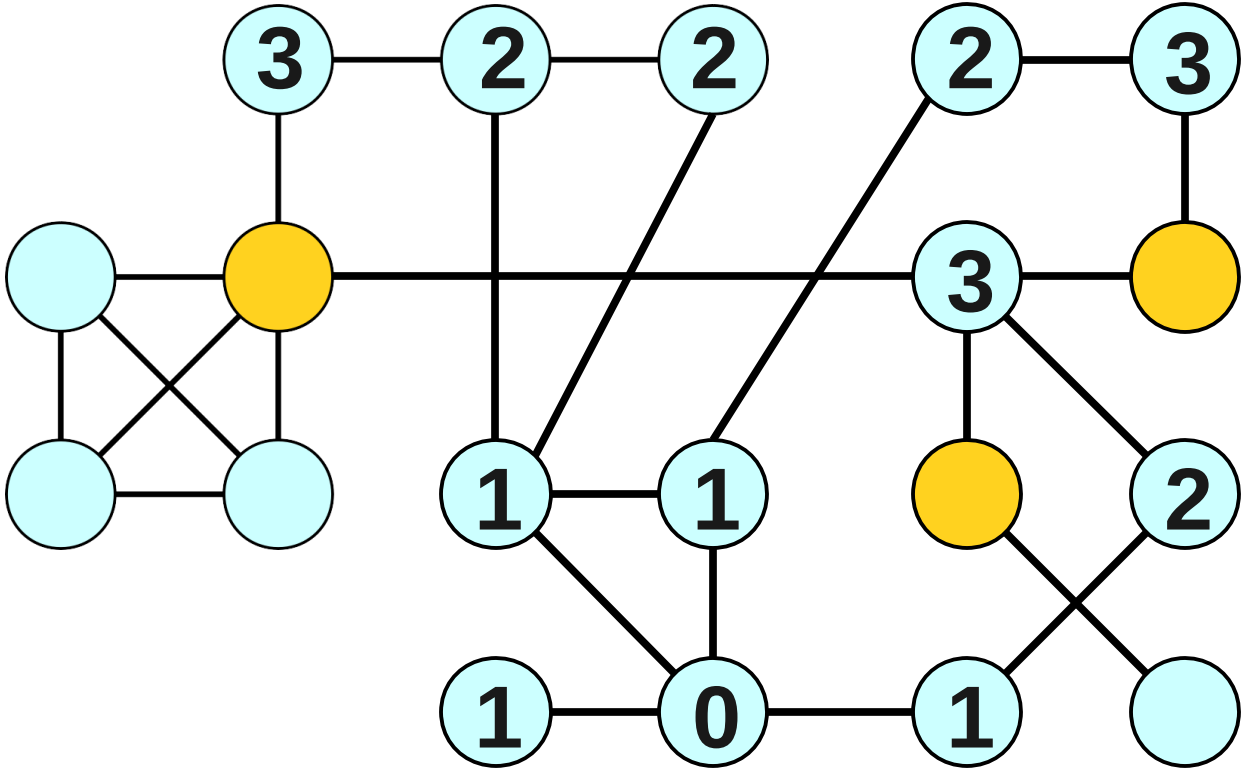
FINDING DISTANCES: A Better Algorithm



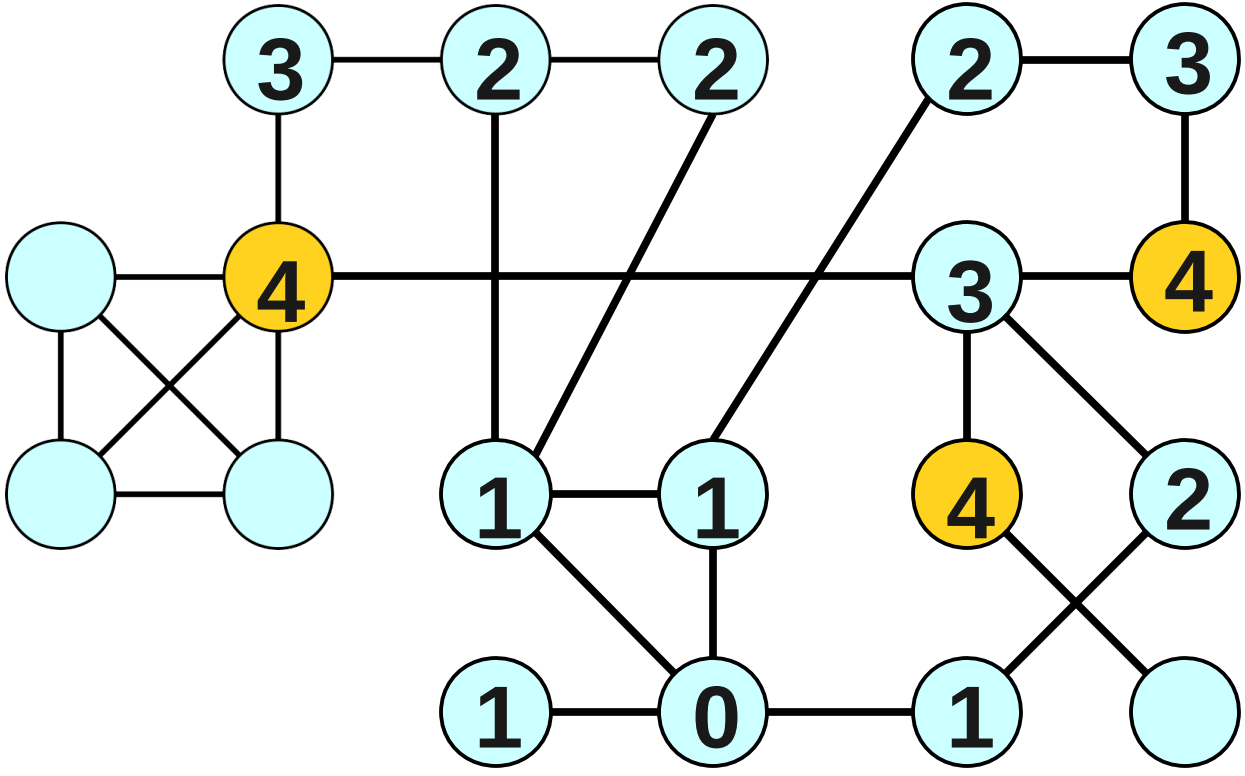
FINDING DISTANCES: A Better Algorithm



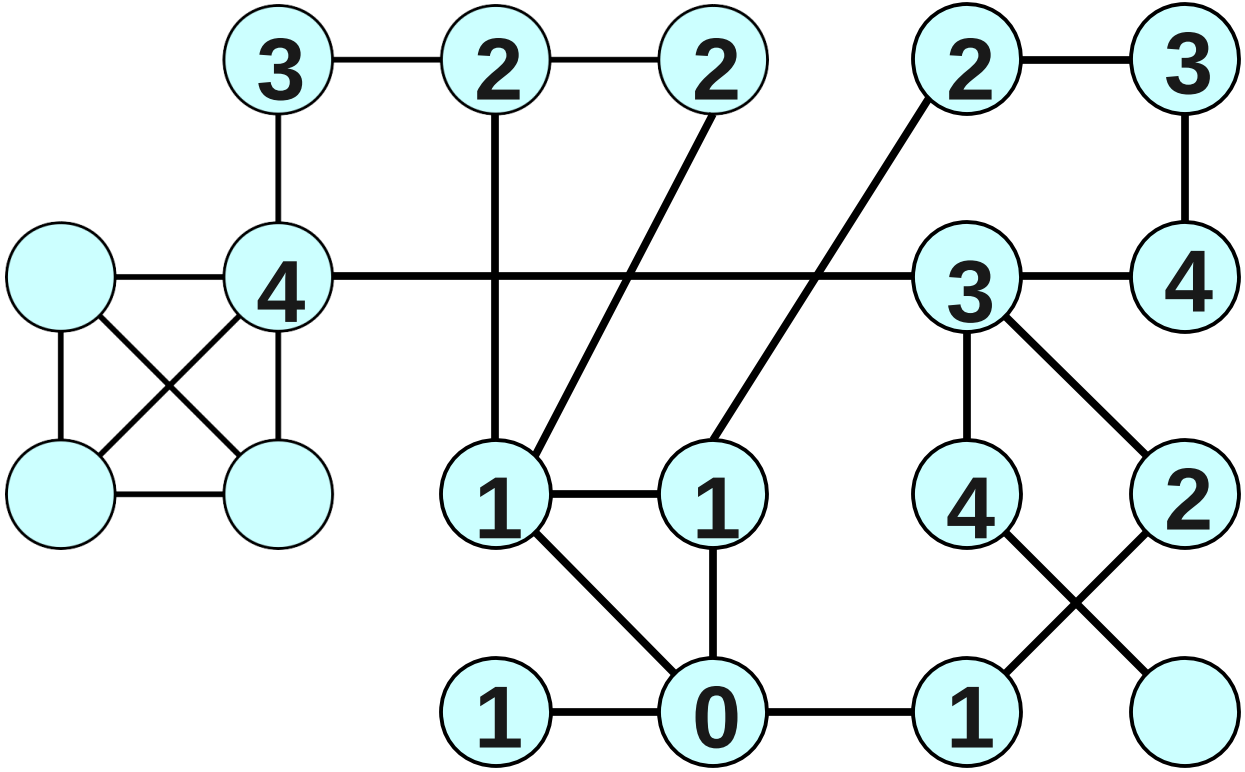
FINDING DISTANCES: A Better Algorithm



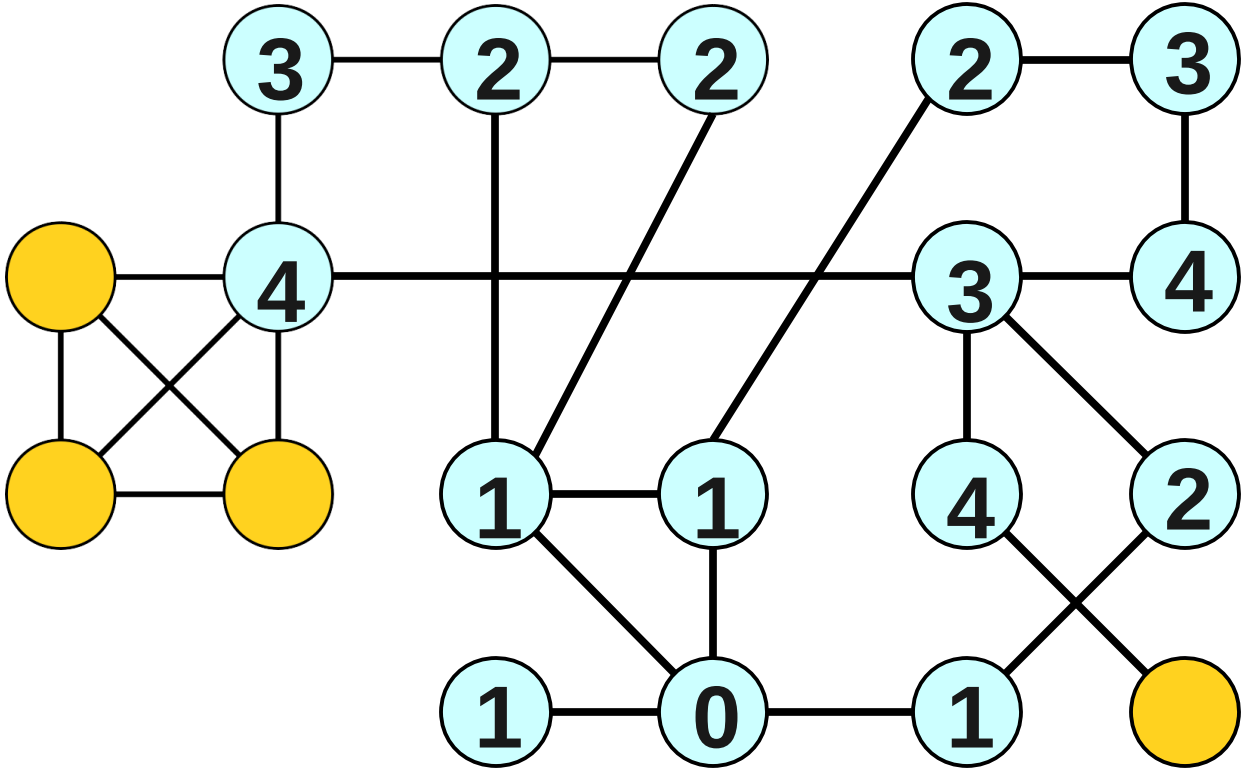
FINDING DISTANCES: A Better Algorithm



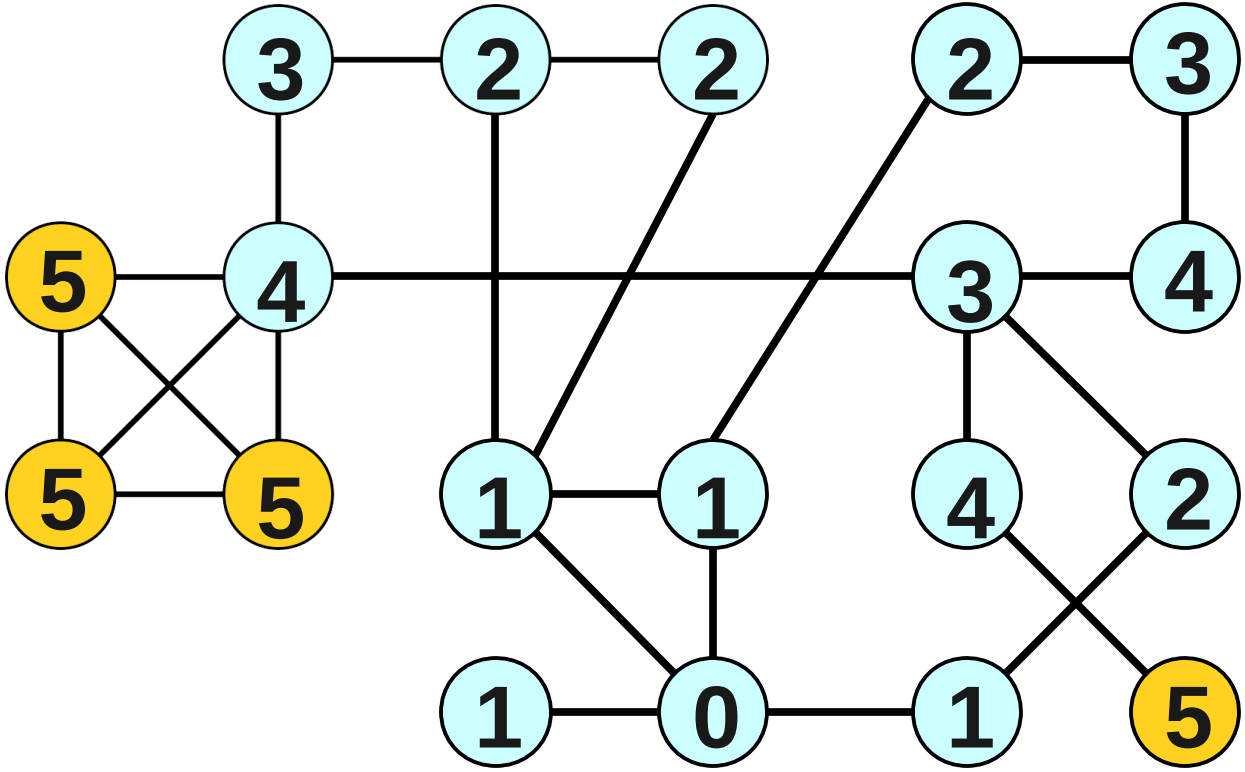
FINDING DISTANCES: A Better Algorithm



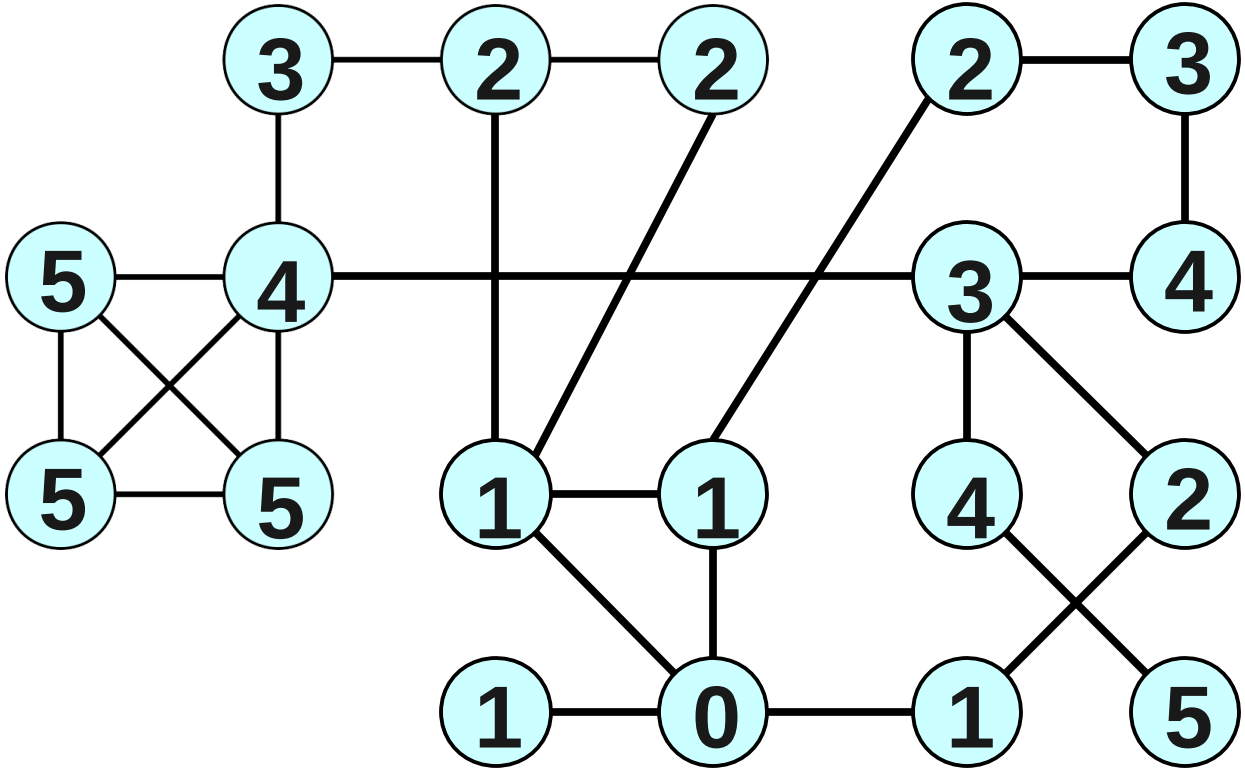
FINDING DISTANCES: A Better Algorithm



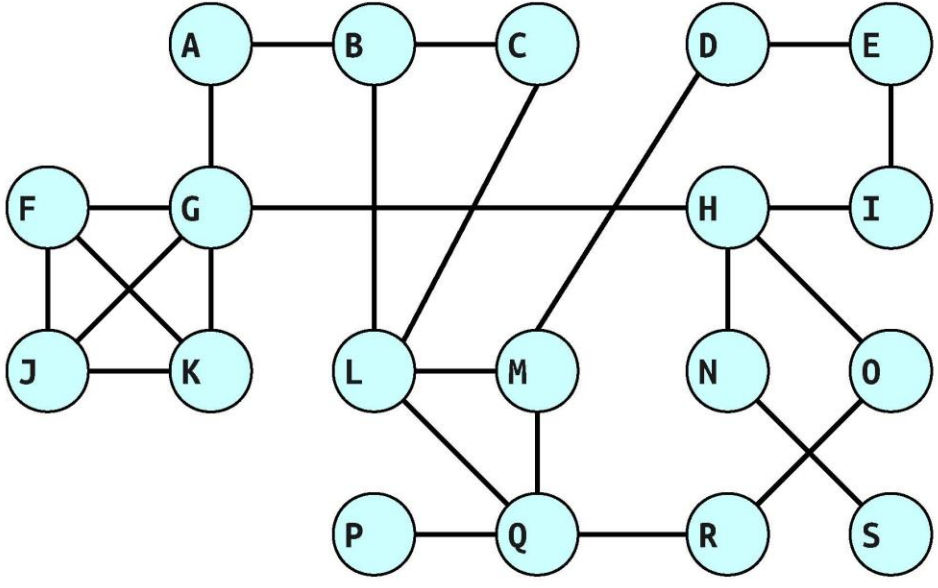
FINDING DISTANCES: A Better Algorithm



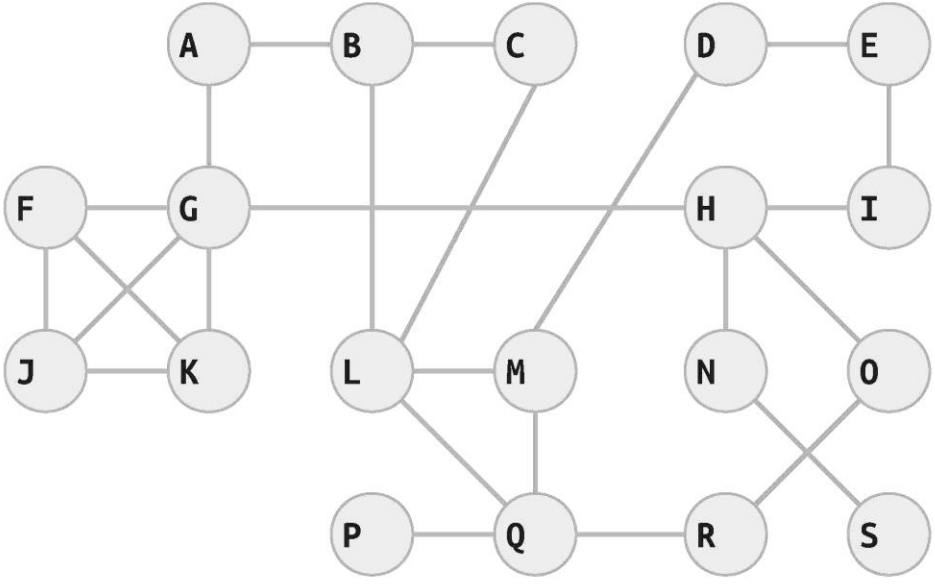
FINDING DISTANCES: A Better Algorithm



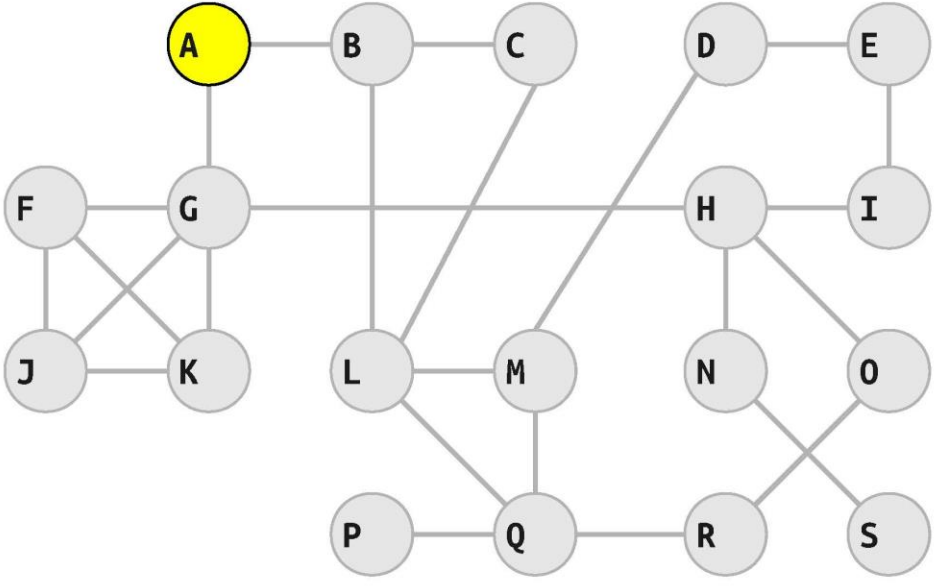
Breadth-First Search



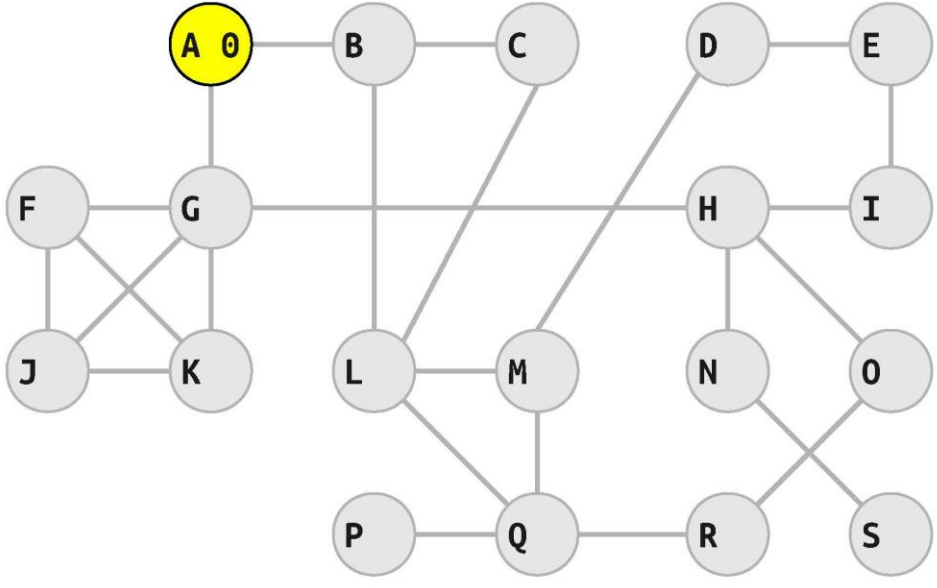
Breadth-First Search



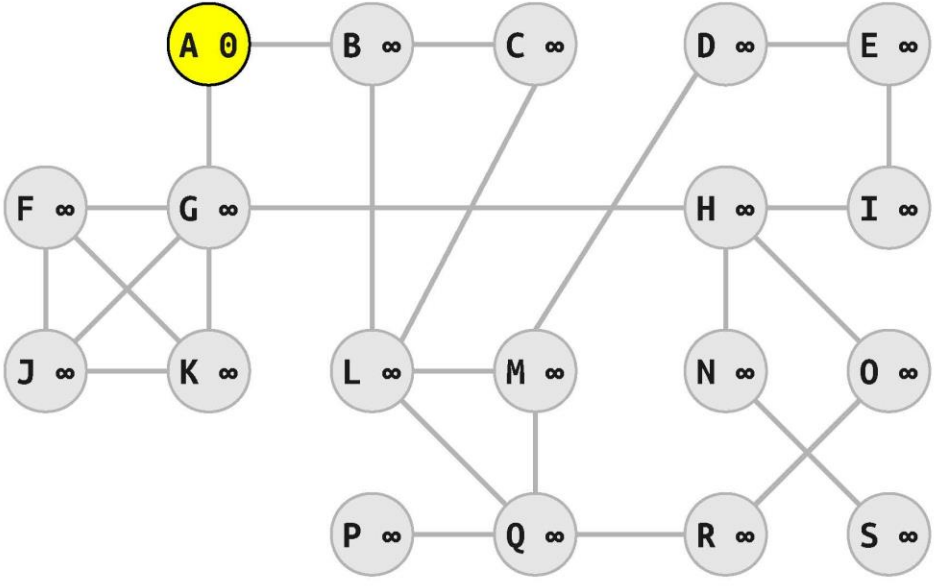
Breadth-First Search



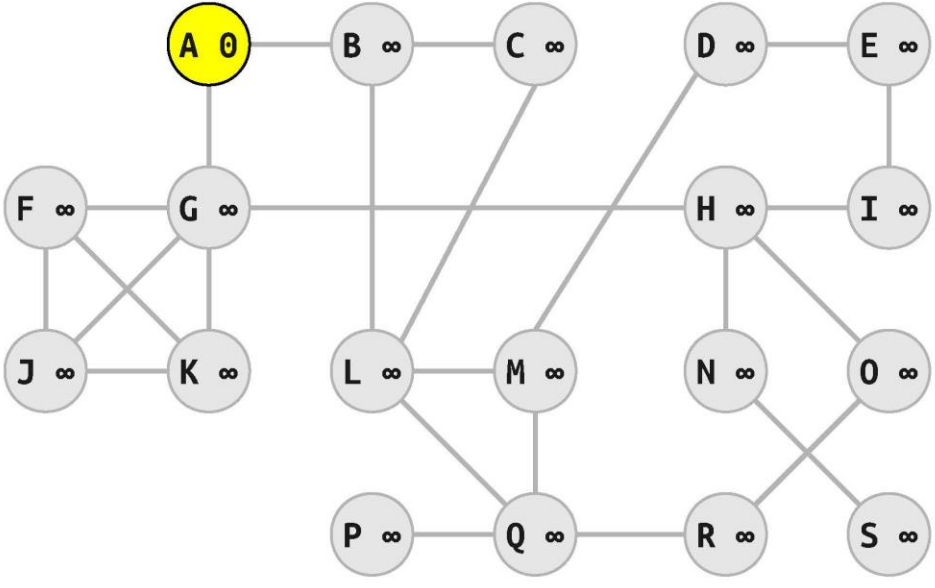
Breadth-First Search



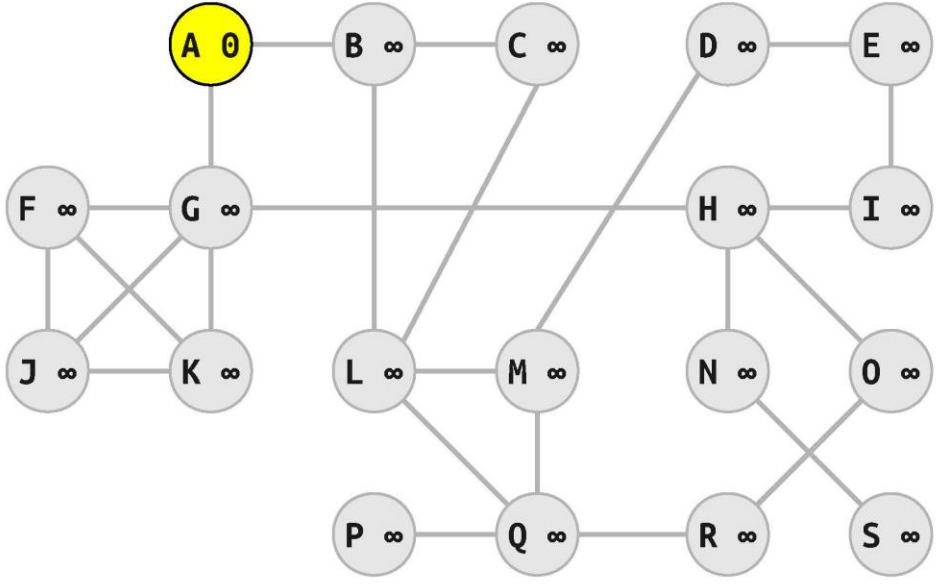
Breadth-First Search



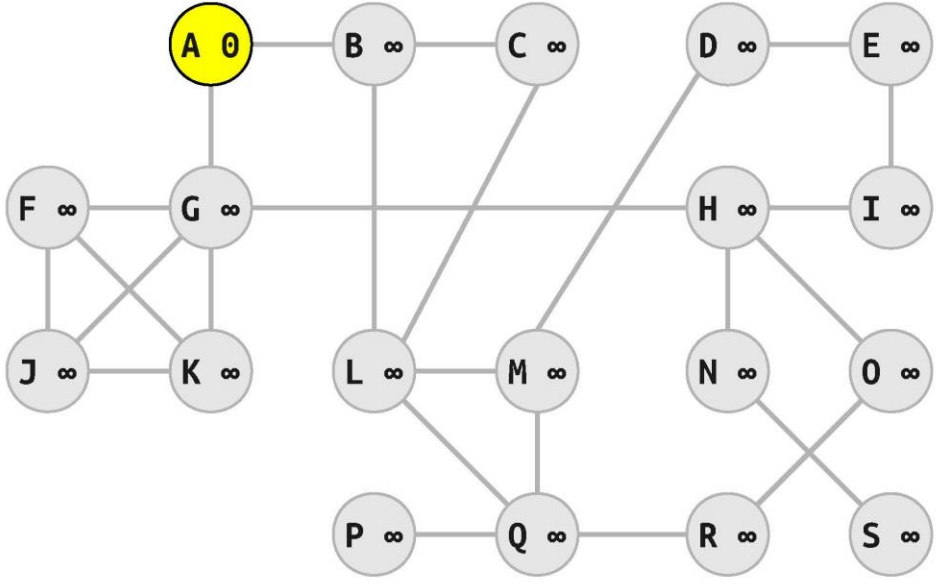
Breadth-First Search



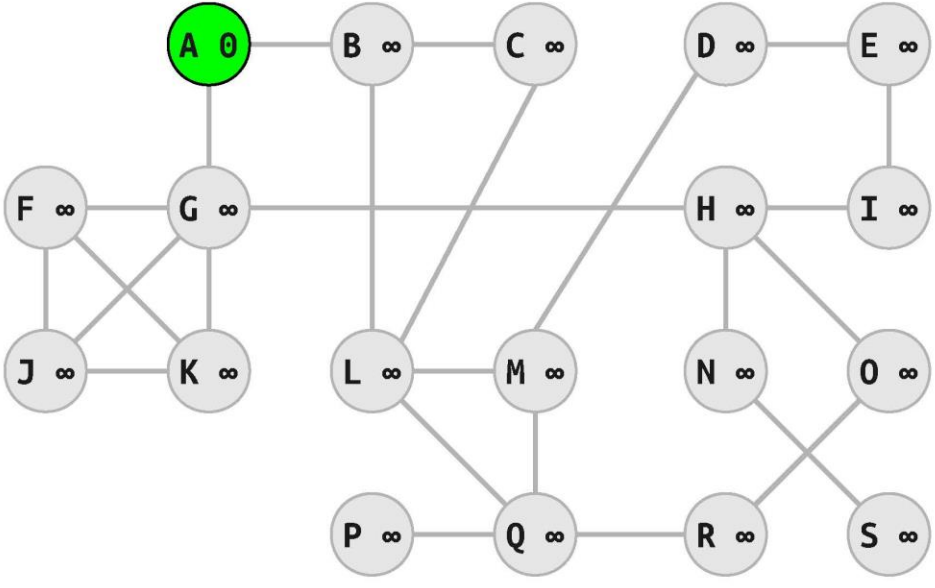
Breadth-First Search



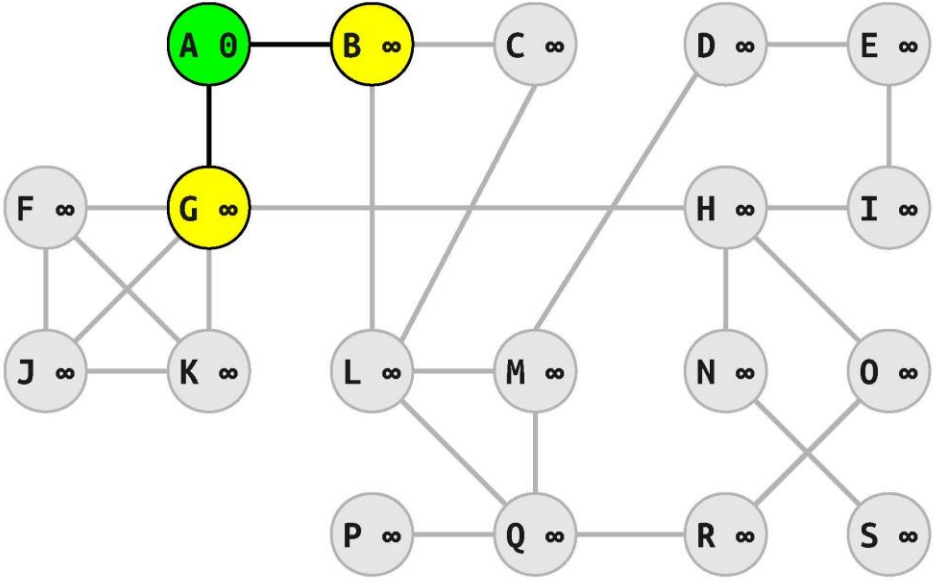
Breadth-First Search



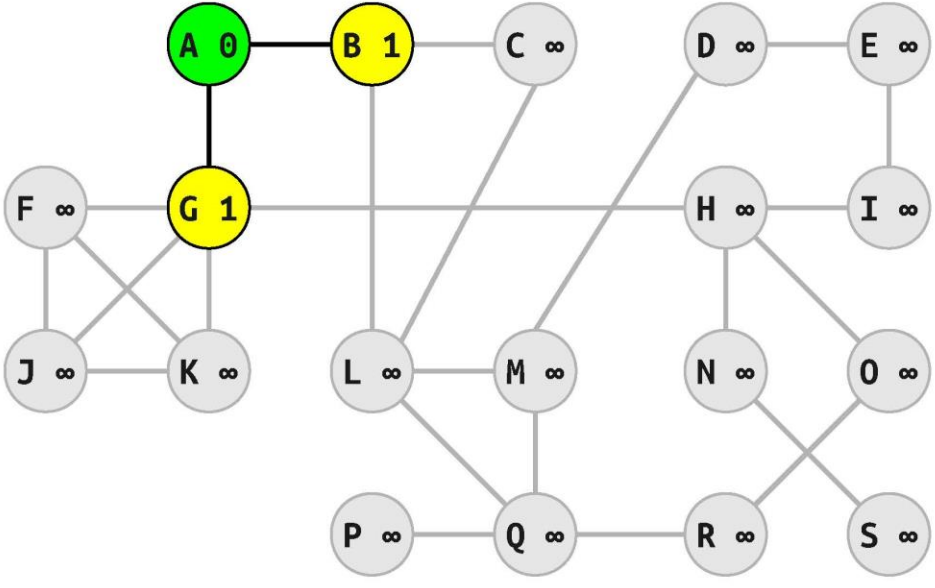
Breadth-First Search



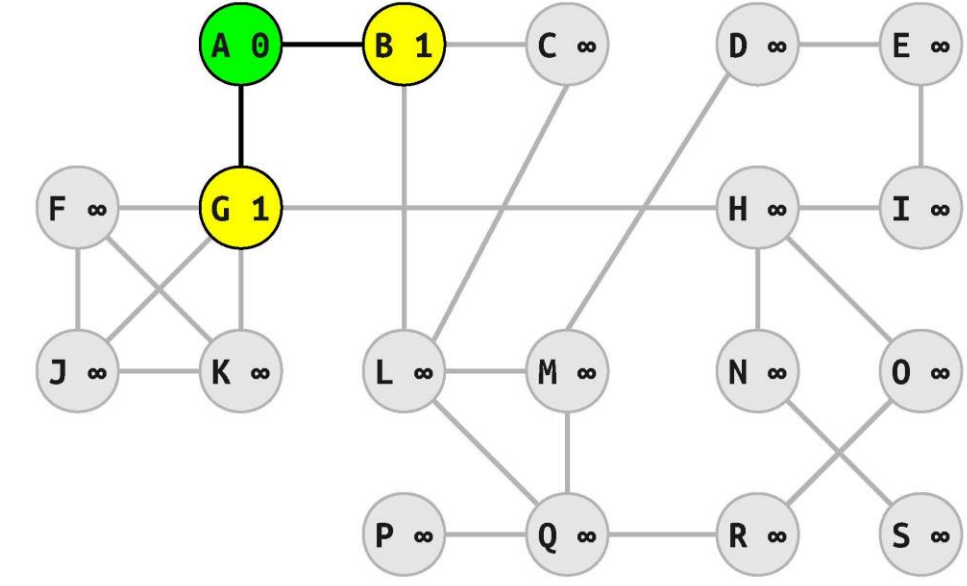
Breadth-First Search

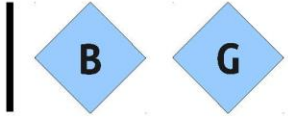


Breadth-First Search

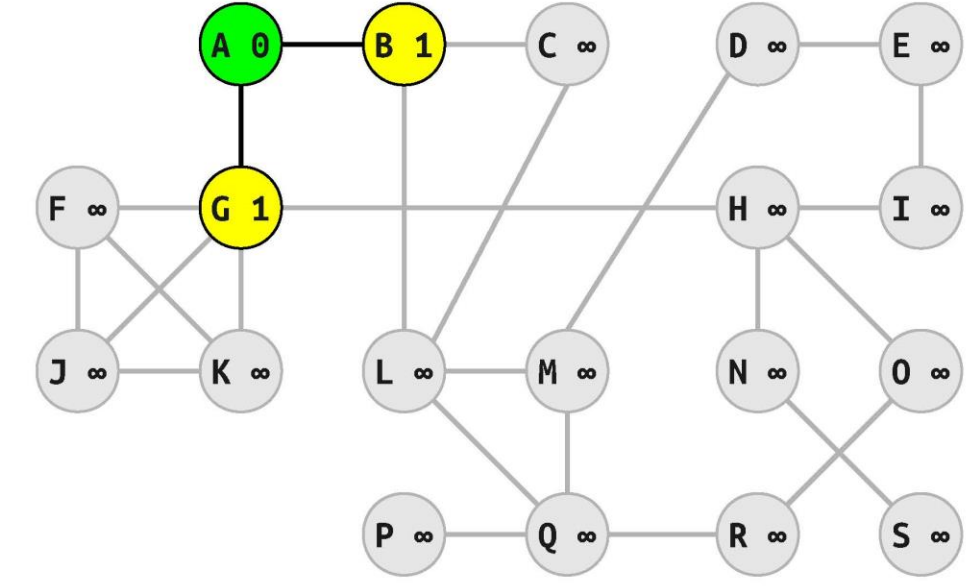


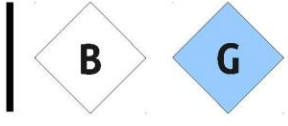
Breadth-First Search



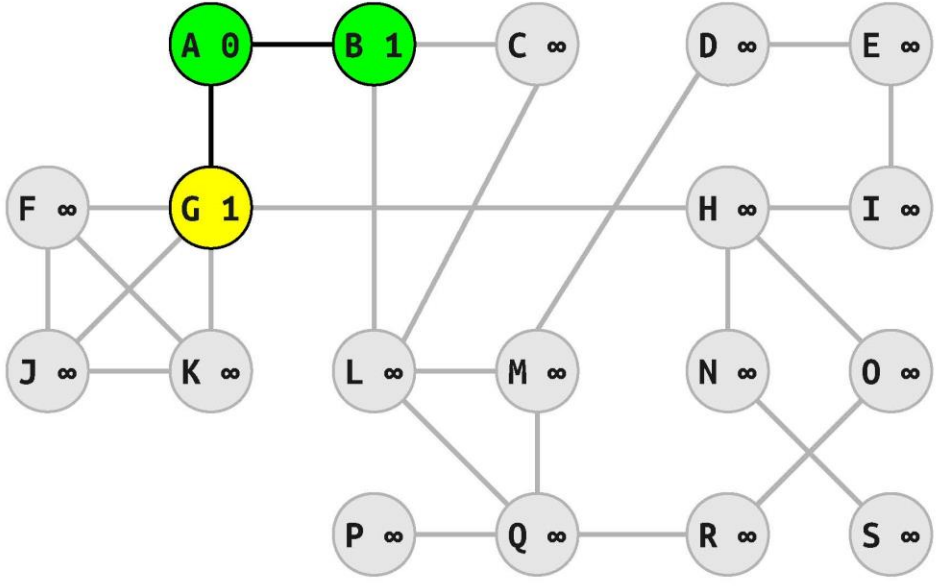


Breadth-First Search

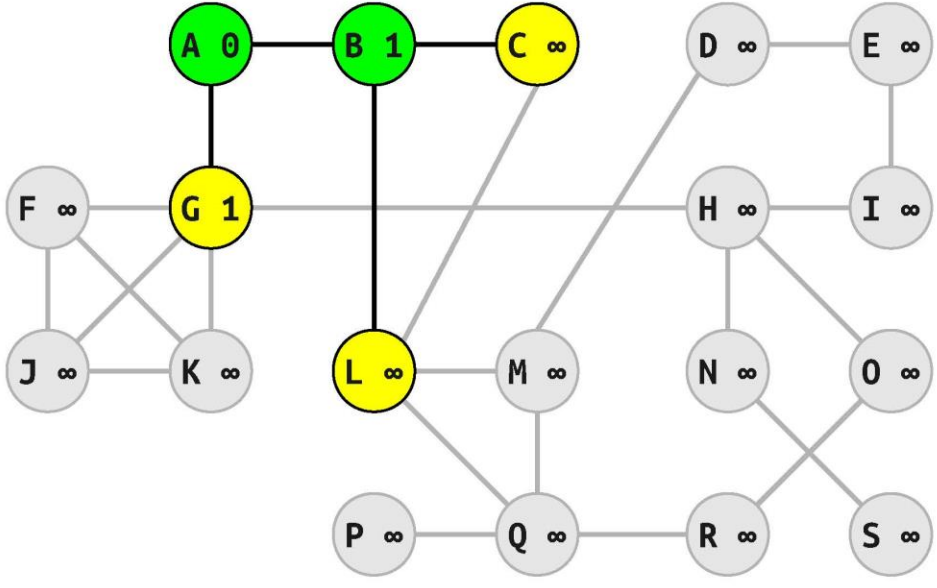




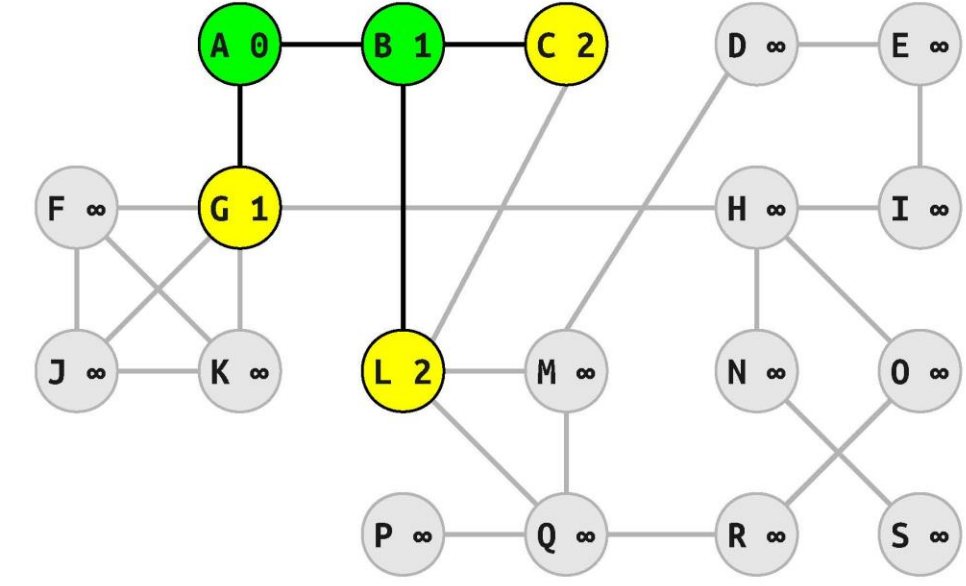
Breadth-First Search



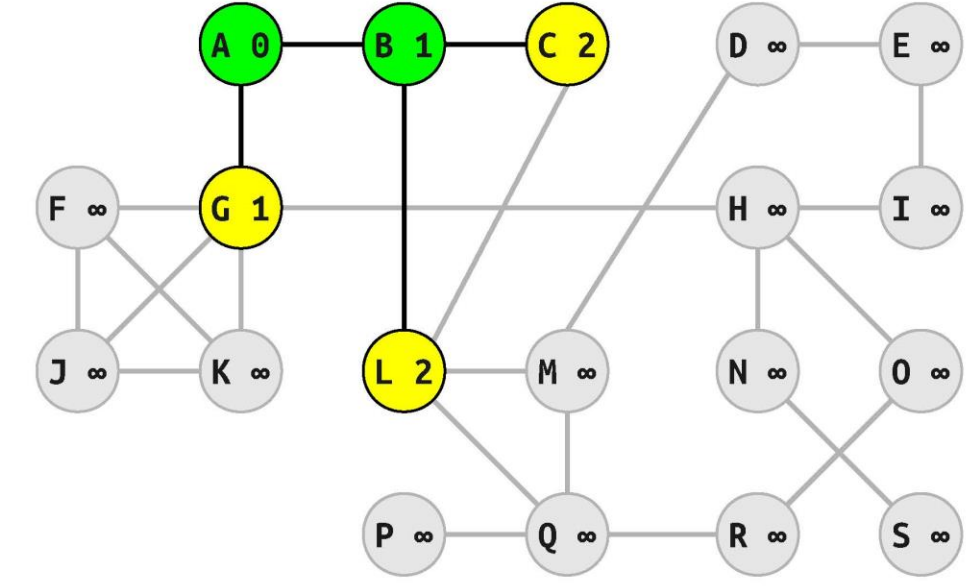
Breadth-First Search

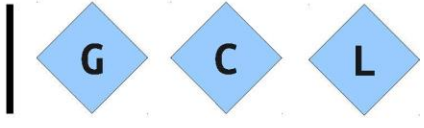


Breadth-First Search

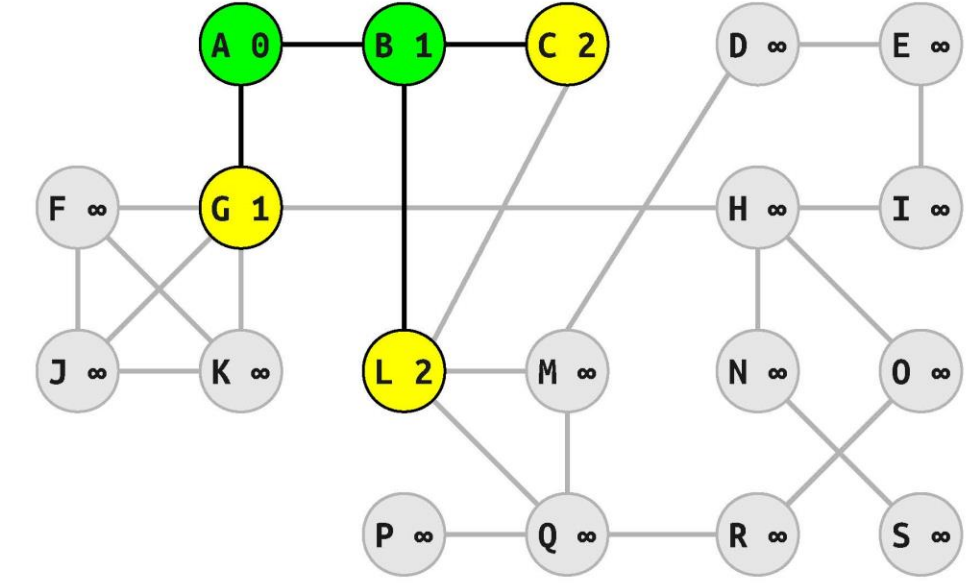


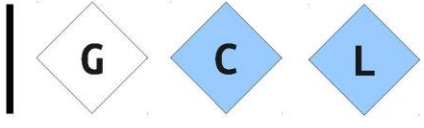
Breadth-First Search



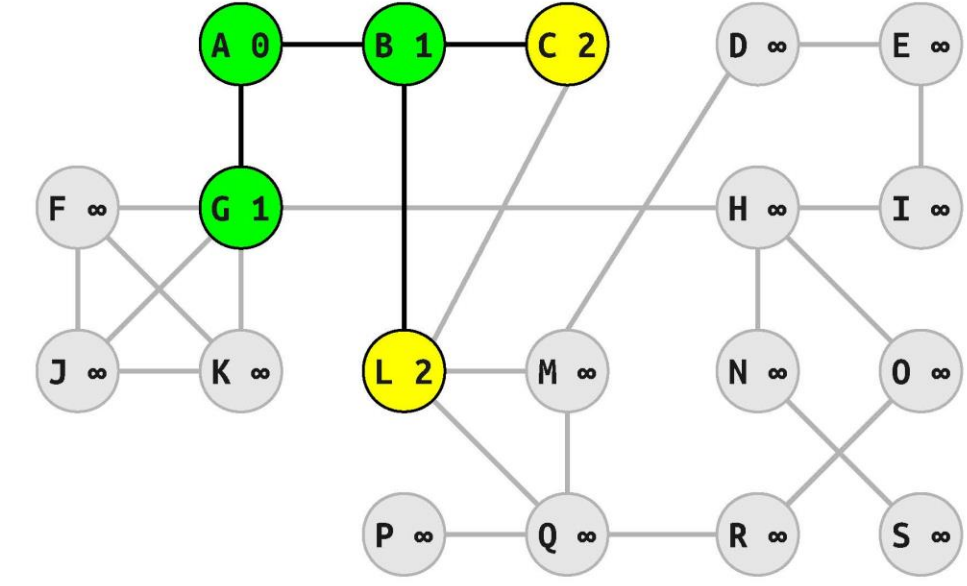


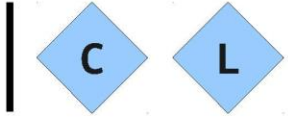
Breadth-First Search



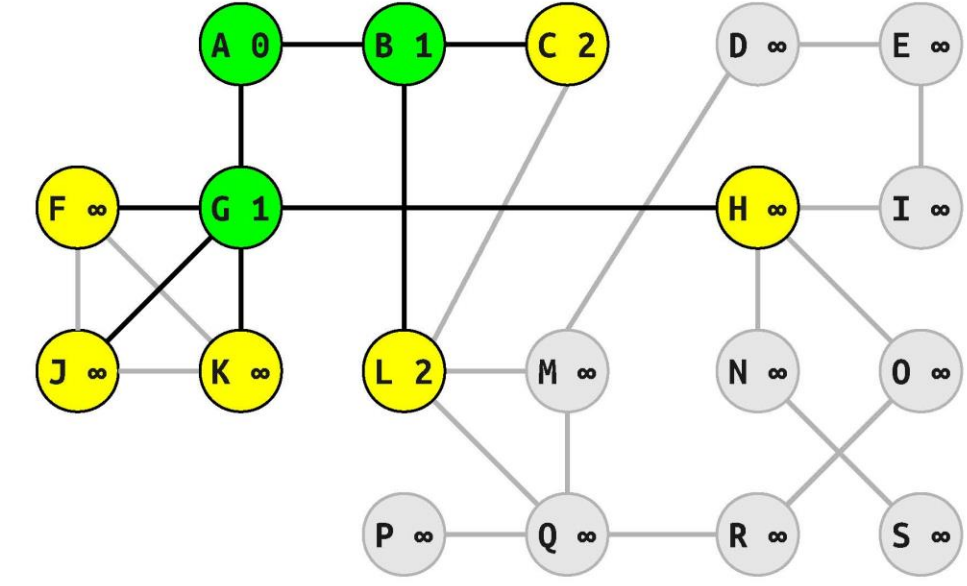


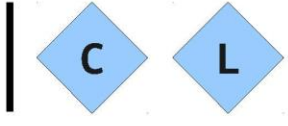
Breadth-First Search



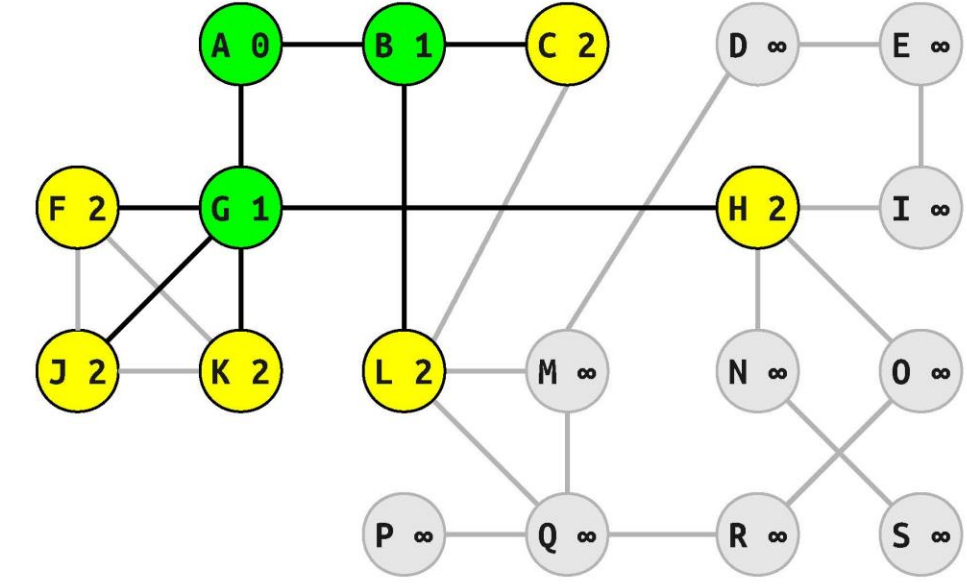


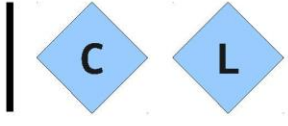
Breadth-First Search



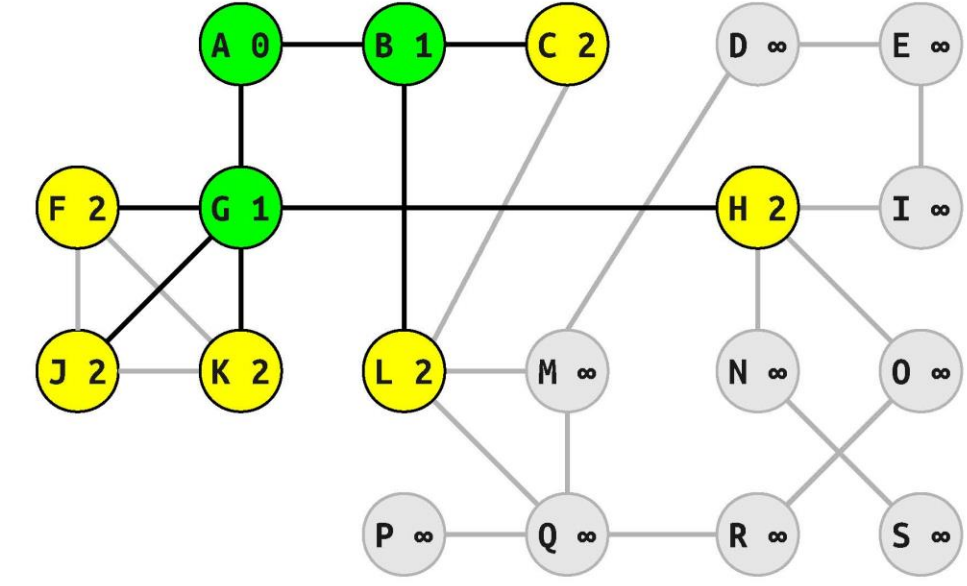


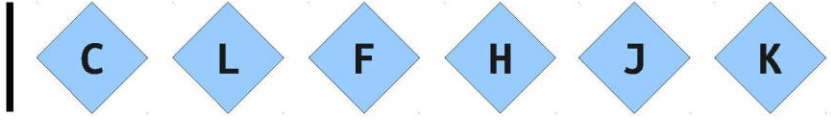
Breadth-First Search



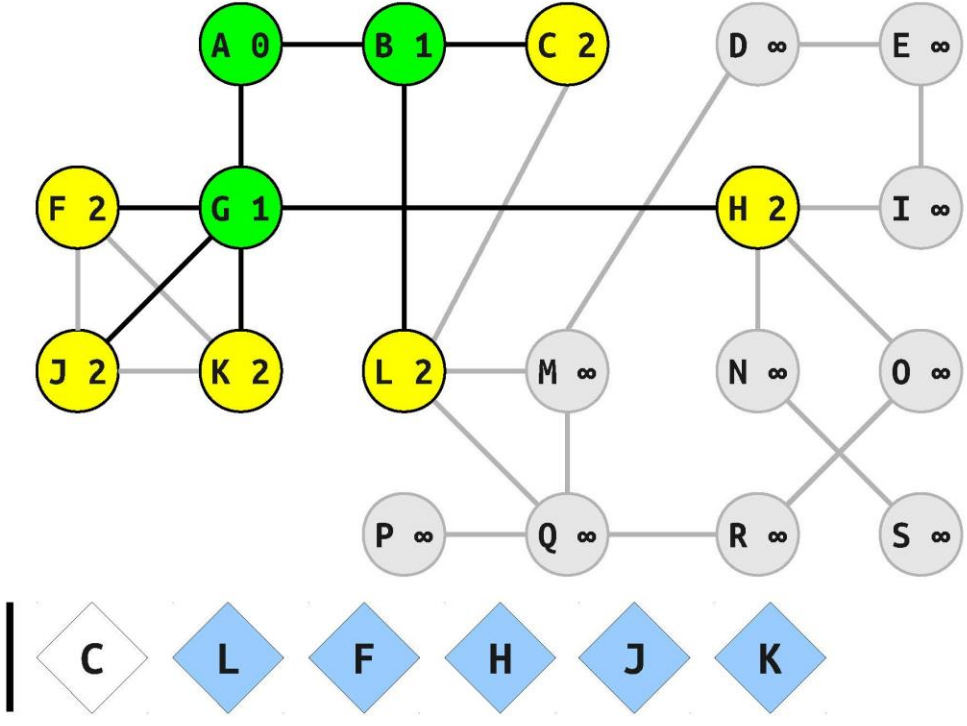


Breadth-First Search

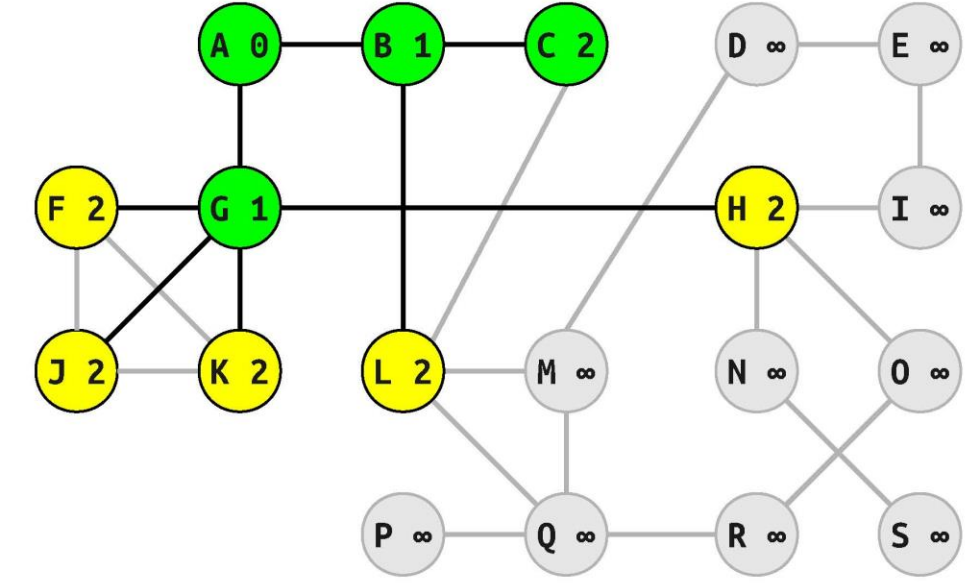




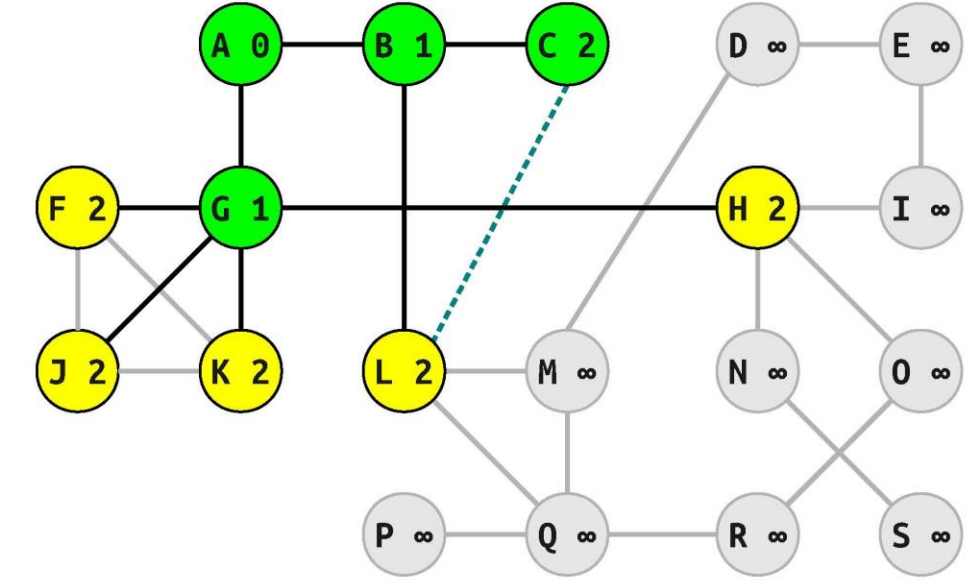
Breadth-First Search



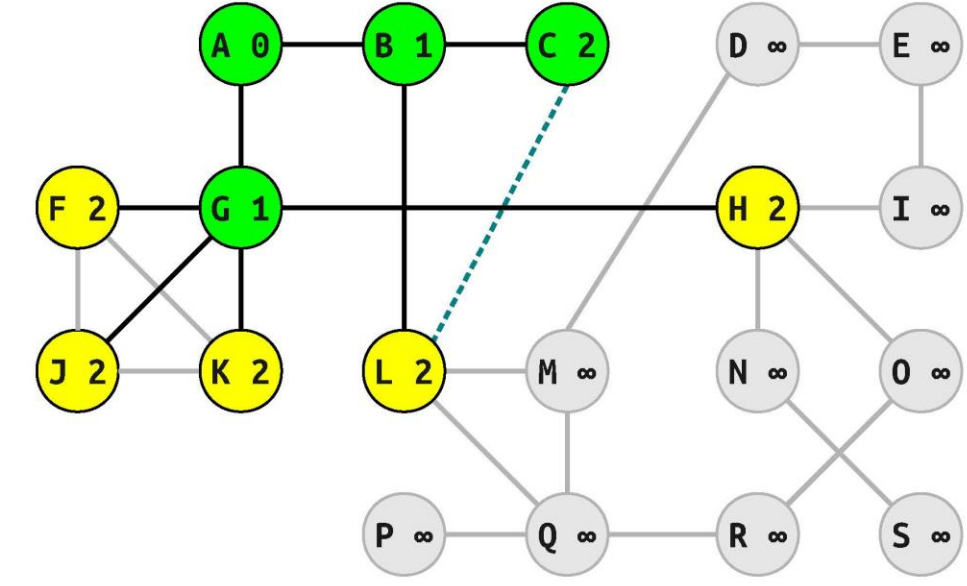
Breadth-First Search

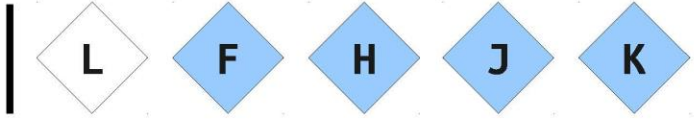


Breadth-First Search

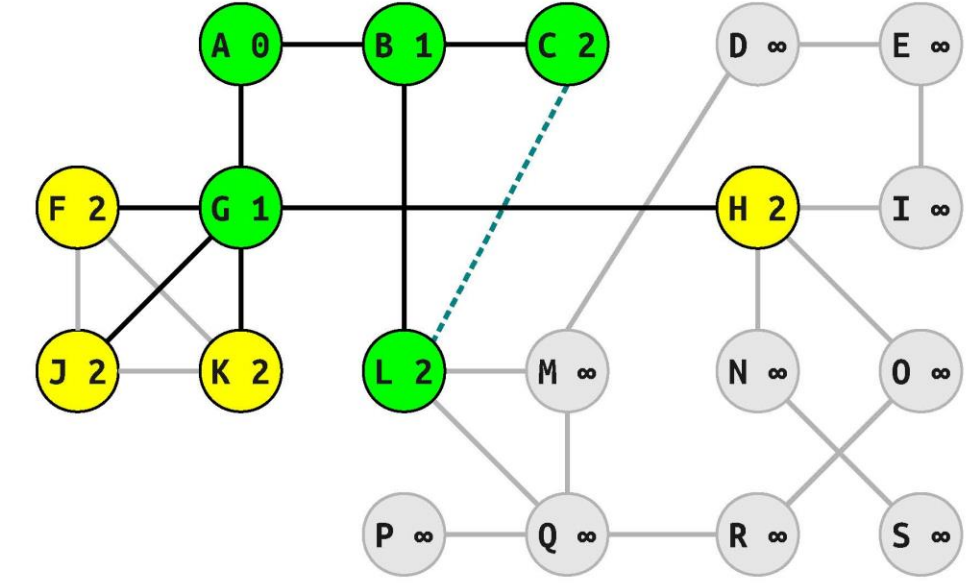


Breadth-First Search

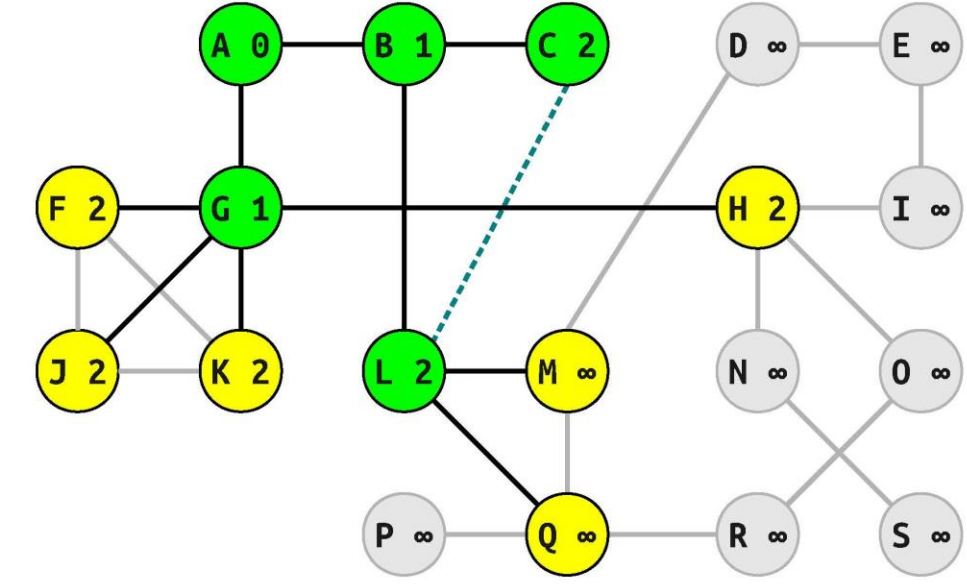




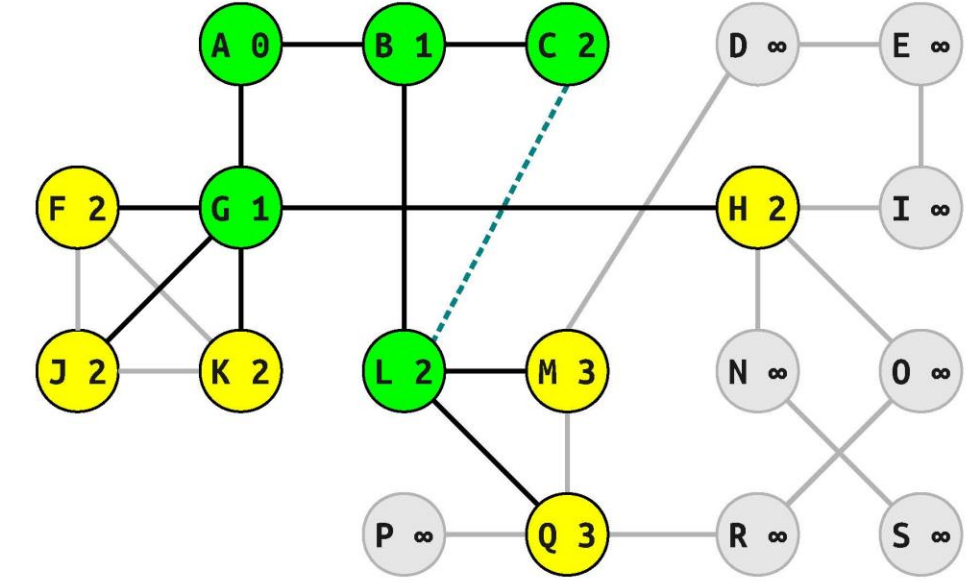
Breadth-First Search



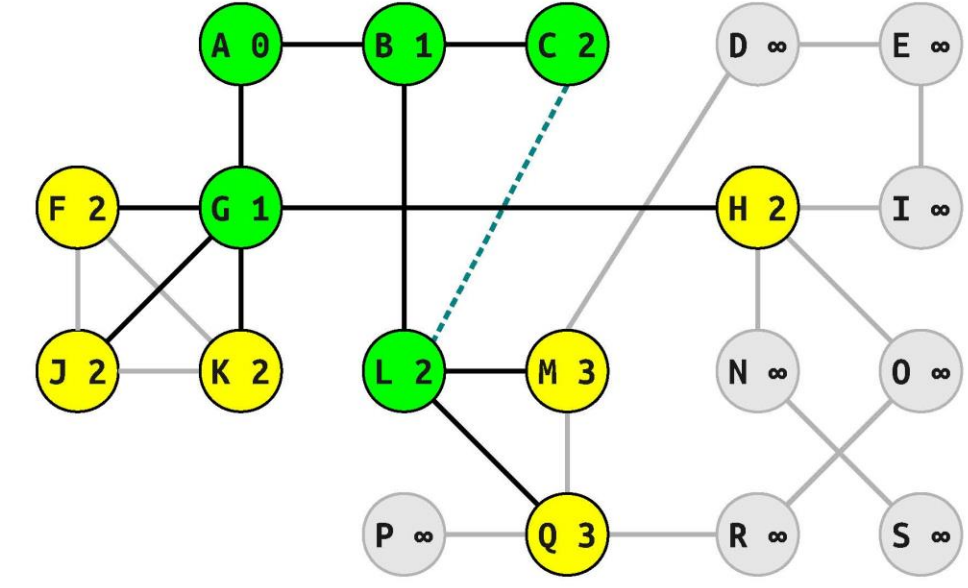
Breadth-First Search

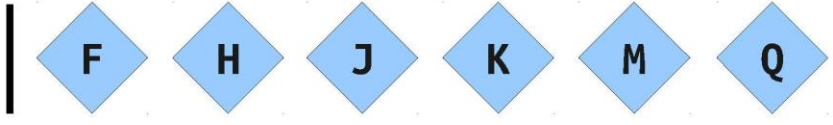


Breadth-First Search

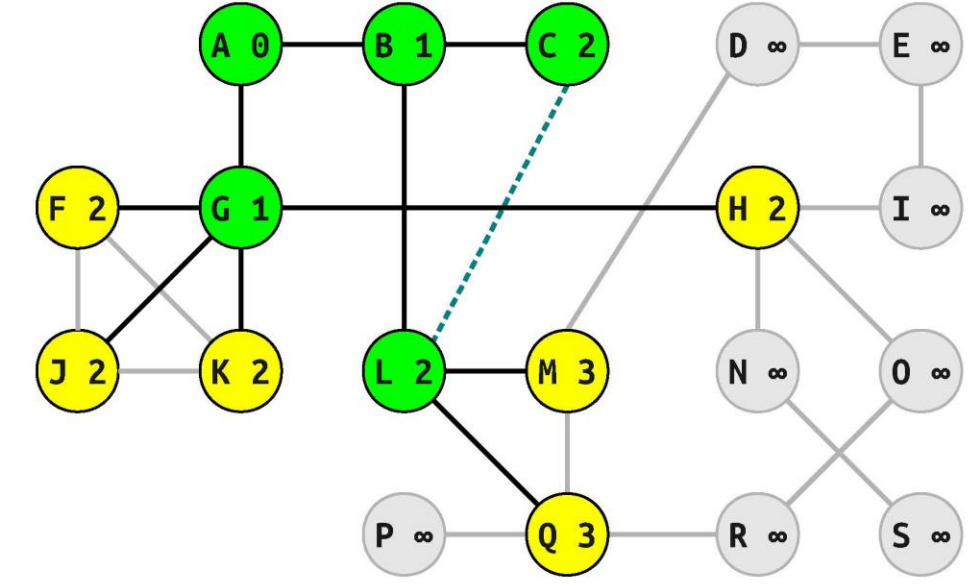


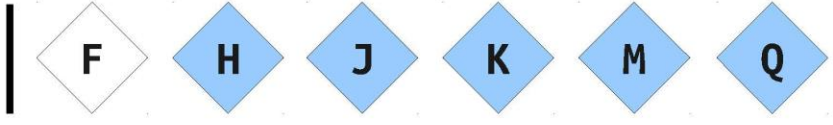
Breadth-First Search



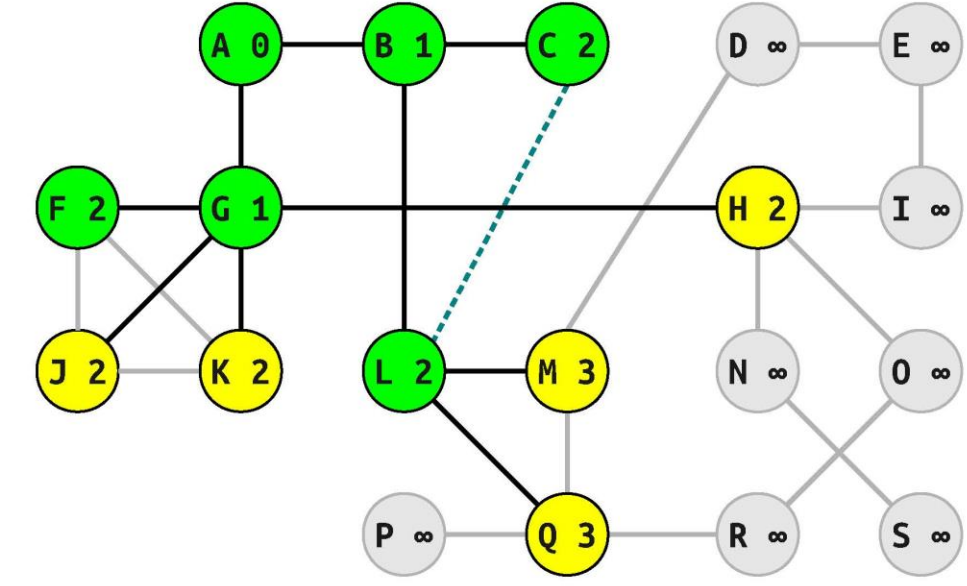


Breadth-First Search

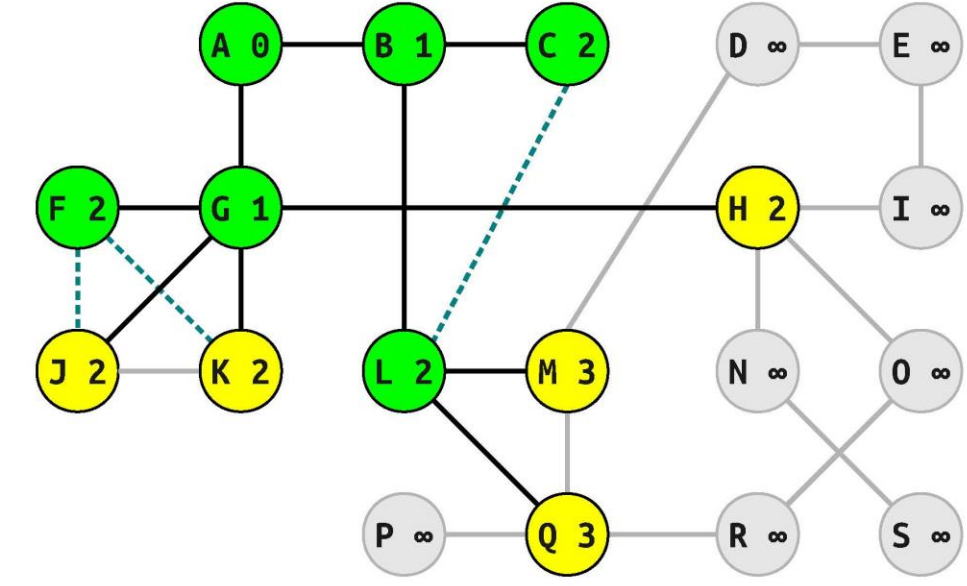




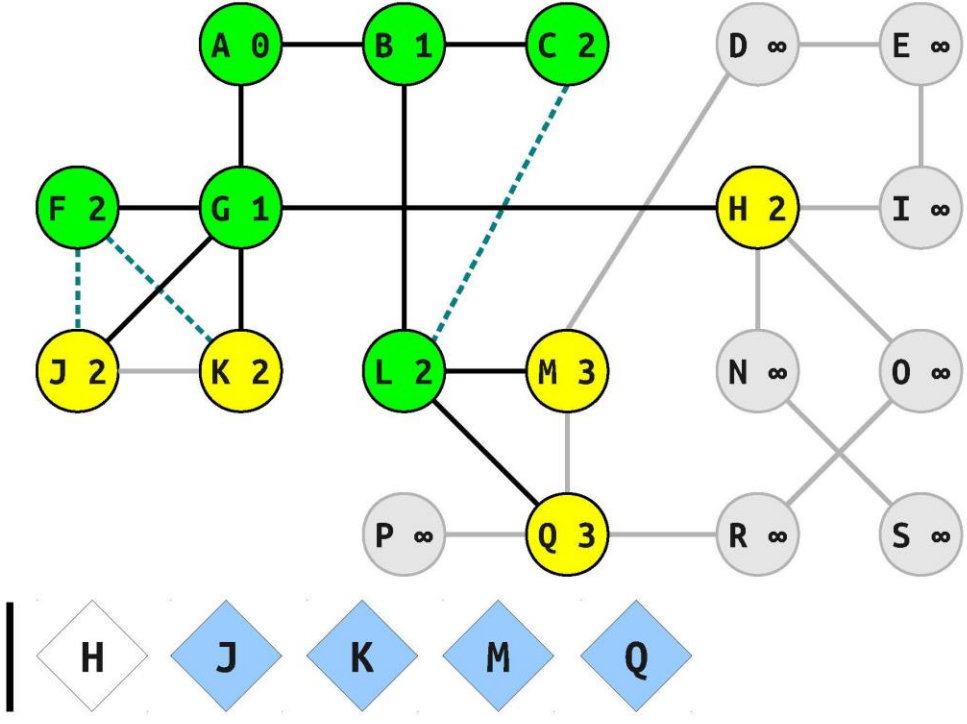
Breadth-First Search



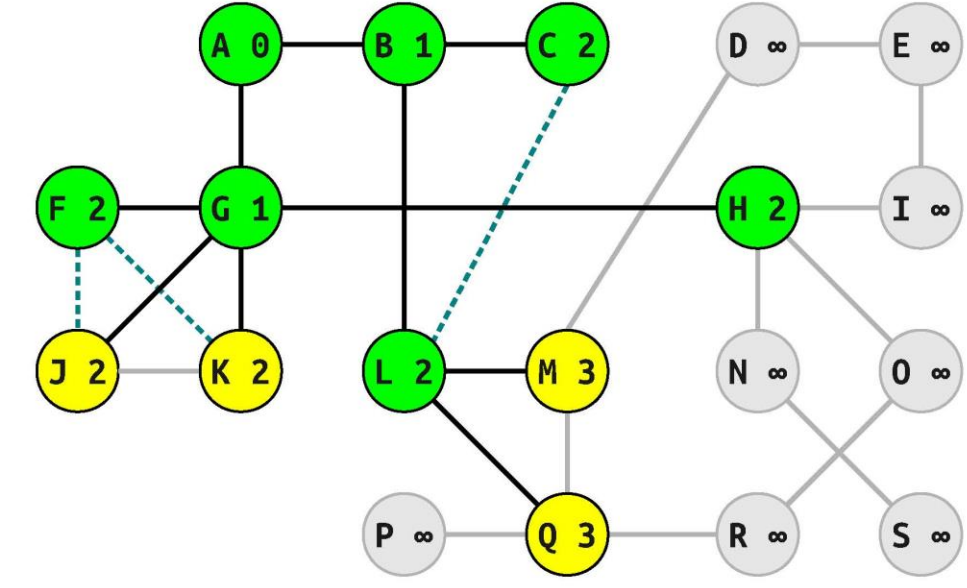
Breadth-First Search

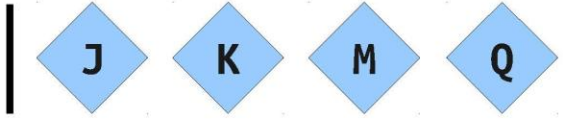


Breadth-First Search

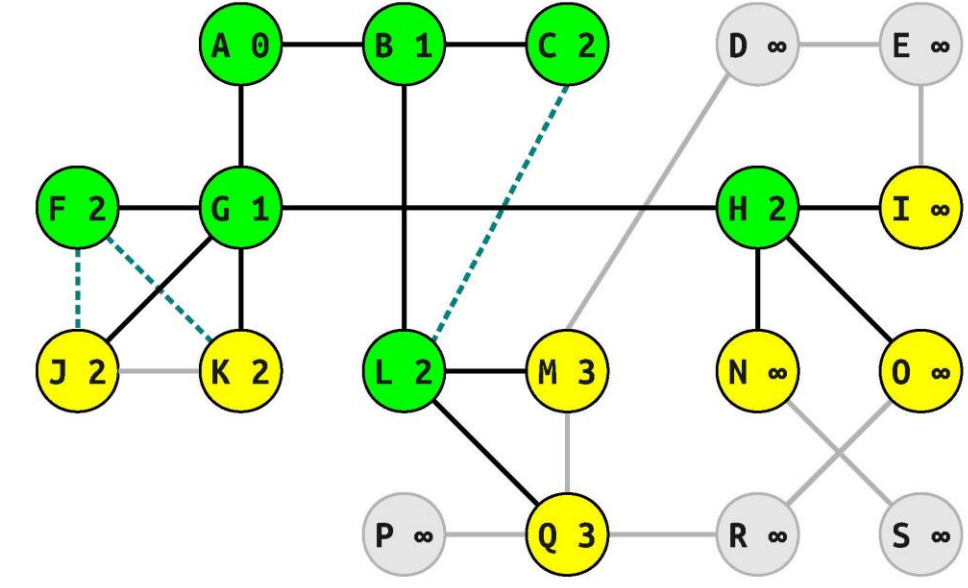


Breadth-First Search

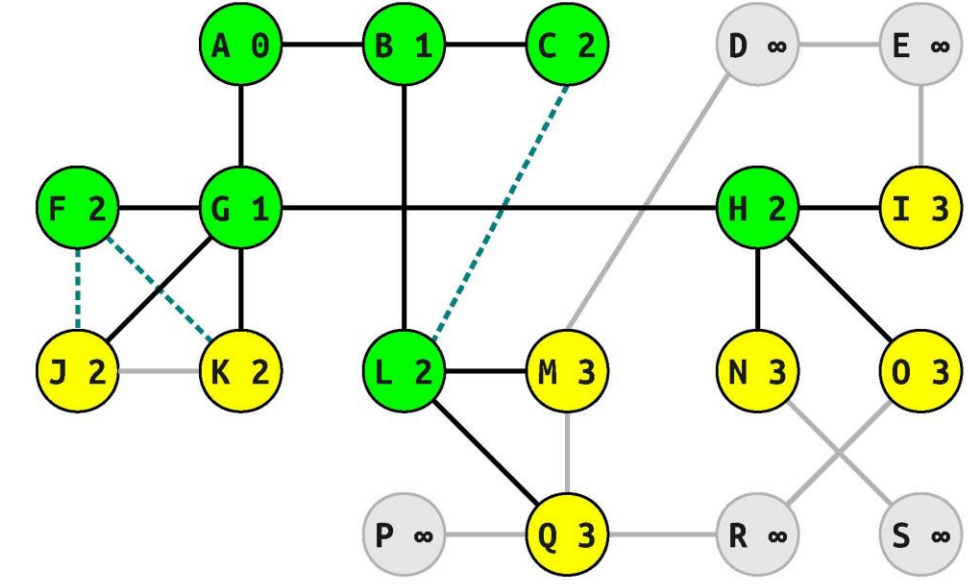




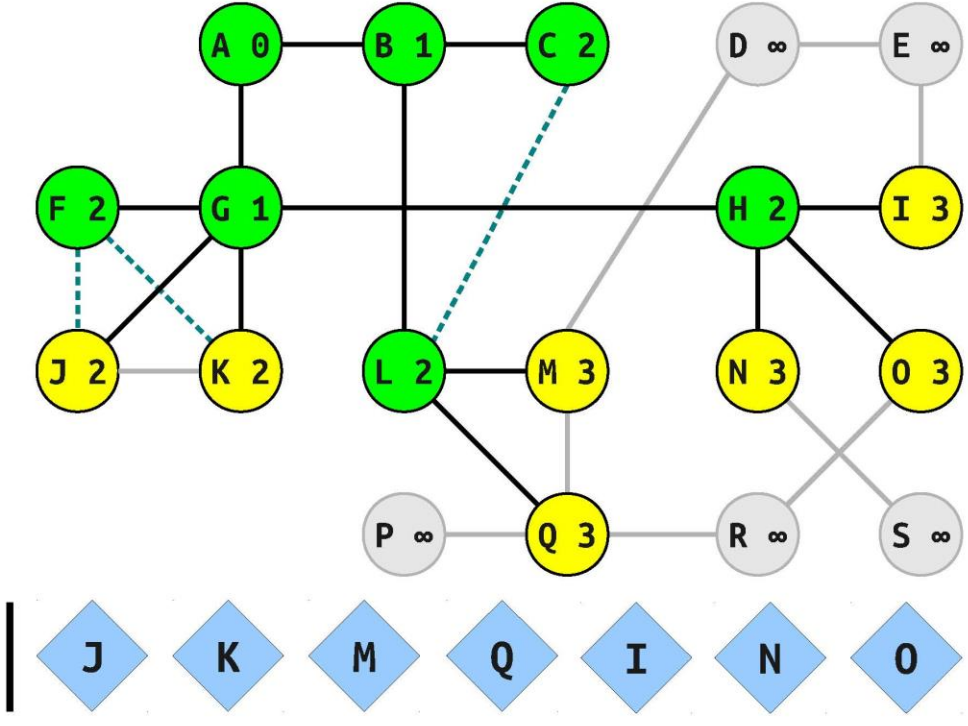
Breadth-First Search



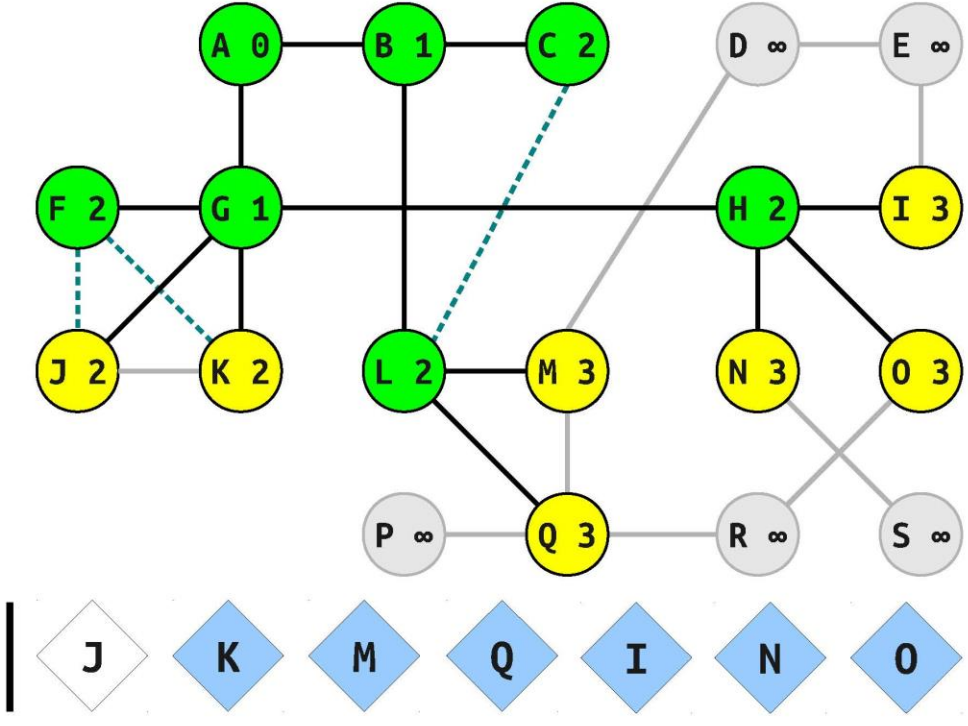
Breadth-First Search



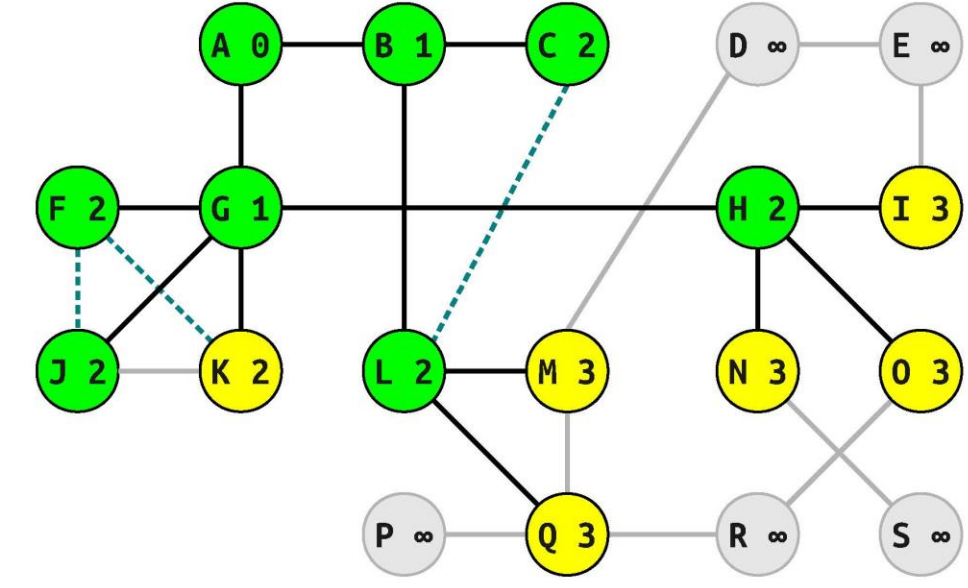
Breadth-First Search

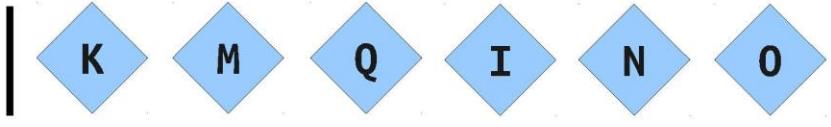


Breadth-First Search

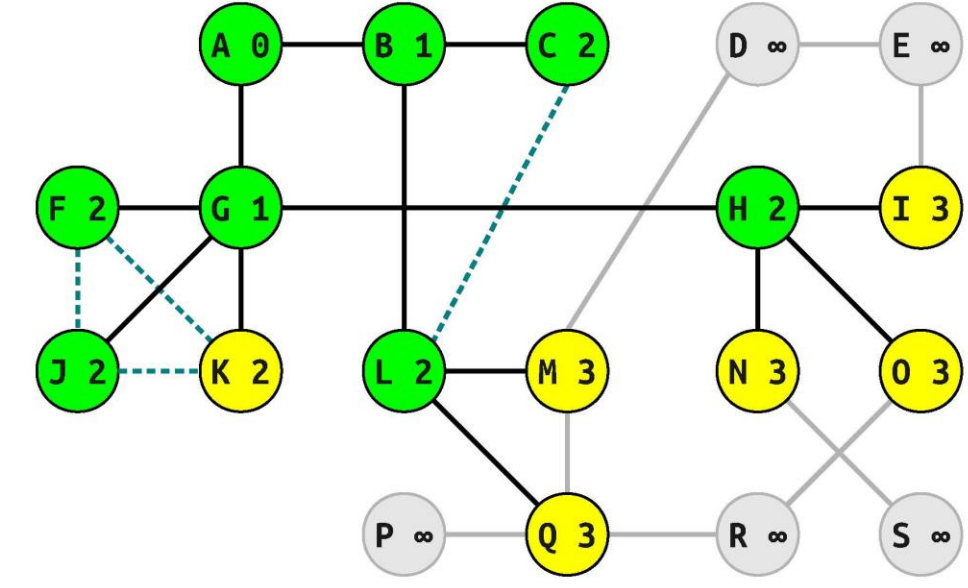


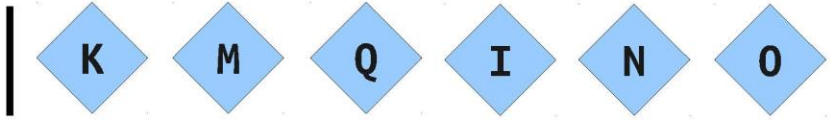
Breadth-First Search



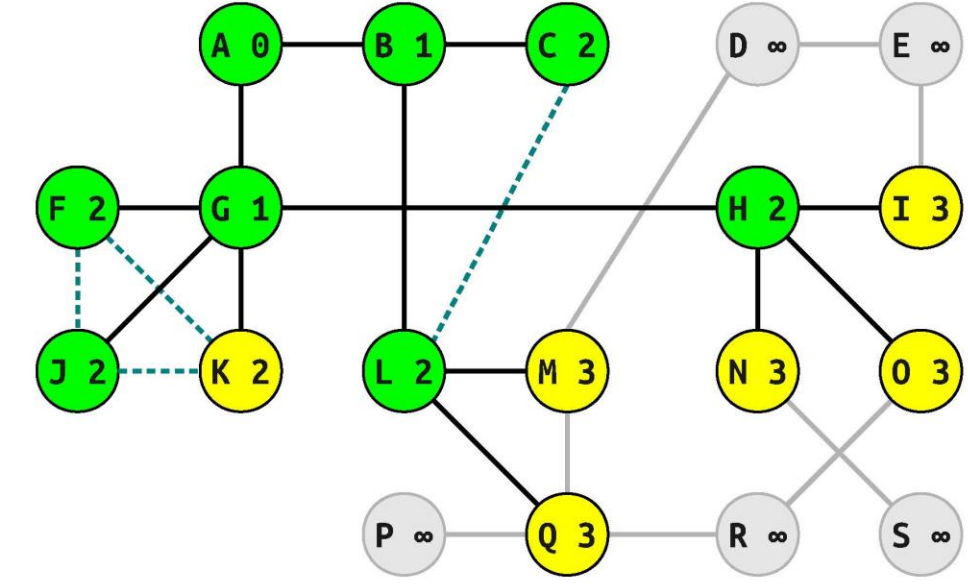


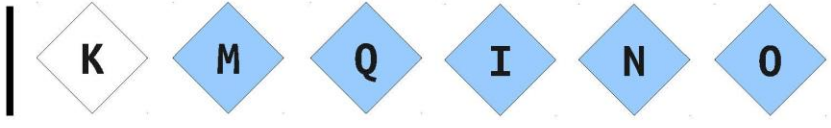
Breadth-First Search



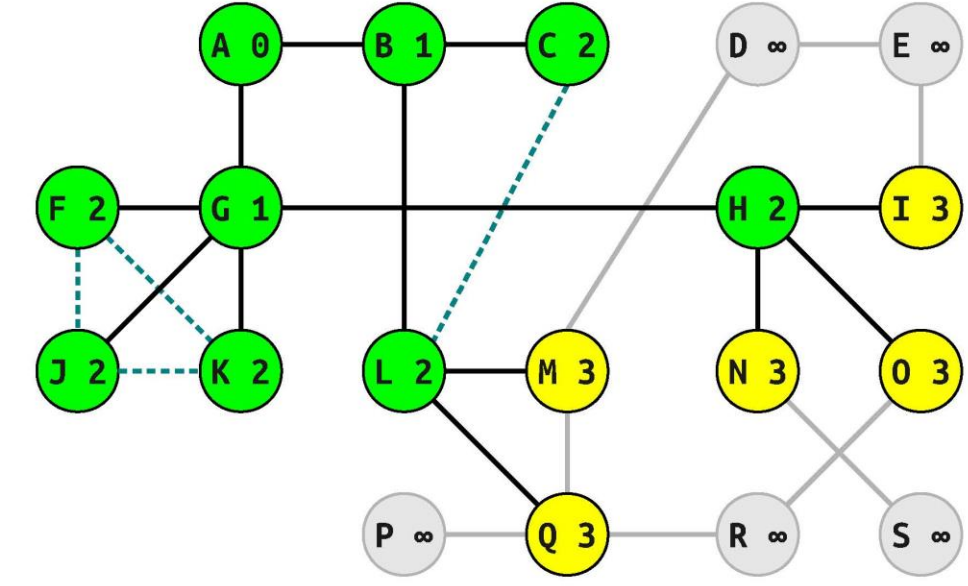


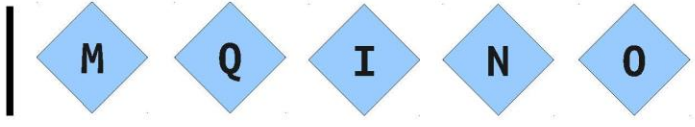
Breadth-First Search



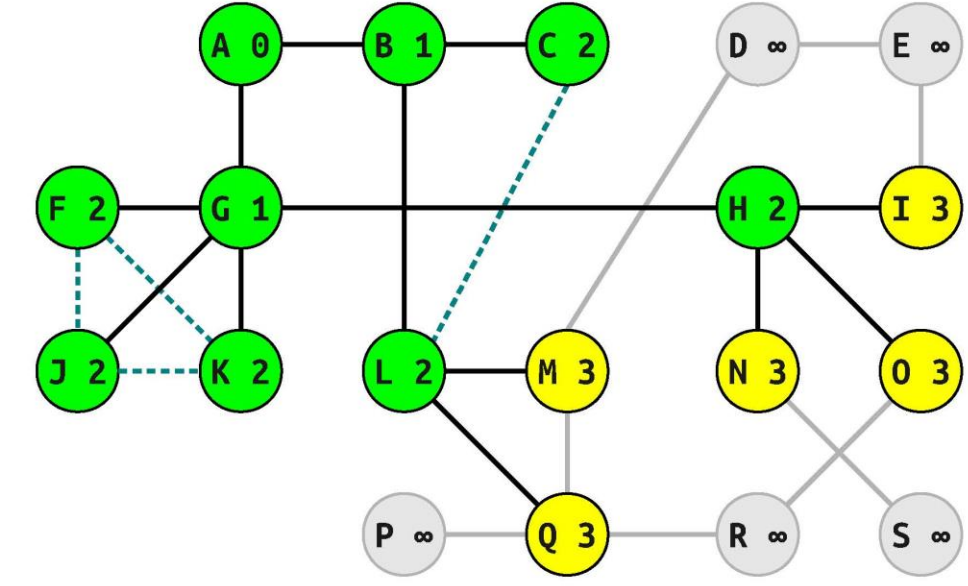


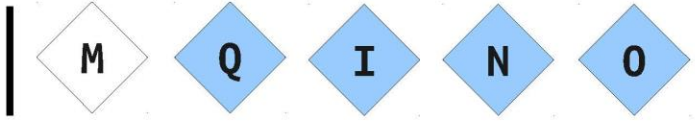
Breadth-First Search



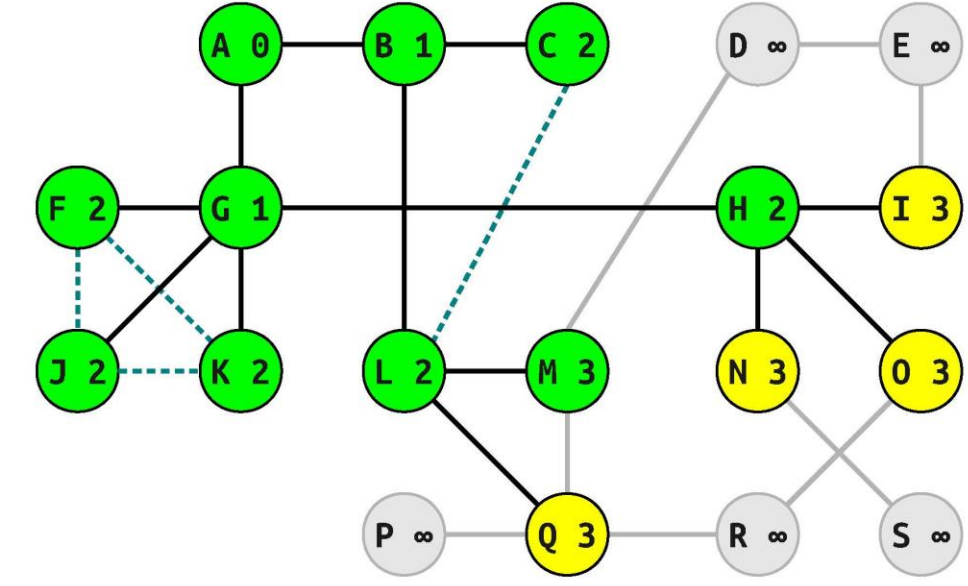


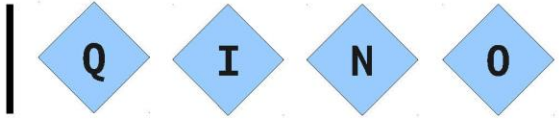
Breadth-First Search



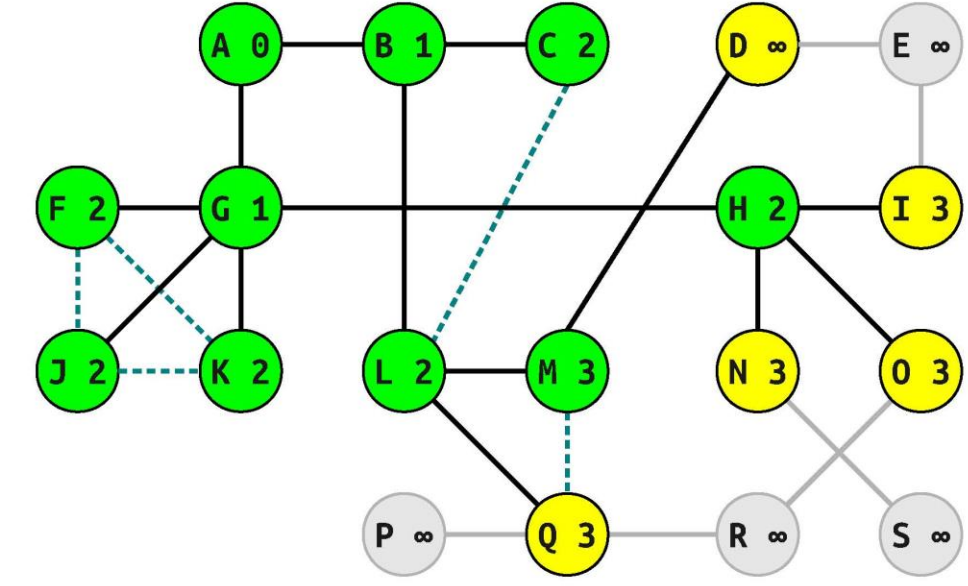


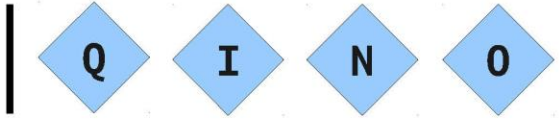
Breadth-First Search



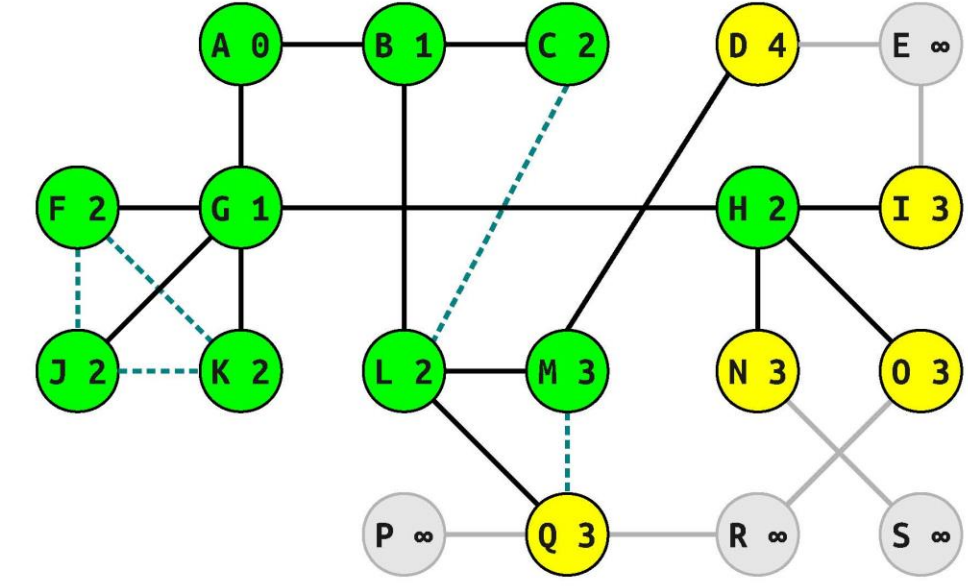


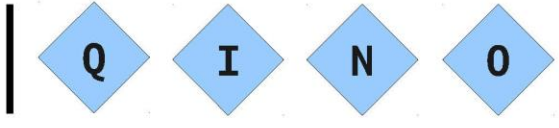
Breadth-First Search



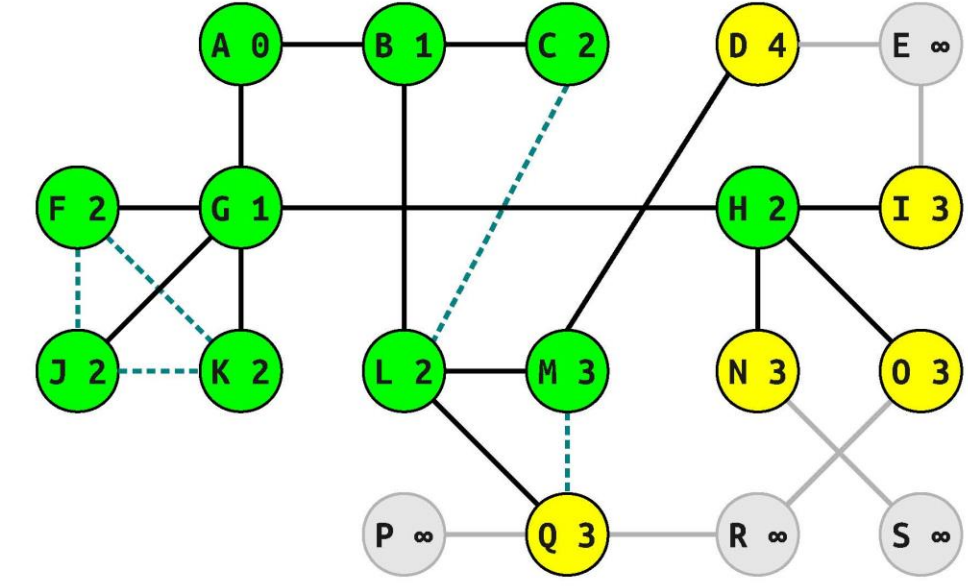


Breadth-First Search

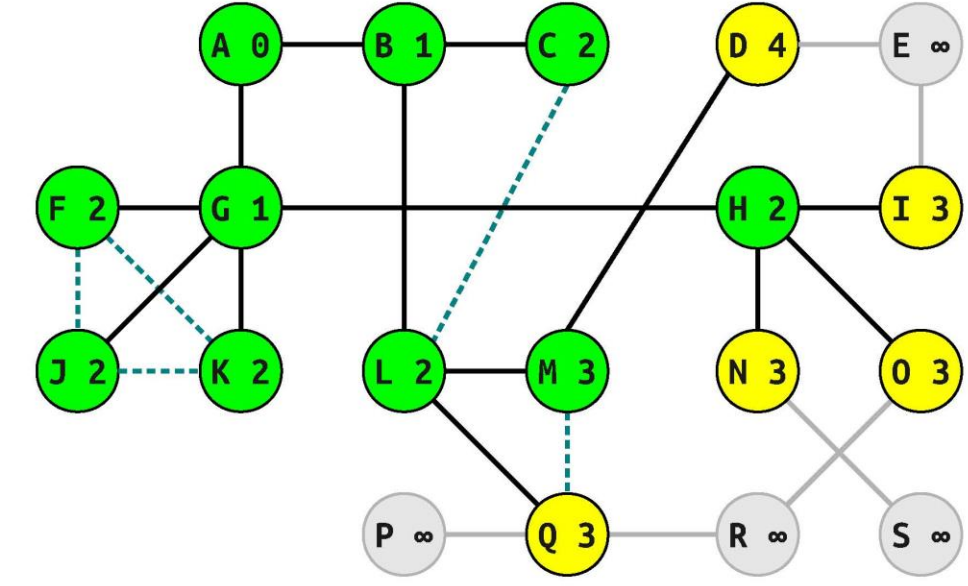




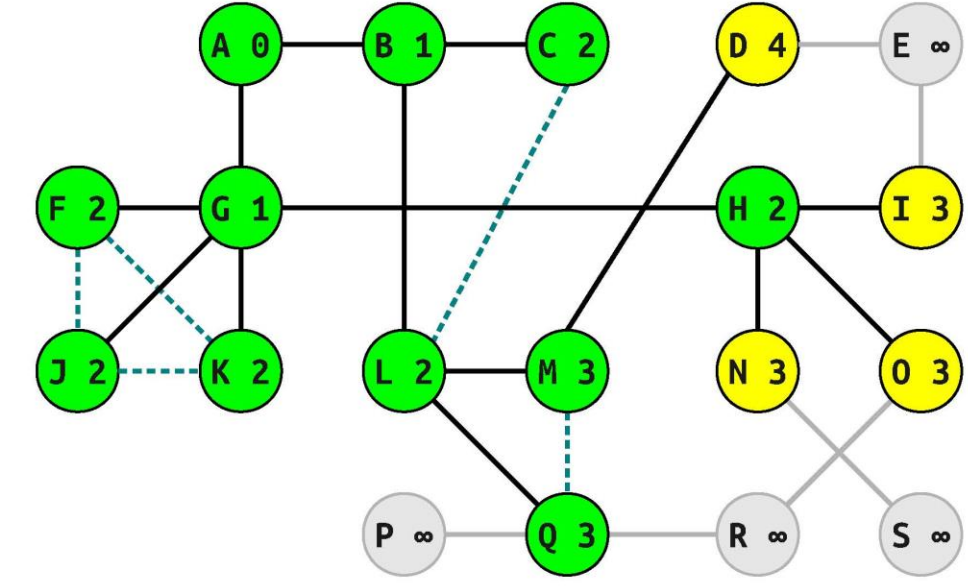
Breadth-First Search

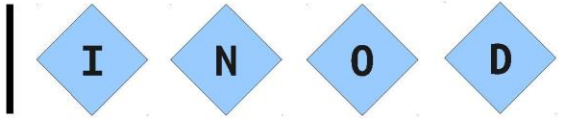


Breadth-First Search

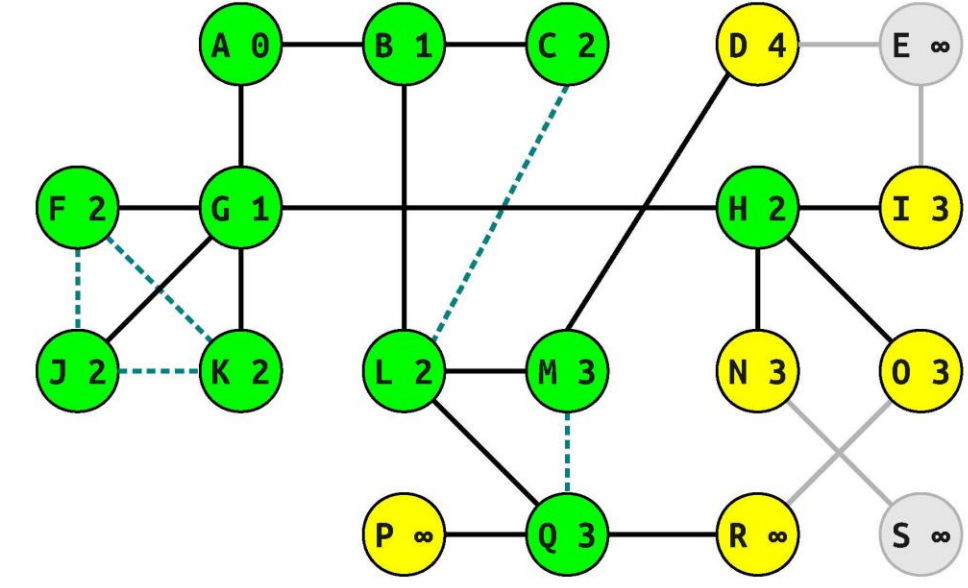


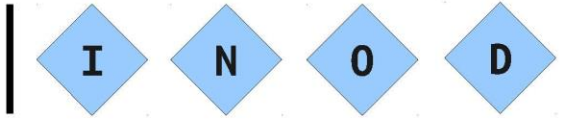
Breadth-First Search



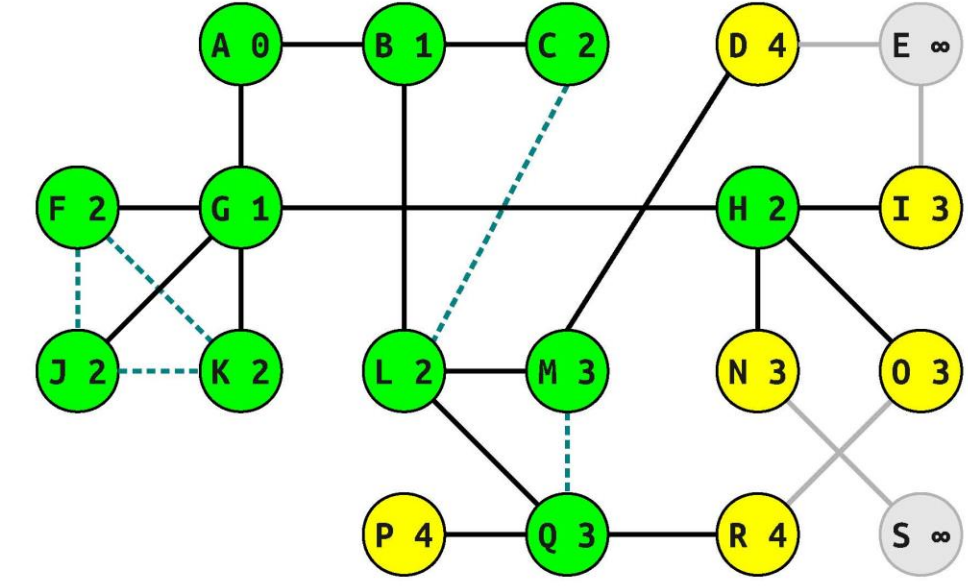


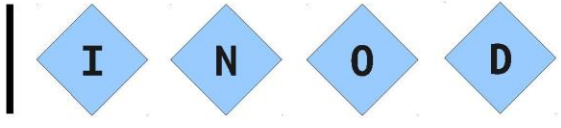
Breadth-First Search



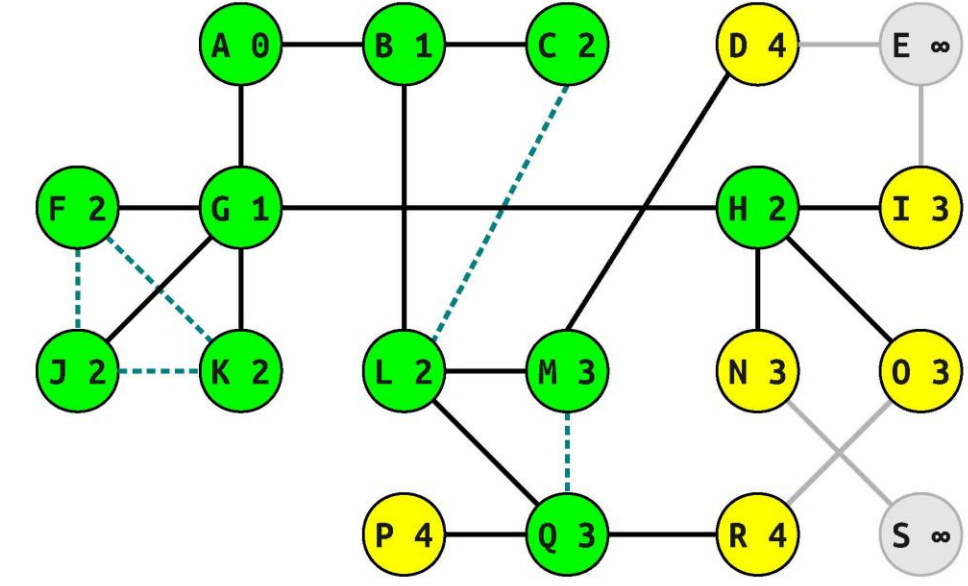


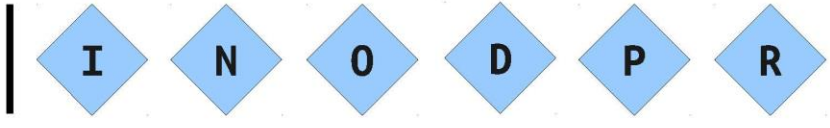
Breadth-First Search



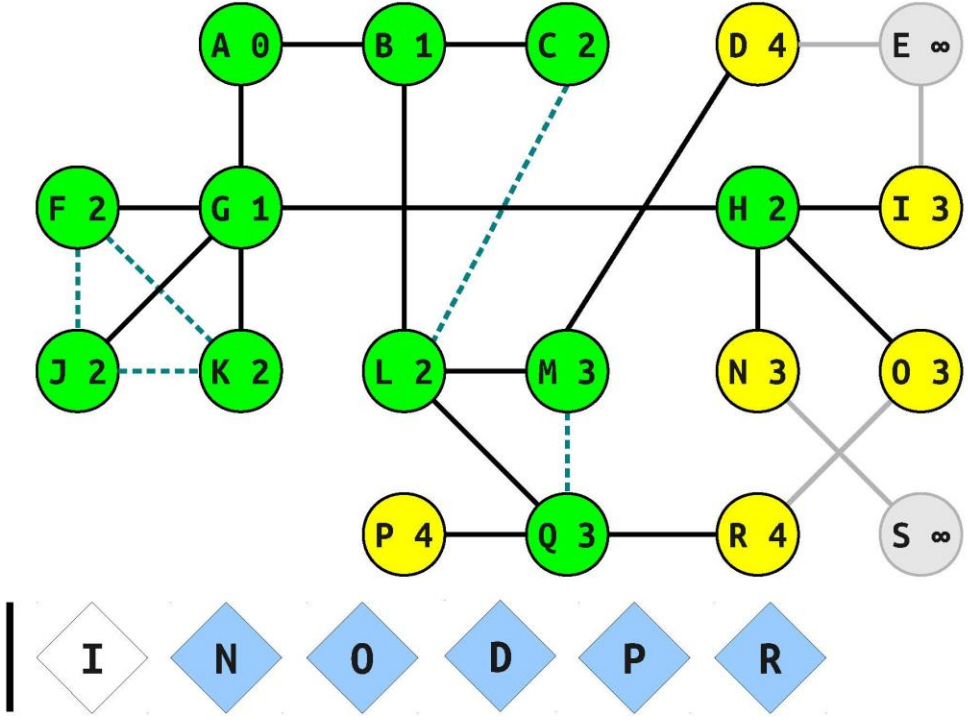


Breadth-First Search

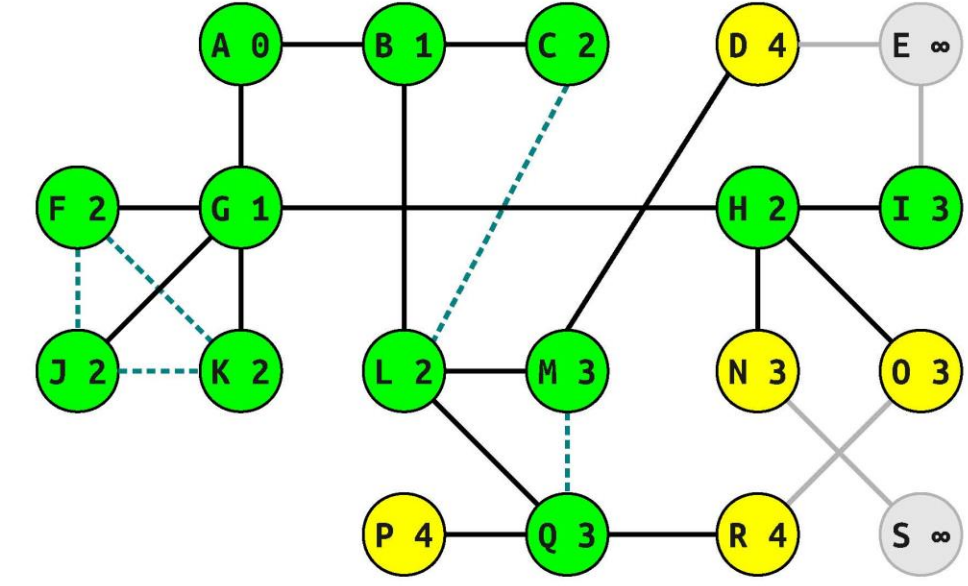


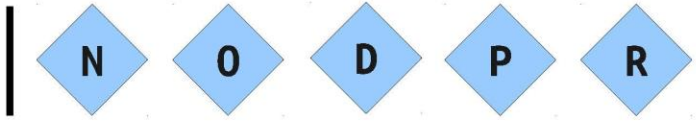


Breadth-First Search

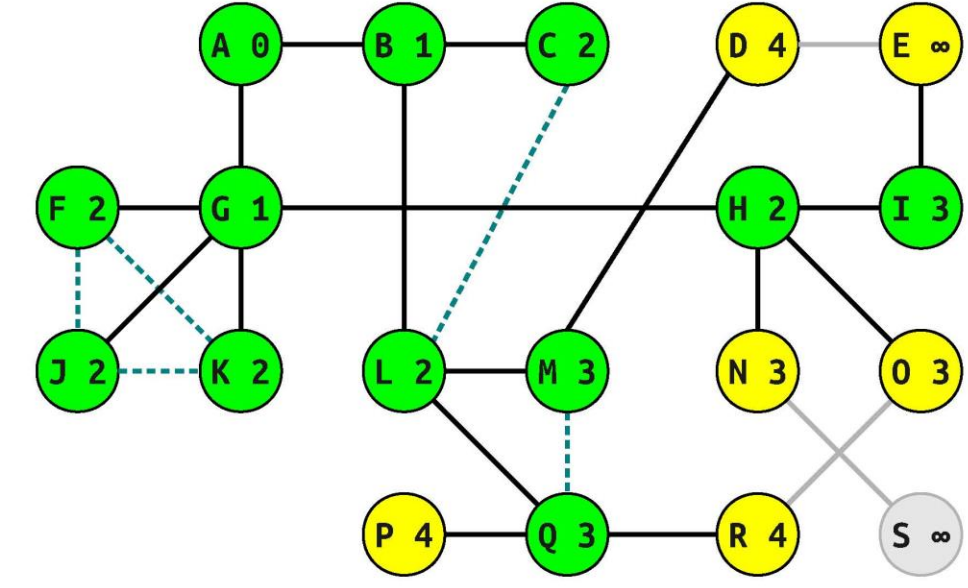


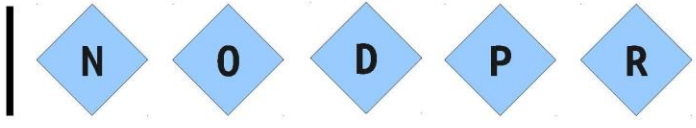
Breadth-First Search



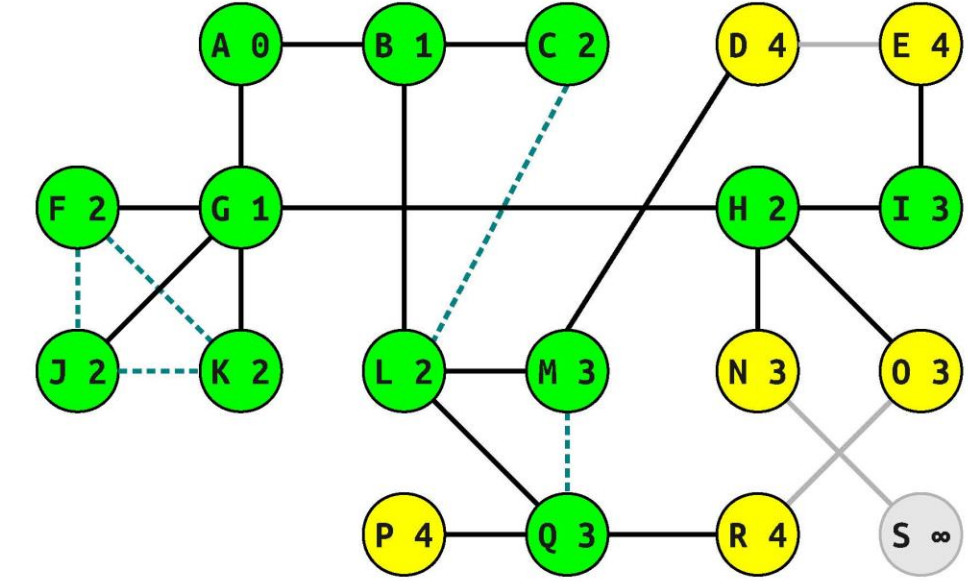


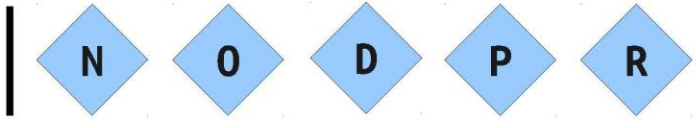
Breadth-First Search



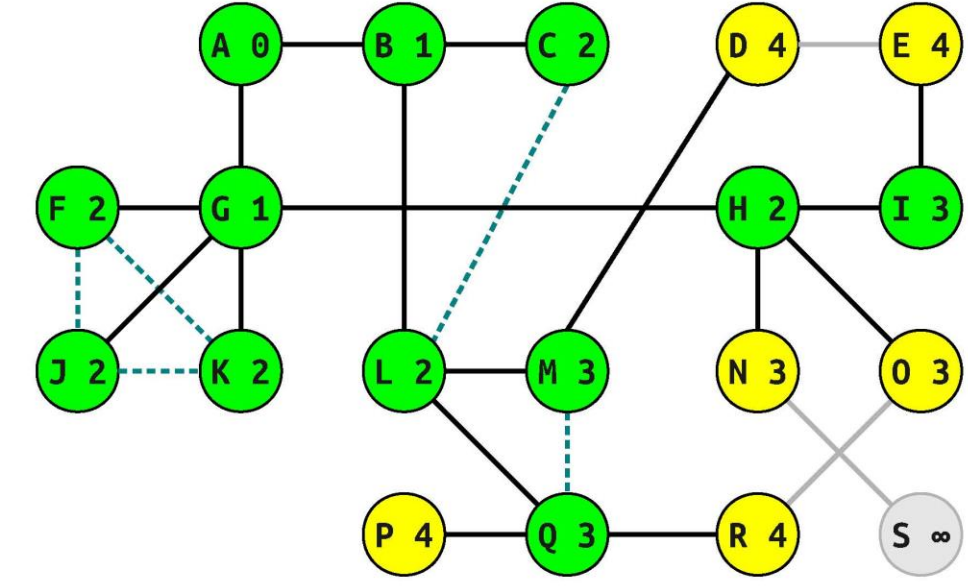


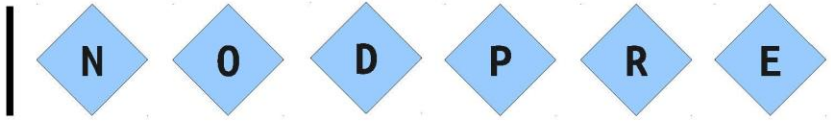
Breadth-First Search



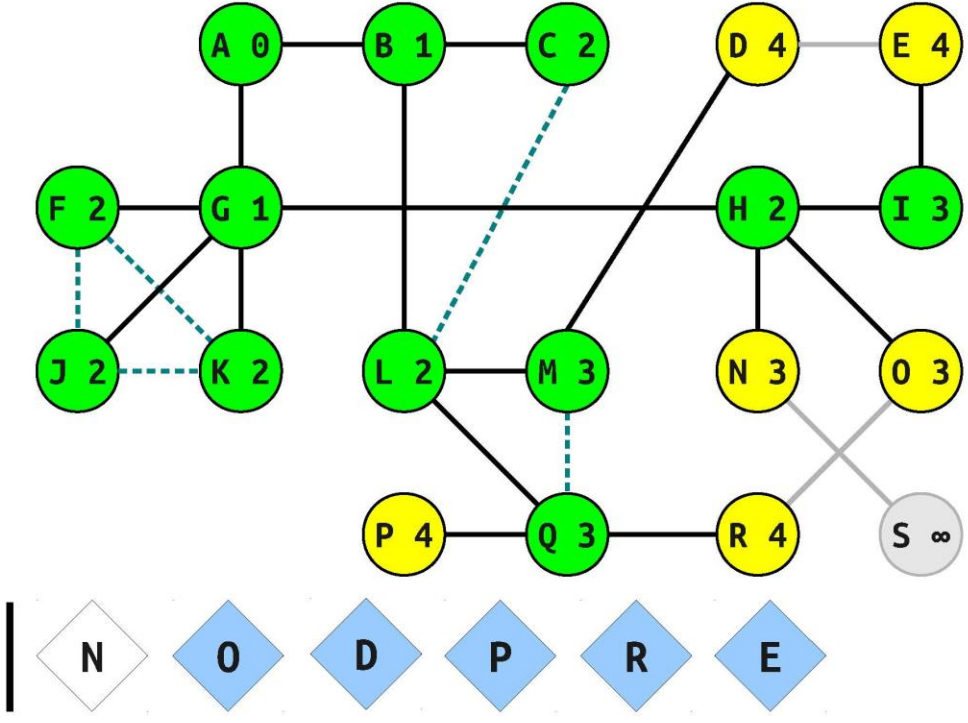


Breadth-First Search

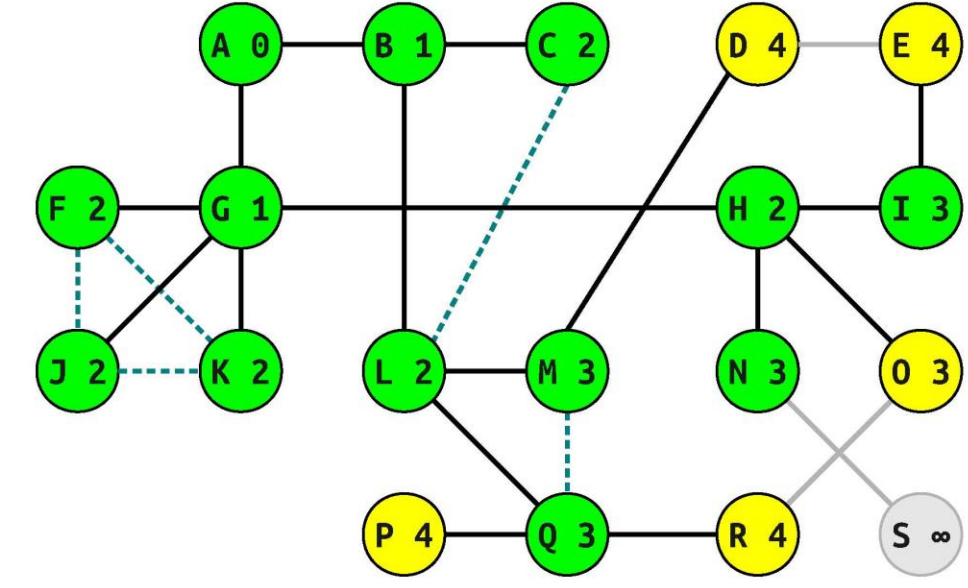


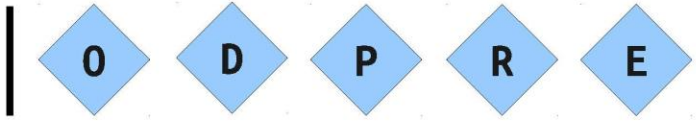


Breadth-First Search

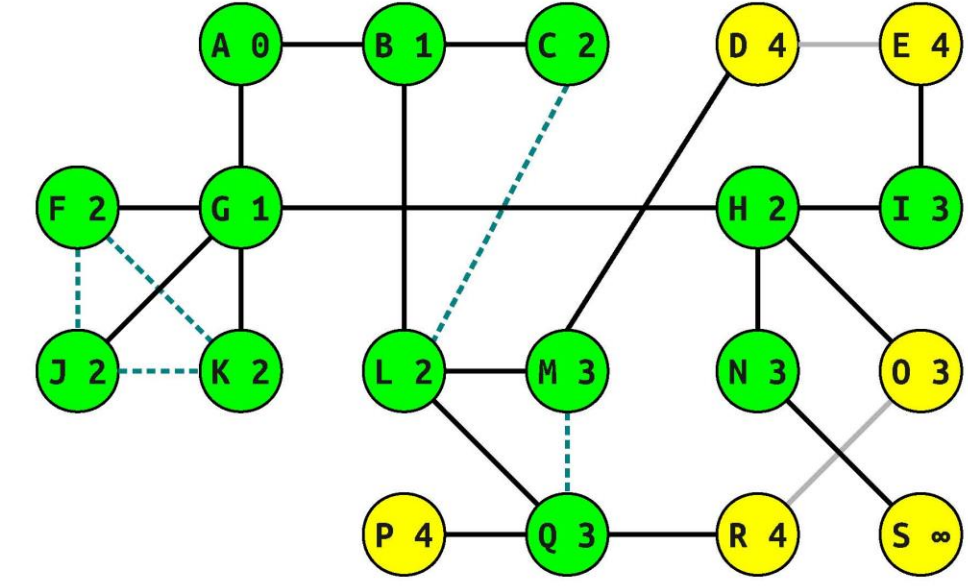


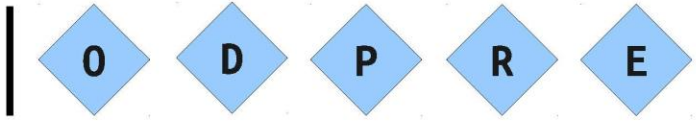
Breadth-First Search



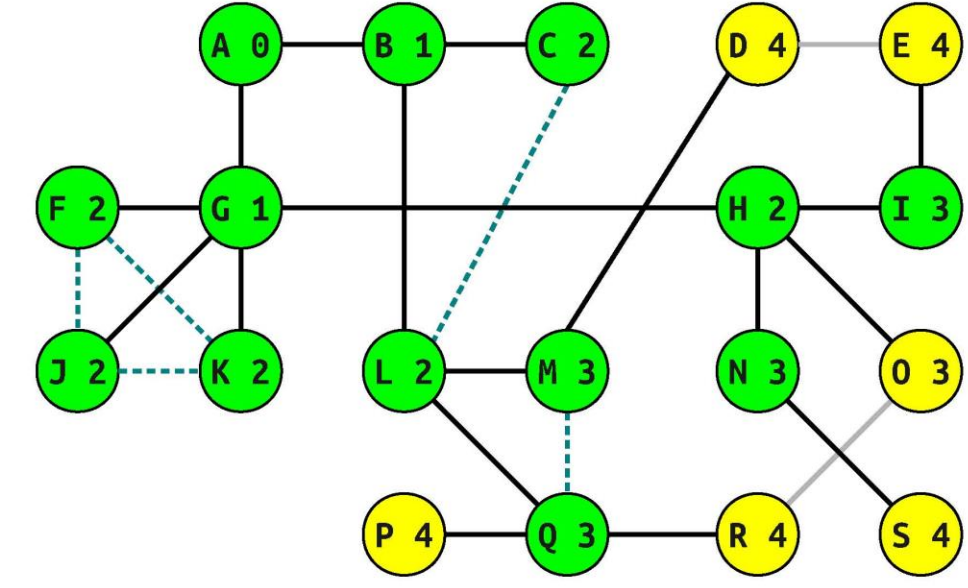


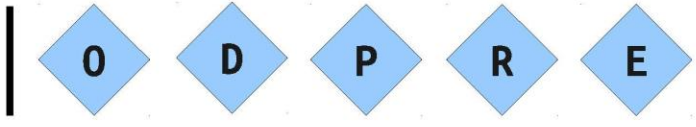
Breadth-First Search



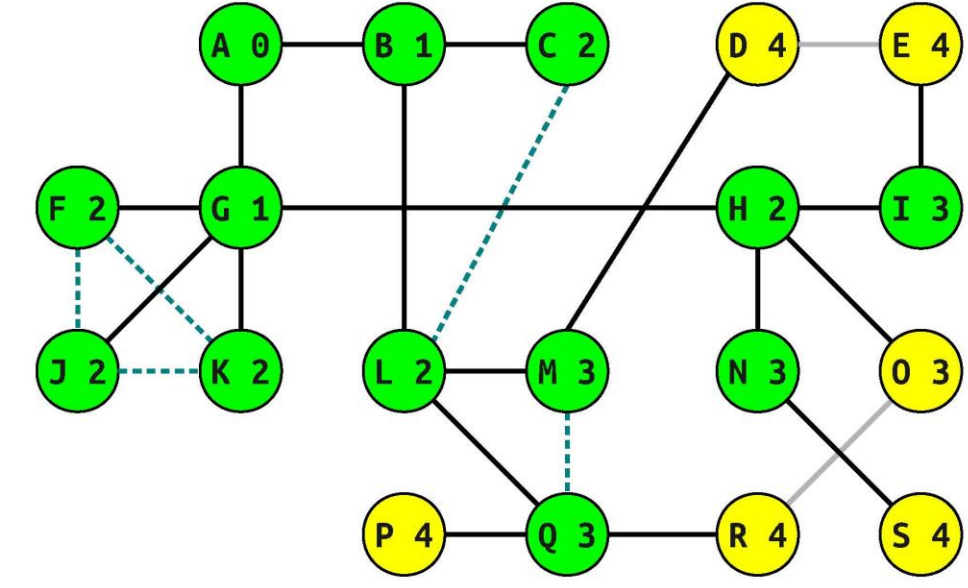


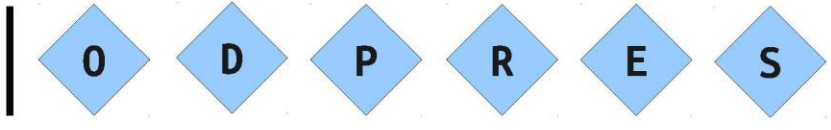
Breadth-First Search



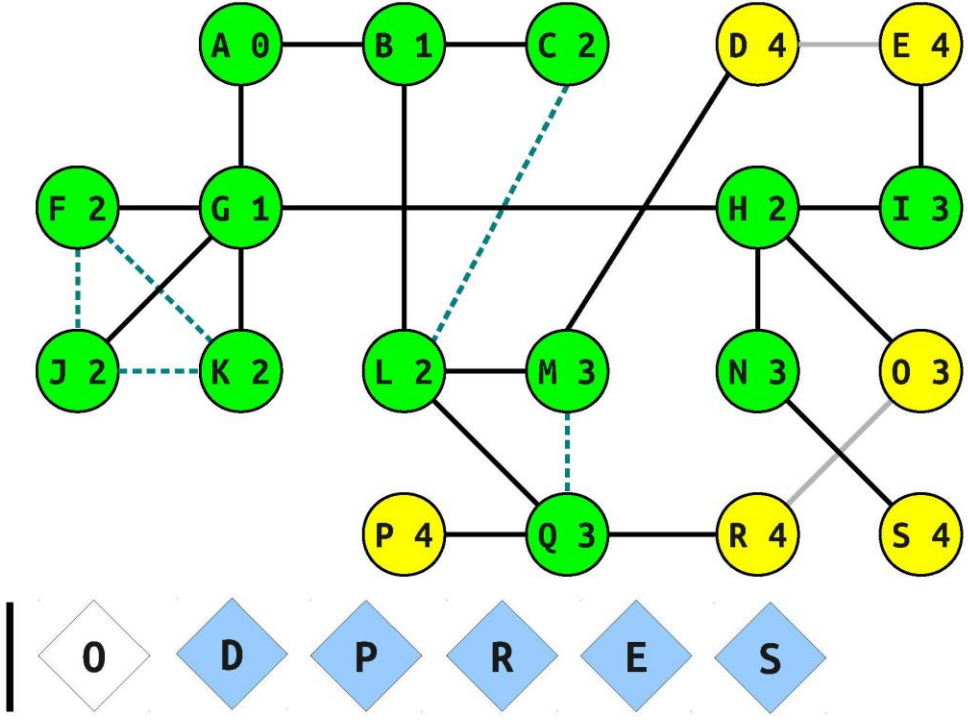


Breadth-First Search

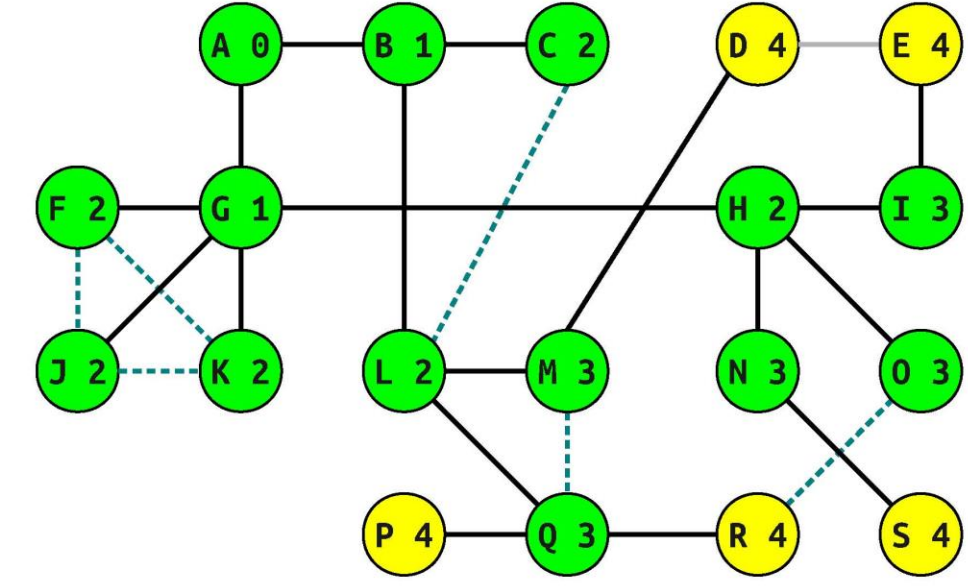


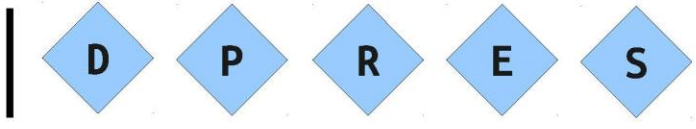


Breadth-First Search

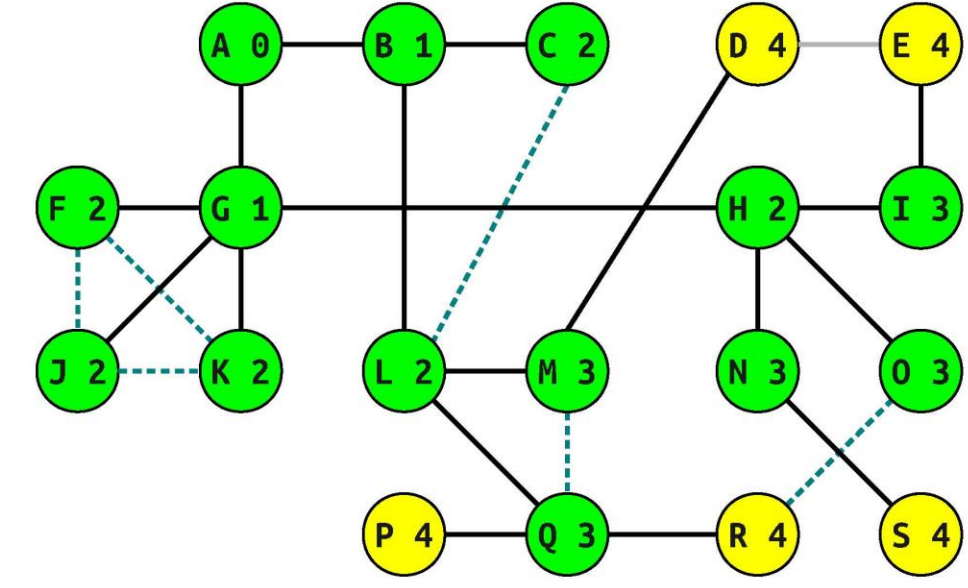


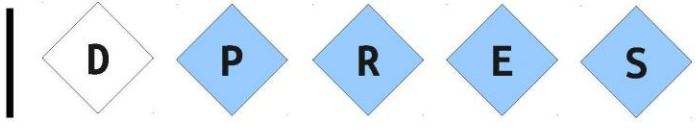
Breadth-First Search



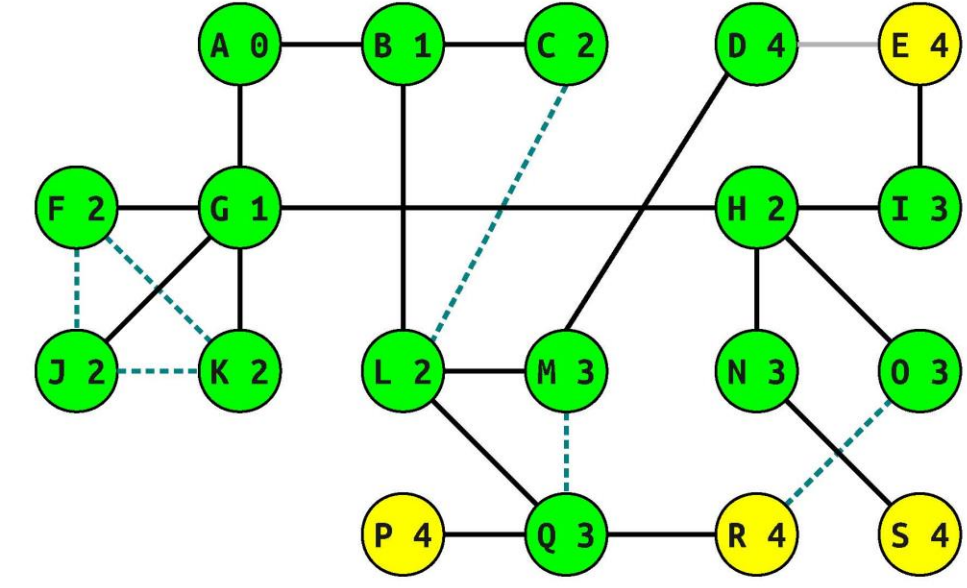


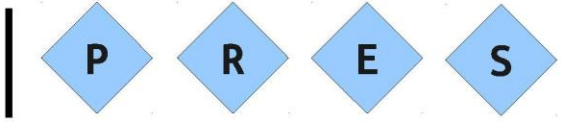
Breadth-First Search



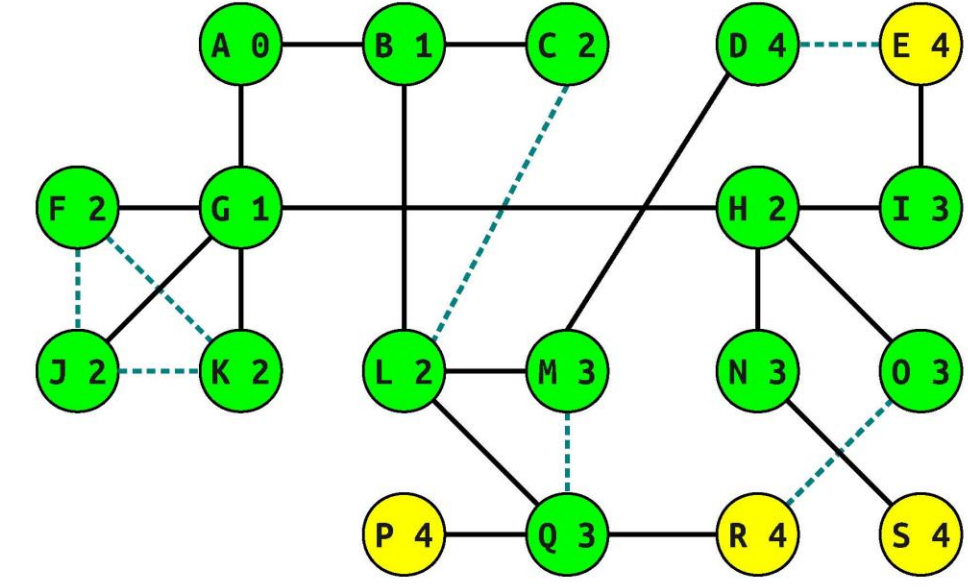


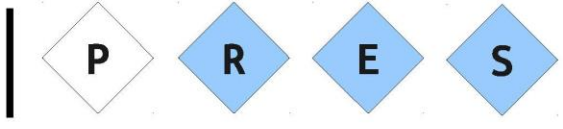
Breadth-First Search



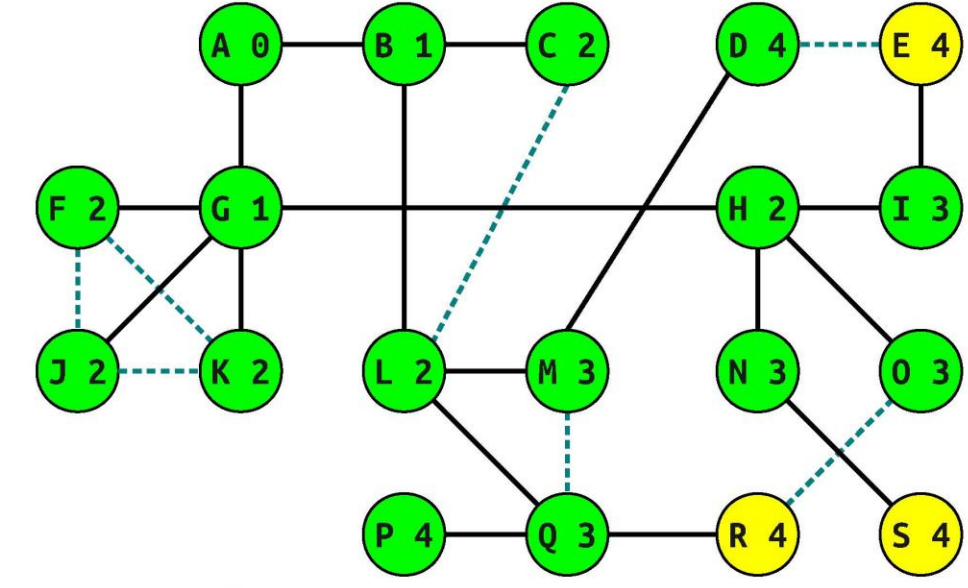


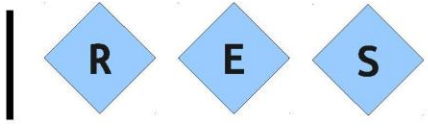
Breadth-First Search



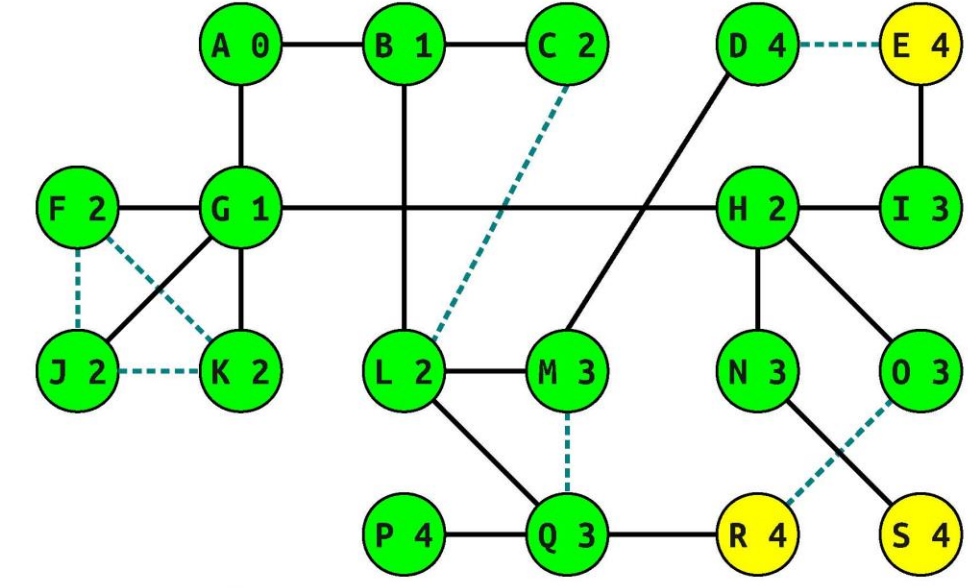


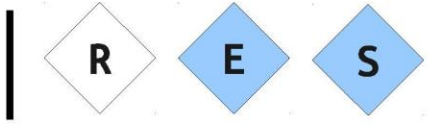
Breadth-First Search



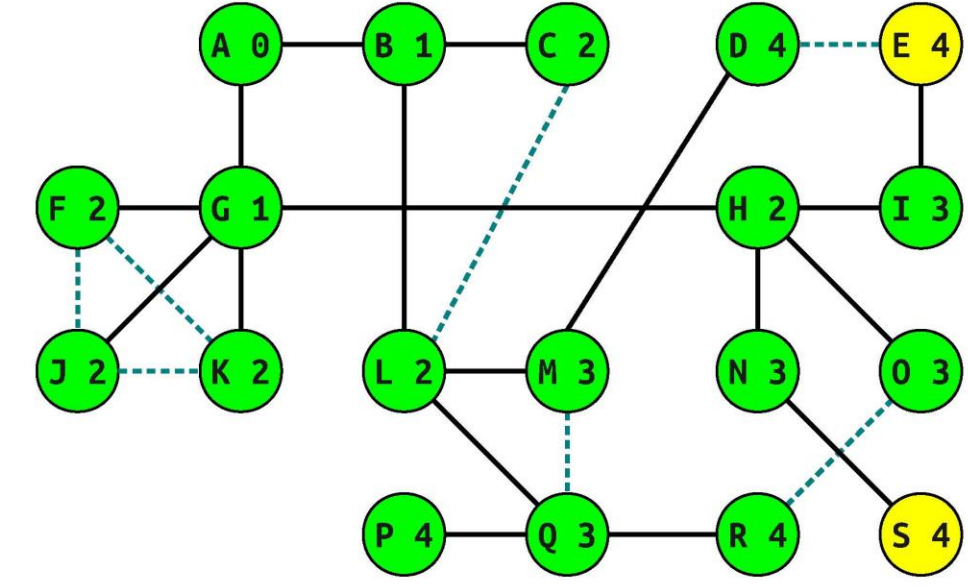


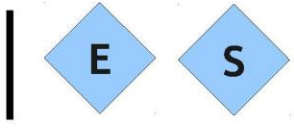
Breadth-First Search



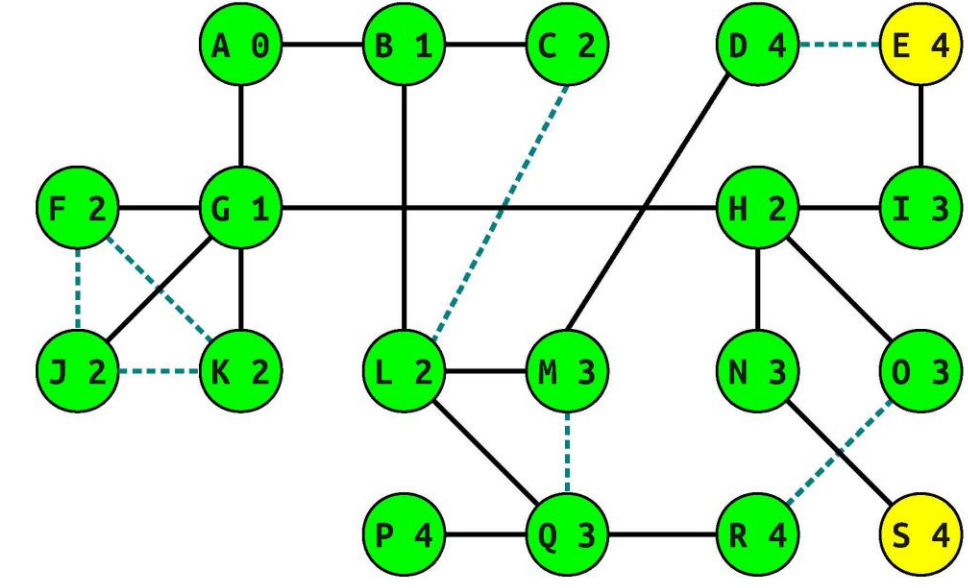


Breadth-First Search

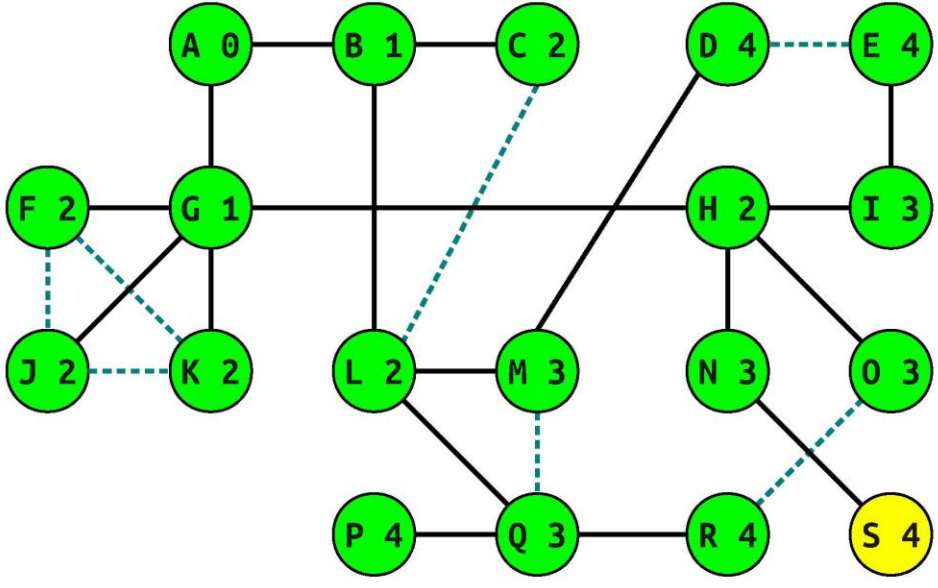




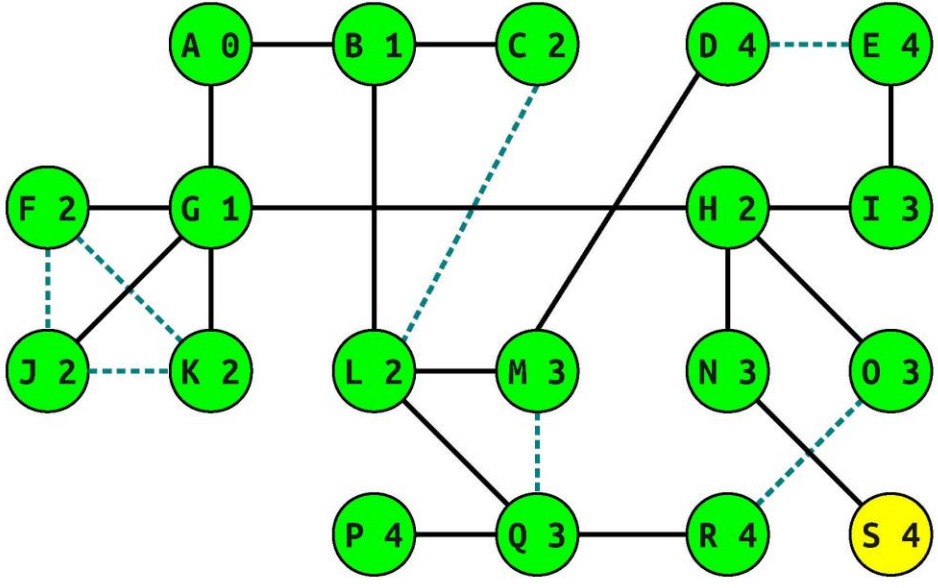
Breadth-First Search



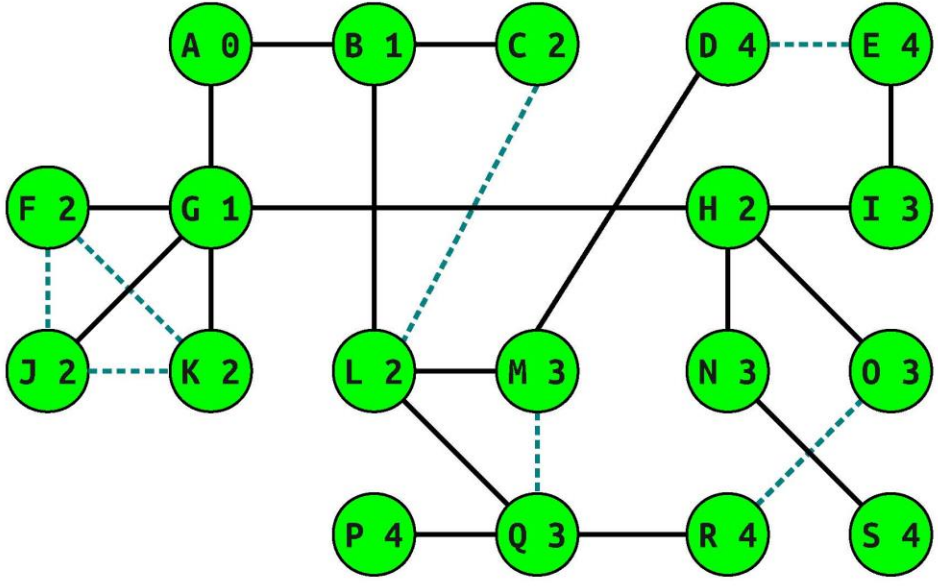
Breadth-First Search



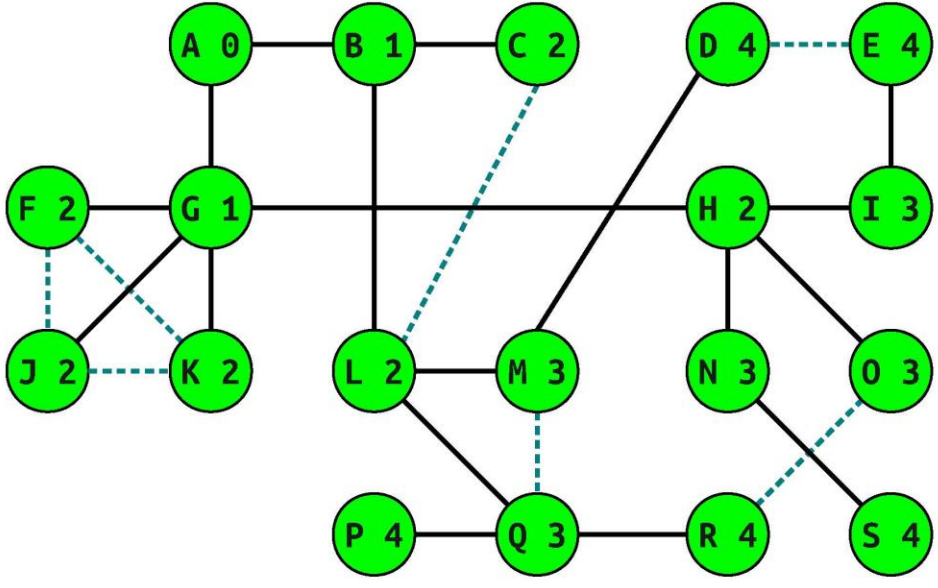
Breadth-First Search



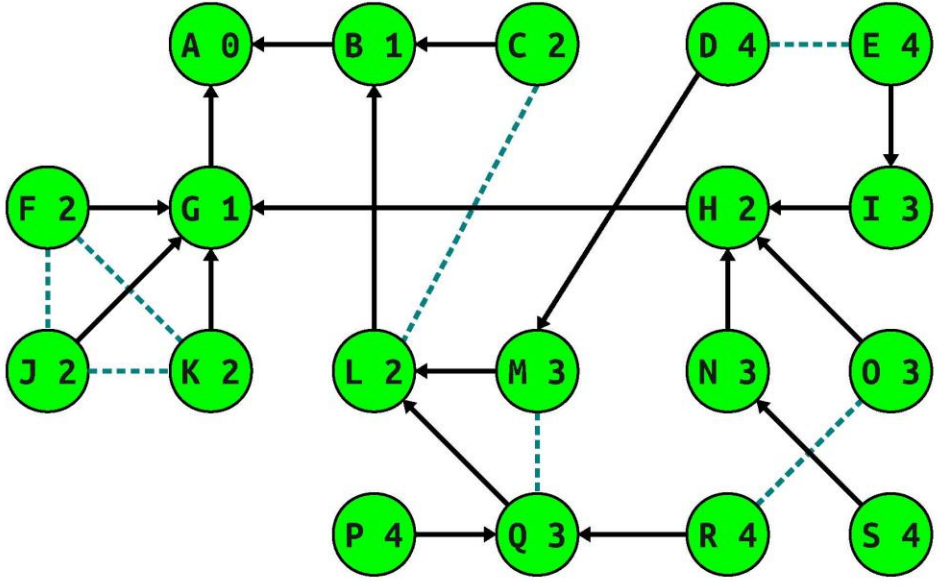
Breadth-First Search



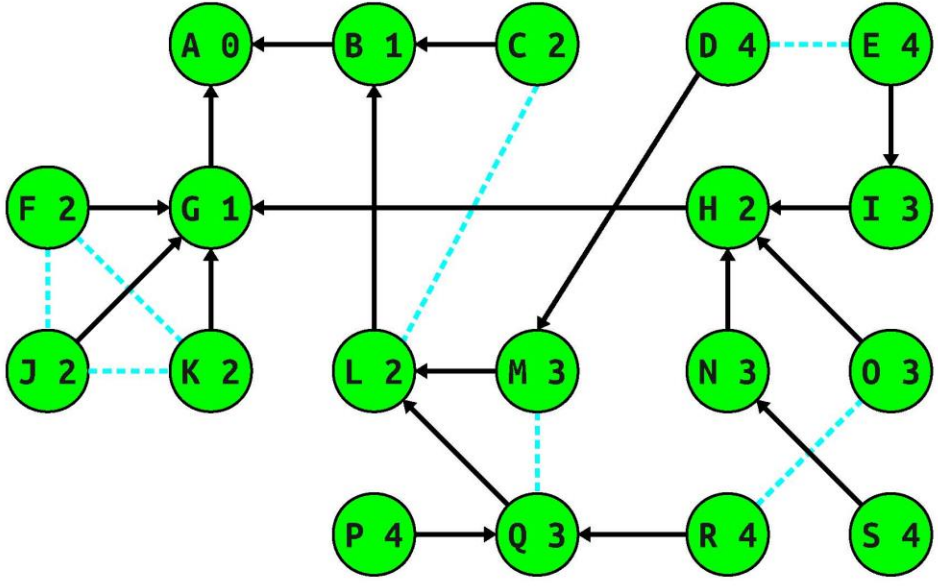
Breadth-First Search



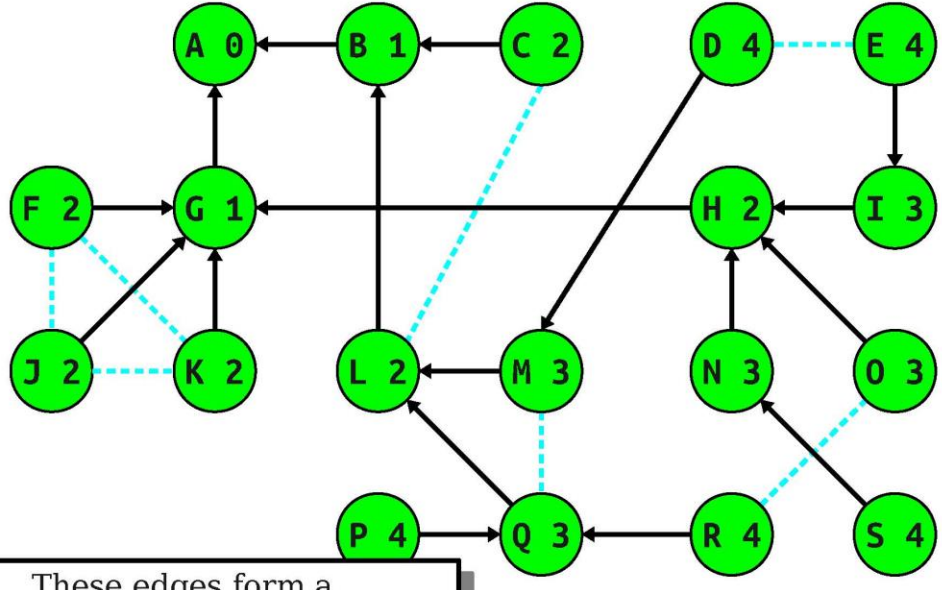
Breadth-First Search



Breadth-First Search

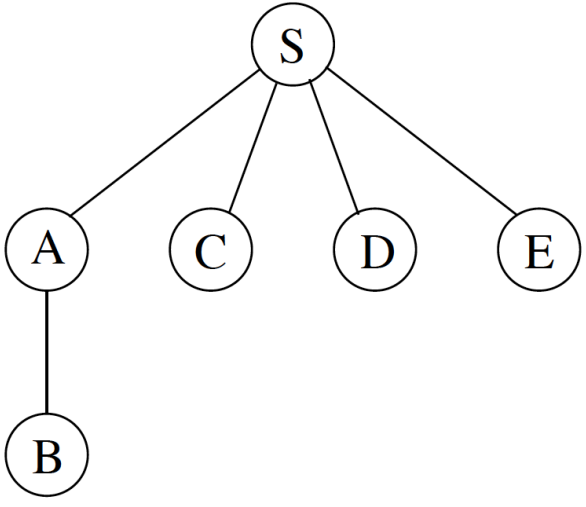


Breadth-First Search



These edges form a **breadth-first search tree**: the path from any v to node A gives a shortest path from v to A.

Example from the Book



Order of visitation	Queue contents after processing node
	$[S]$
S	$[A\ C\ D\ E]$
A	$[C\ D\ E\ B]$
C	$[D\ E\ B]$
D	$[E\ B]$
E	$[B]$
B	$[\,]$