Welcome (Again!) to
MATH 4100/COMP 5360– Introduction to Data Science

Finishing up Networks

April 20, 2023

Based on prior lectures from
Alex Lex and Bei Wang Phillips

+ others where noted

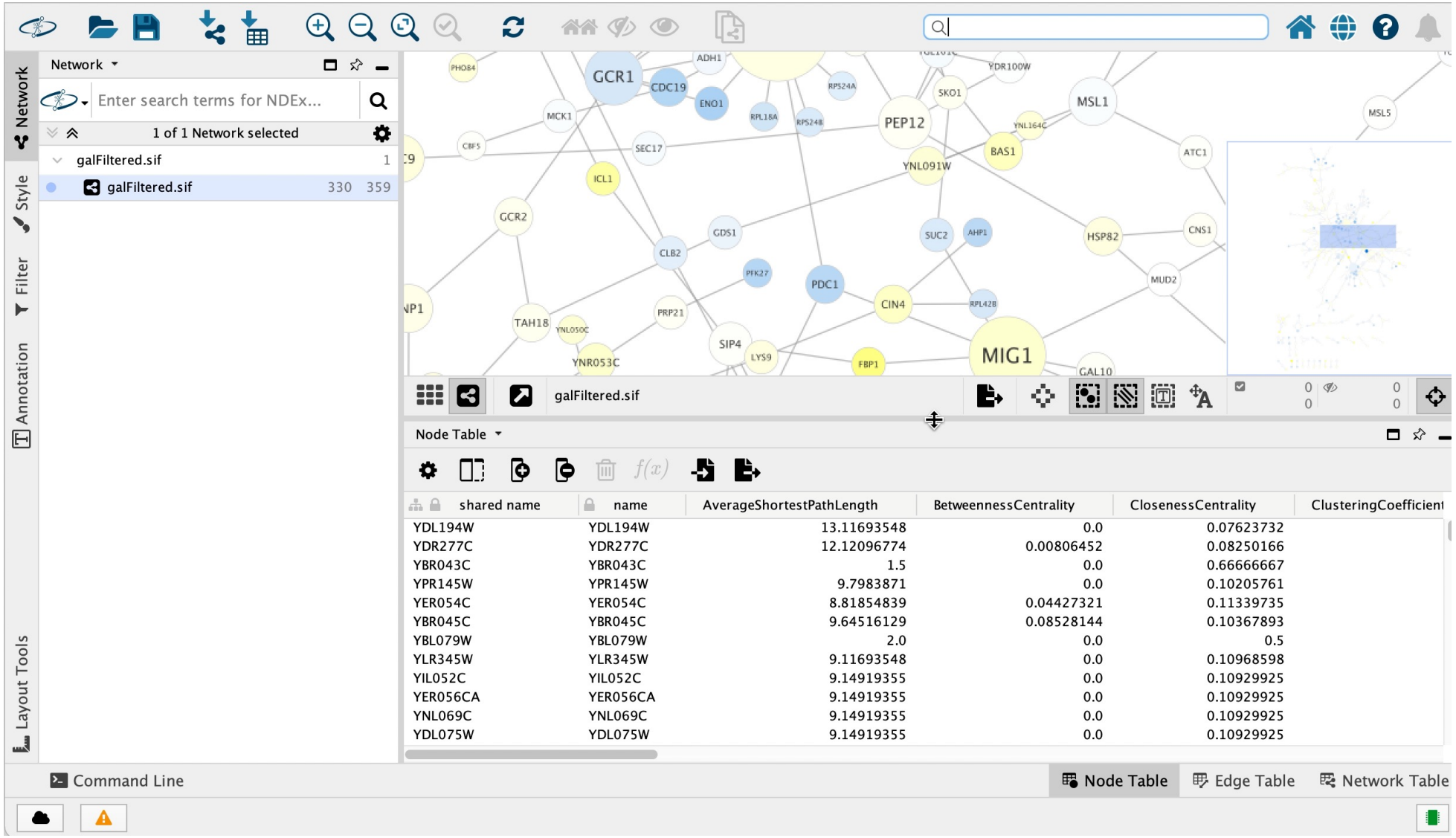
A good place to start for network-centric analysis and visualization is a graph drawing framework.

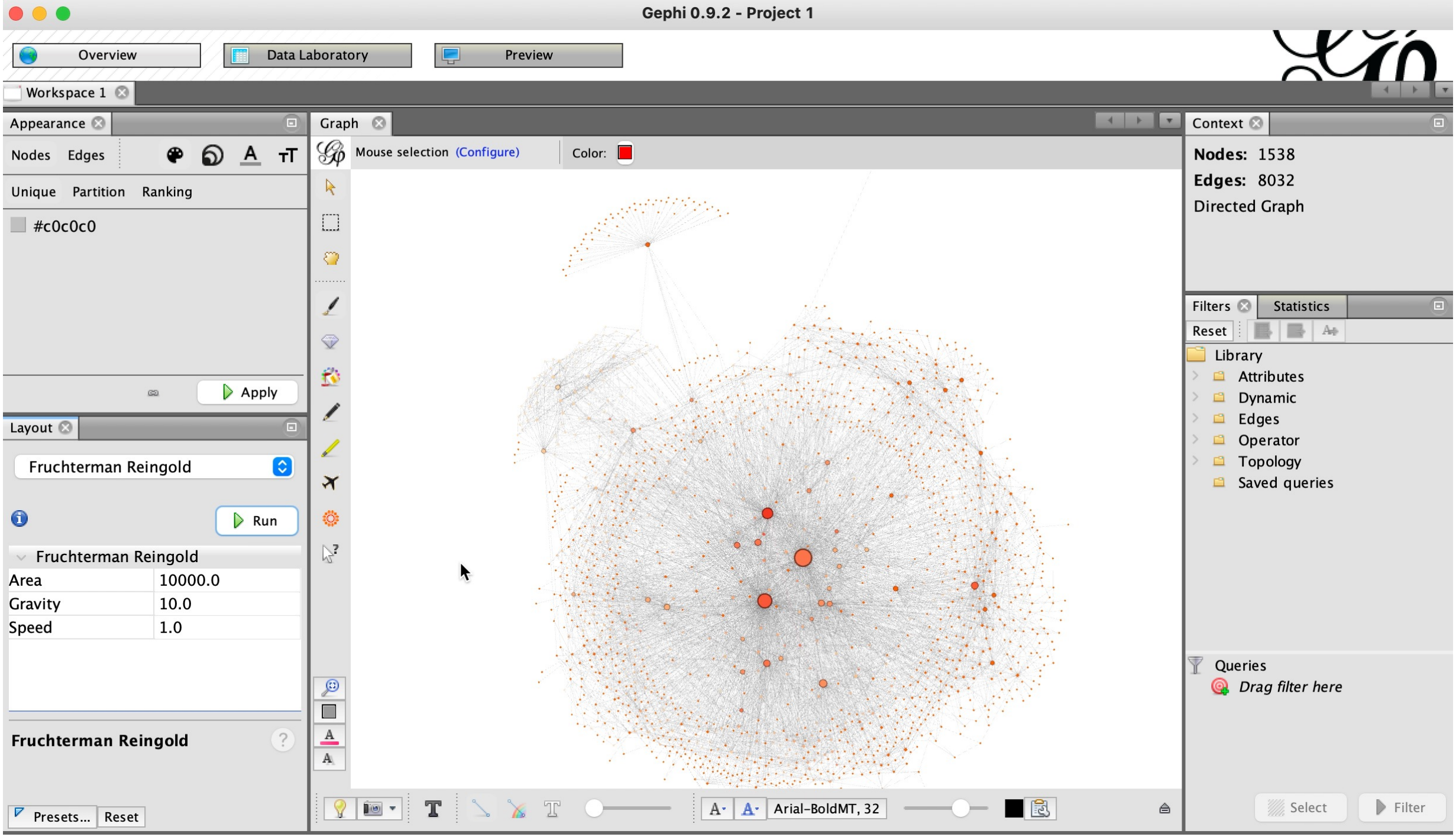
This slide:

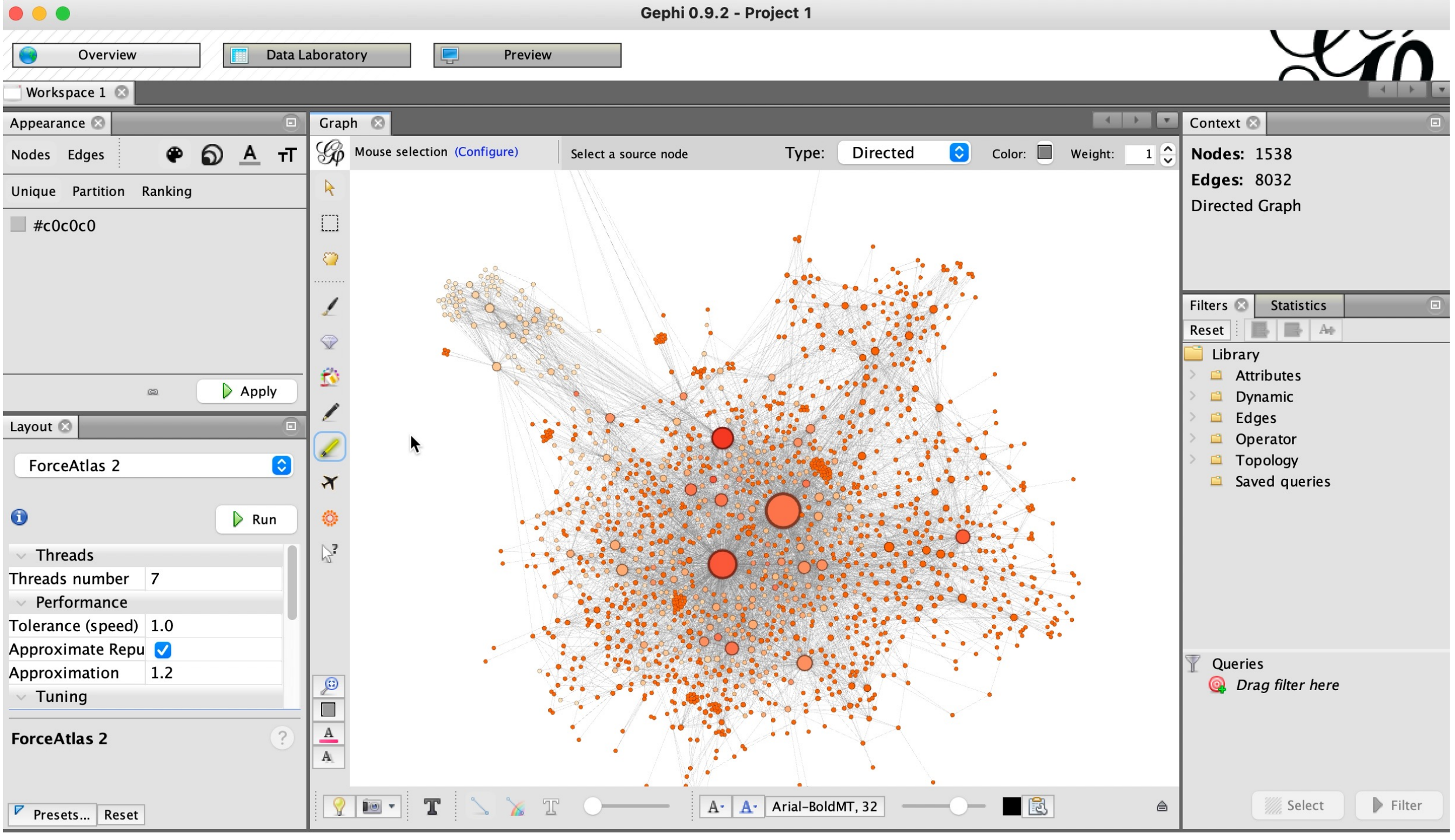
- Cytoscape

Next slide:

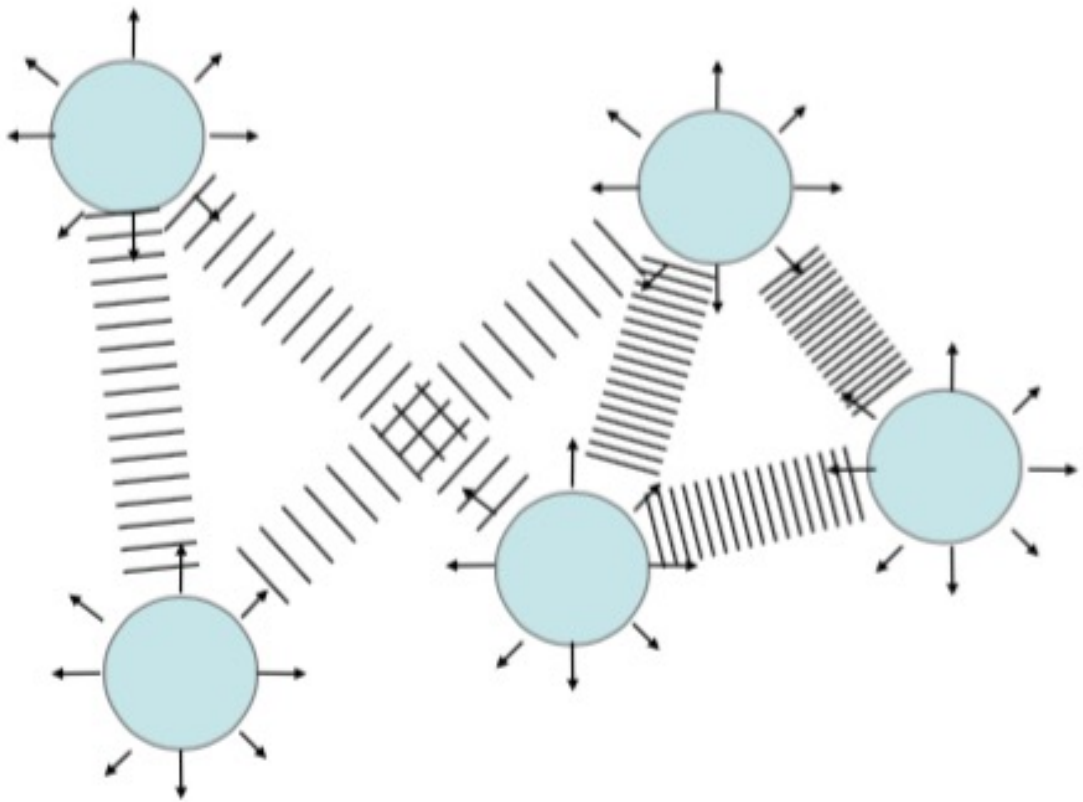
- Gephi




Gephi 0.9.2 - Project 1
Overview
Data Laboratory
Preview
Workspace 1
Appearance
Nodes
Edges
Unique
Partition
Ranking
#c0c0c0
Apply
Layout
Fruchterman Reingold
Run
Fruchterman Reingold
Area
10000.0
Gravity
10.0
Speed
1.0
Fruchterman Reingold
Presets...
Reset
Graph
Mouse selection (Configure)
Color:
Arial-BoldMT, 32
Context
Nodes: 1538
Edges: 8032
Directed Graph
Filters
Statistics
Reset
Library
Attributes
Dynamic
Edges
Operator
Topology
Saved queries
Queries
Drag filter here
Select
Filter


Gephi 0.9.2 - Project 1
Overview
Data Laboratory
Preview
Workspace 1
Appearance
Nodes
Edges
Unique
Partition
Ranking
#c0c0c0
Apply
Layout
ForceAtlas 2
Run
Threads
Threads number
7
Performance
Tolerance (speed)
1.0
Approximate Repu
Approximation
1.2
Tuning
ForceAtlas 2
Presets...
Reset
Graph
Mouse selection (Configure)
Select a source node
Type:
Directed
Color:
Weight:
1
Arial-BoldMT, 32
Context
Nodes: 1538
Edges: 8032
Directed Graph
Filters
Statistics
Reset
Library
Attributes
Dynamic
Edges
Operator
Topology
Saved queries
Queries
Drag filter here
Select
Filter

Force-directed layouts model networks as a physical system where nodes repel but edges attract



These layouts “work” on a wide variety of graphs.

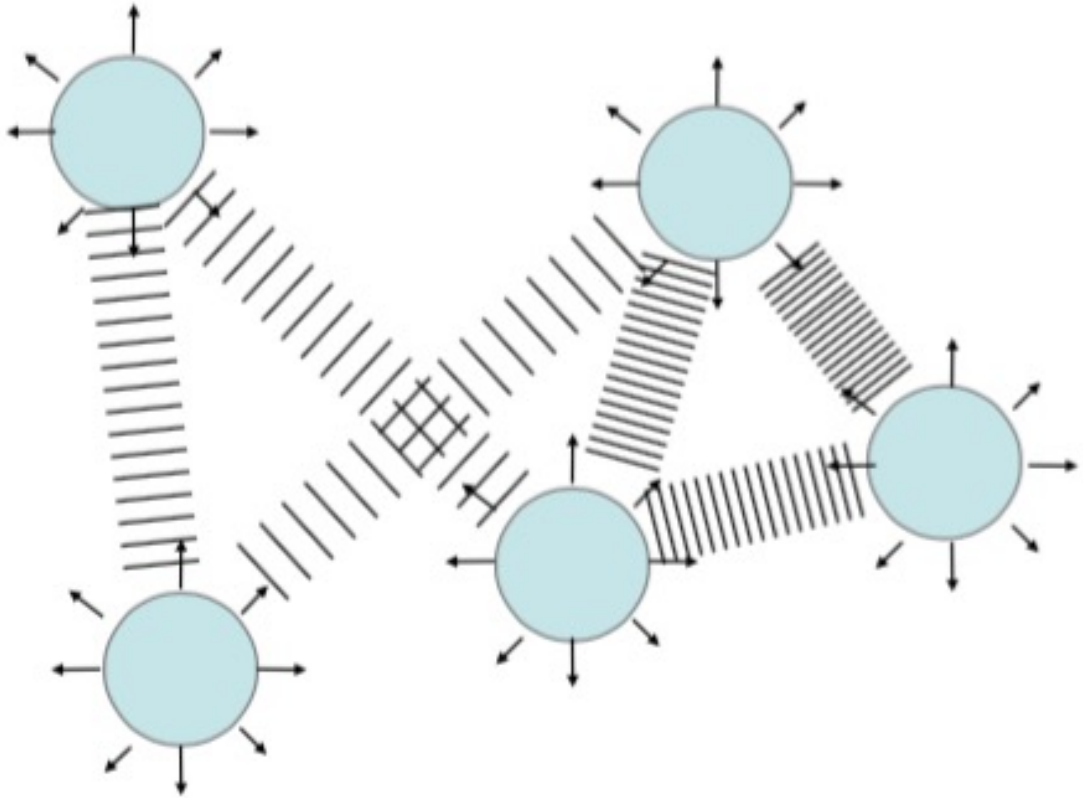
They also scale better than other approaches.

Issues:

- Lots of tweakable settings
- Are unstable - hay not be repeatable

Image from Josh Levine & Miriah Meyer, slides

Force-directed layout animation for Les Miserable graph



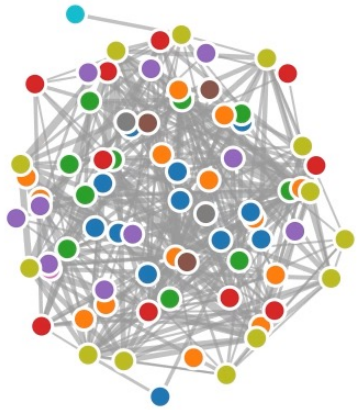


Image from Josh Levine & Miriah Meyer, slides

https://beta.observablehq.com/@mbostock/d3-force-directed-graph

If you know more about your network, another layout may perform “better.”

Hierarchical networks assume data has a layered, ordered, or otherwise hierarchical structure.

Examples:

- Workflows
- Computing
- Geneaology

Well-known hierarchical layout:

- **`dot`** in GraphViz (pygraphviz)

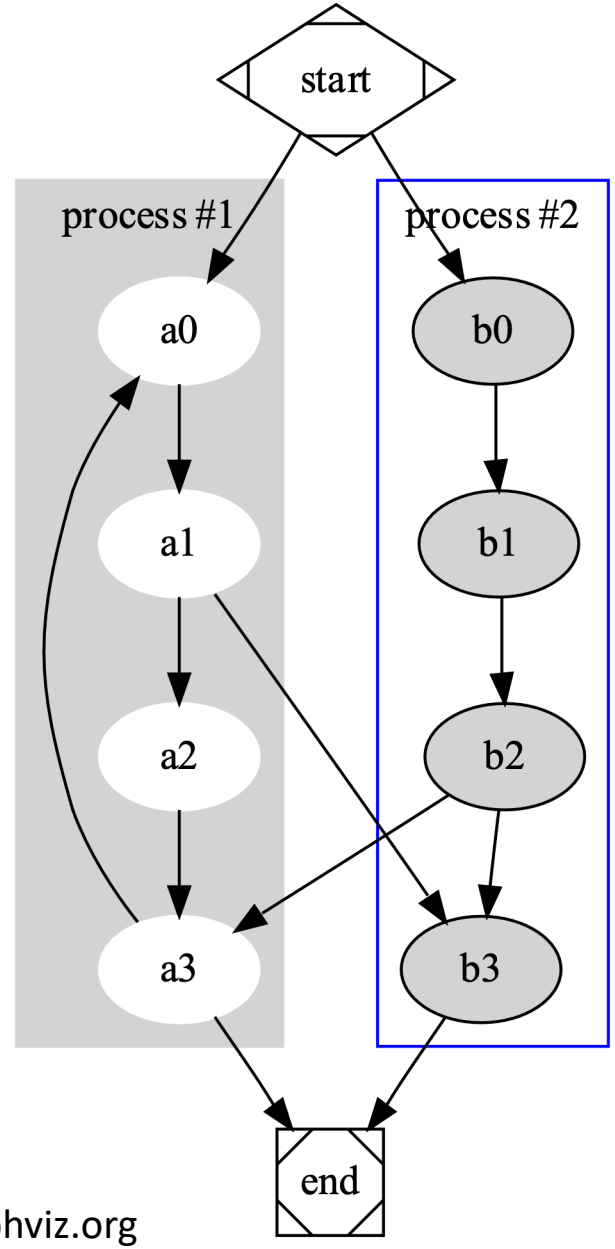


Image from graphviz.org

How to cite this site?

Check out other surveys!

treevis.net - A Visual Bibliography of Tree Visualization 2.0 by Hans-Jörg Schulz

v.18-NOV-2022

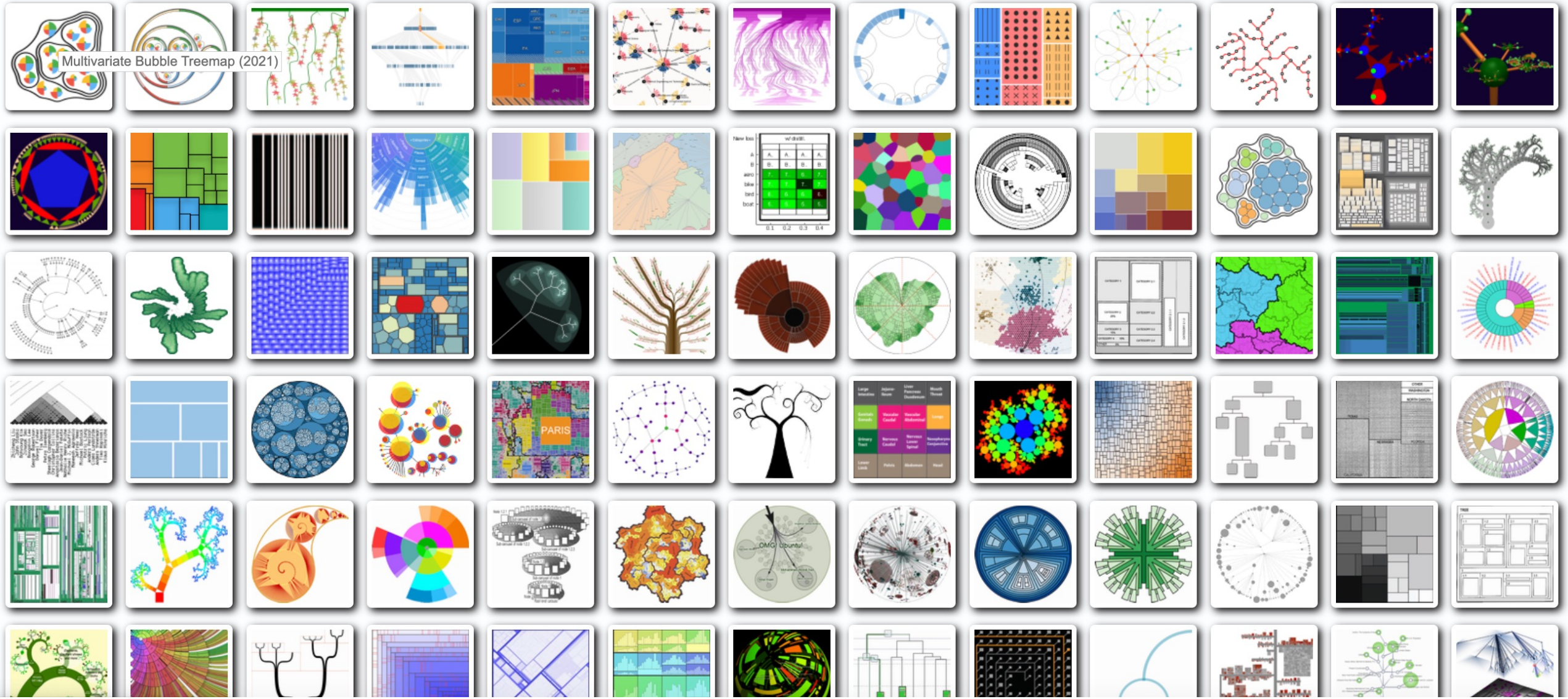
Dimensionality: All

Representation: All

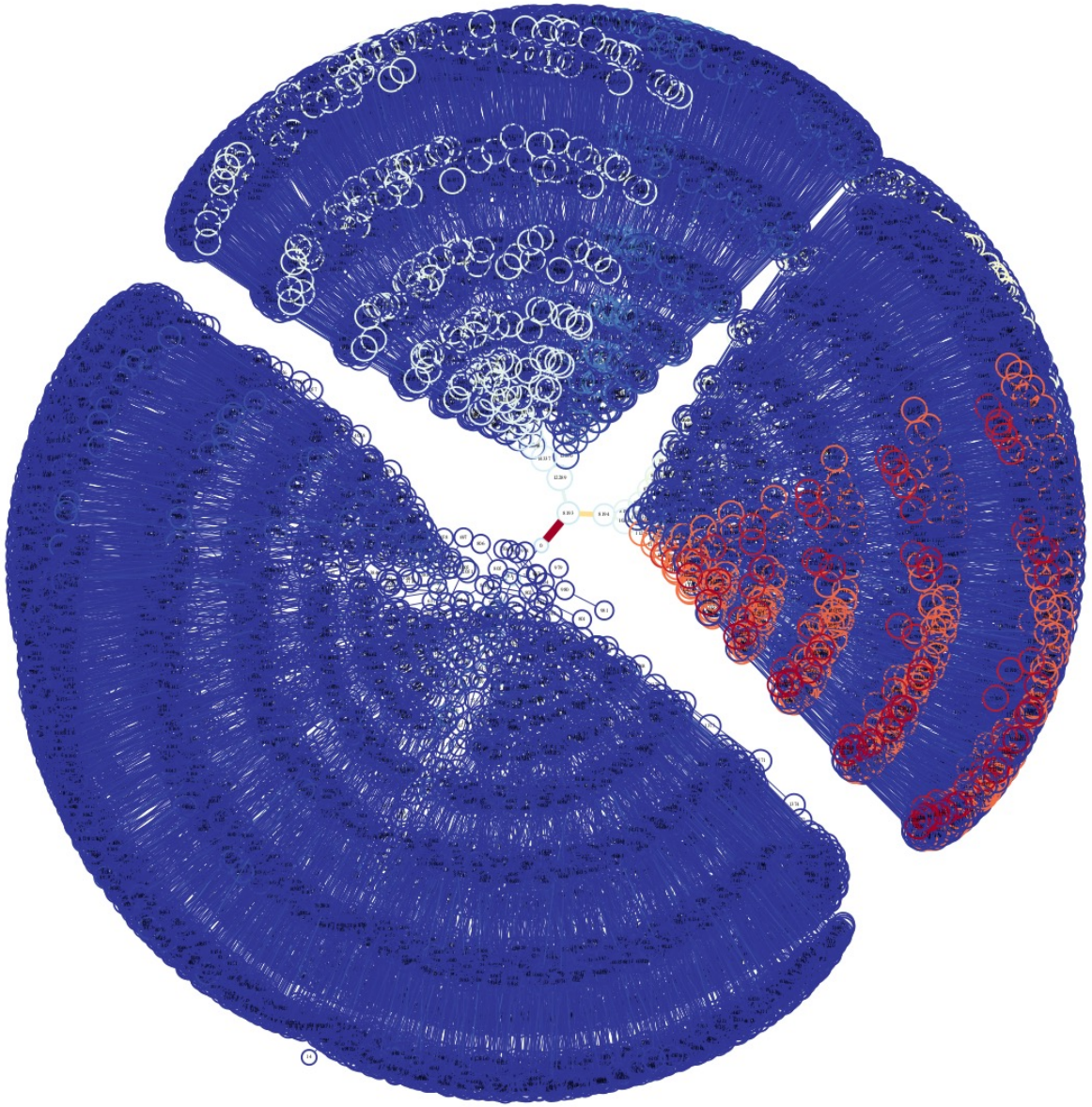
Alignment: All

Fulltext Search

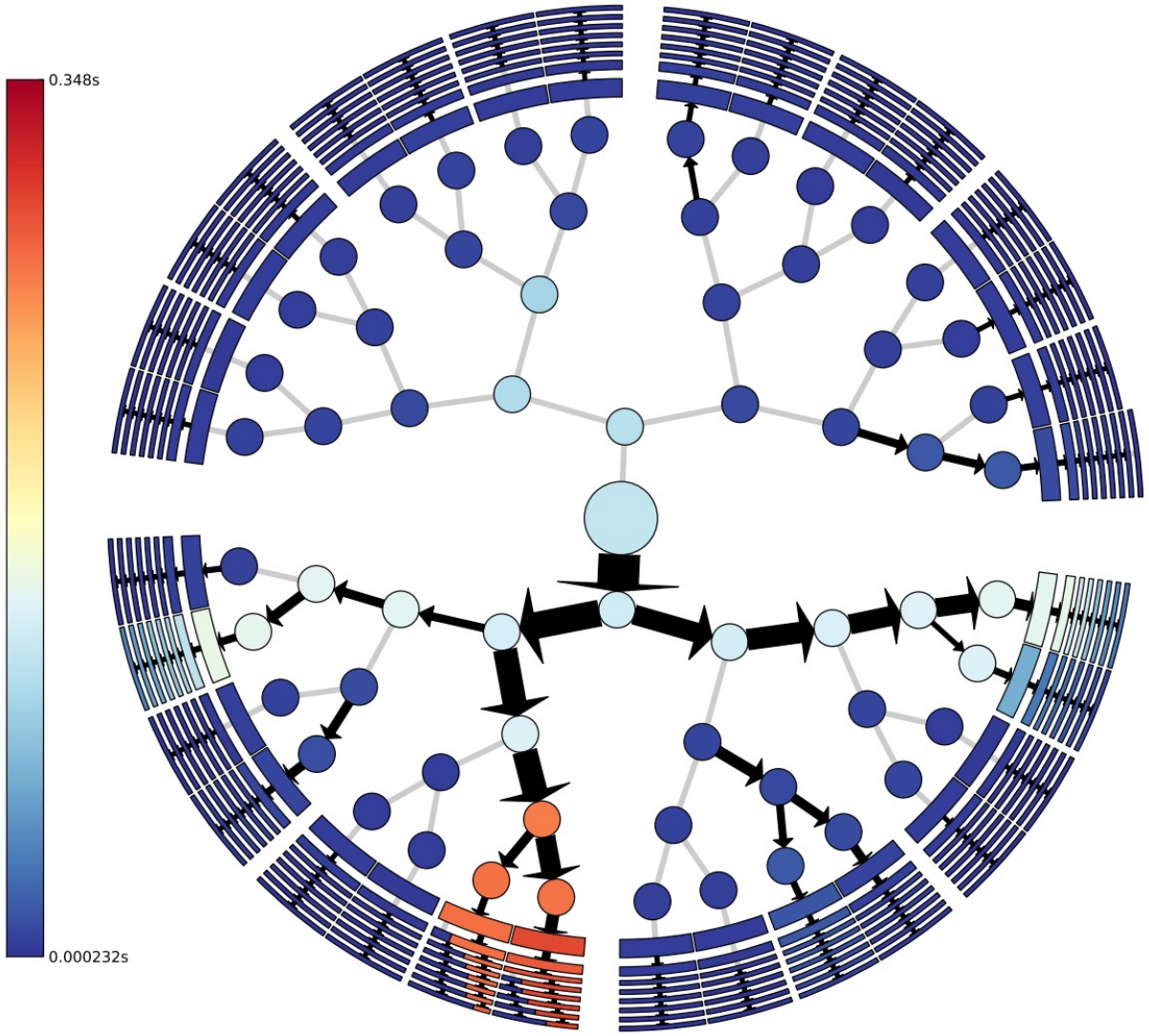
Techniques Shown: 333



When networks get large, aggregation can help.

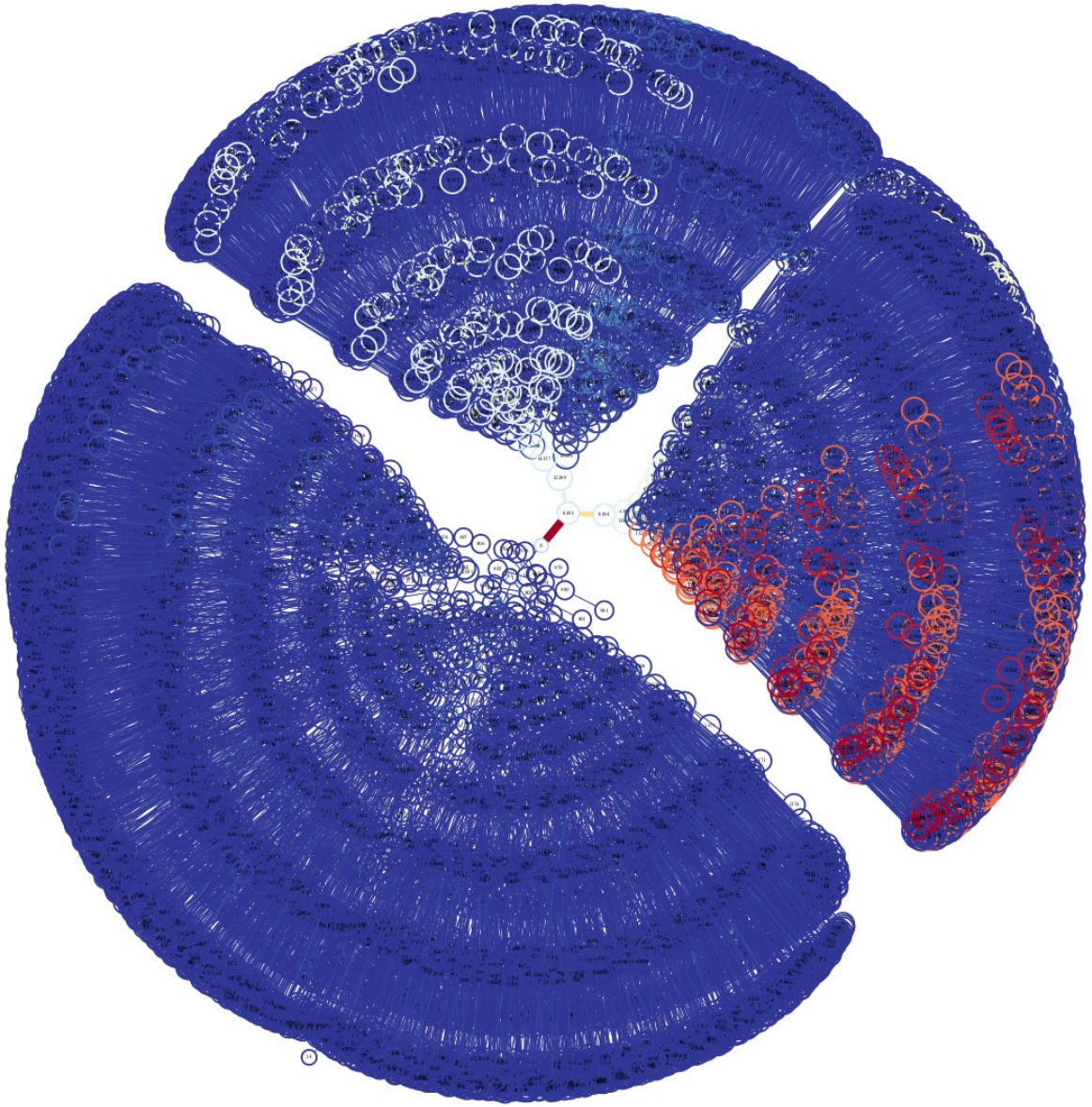


16,384 nodes, force-directed, GraphViz

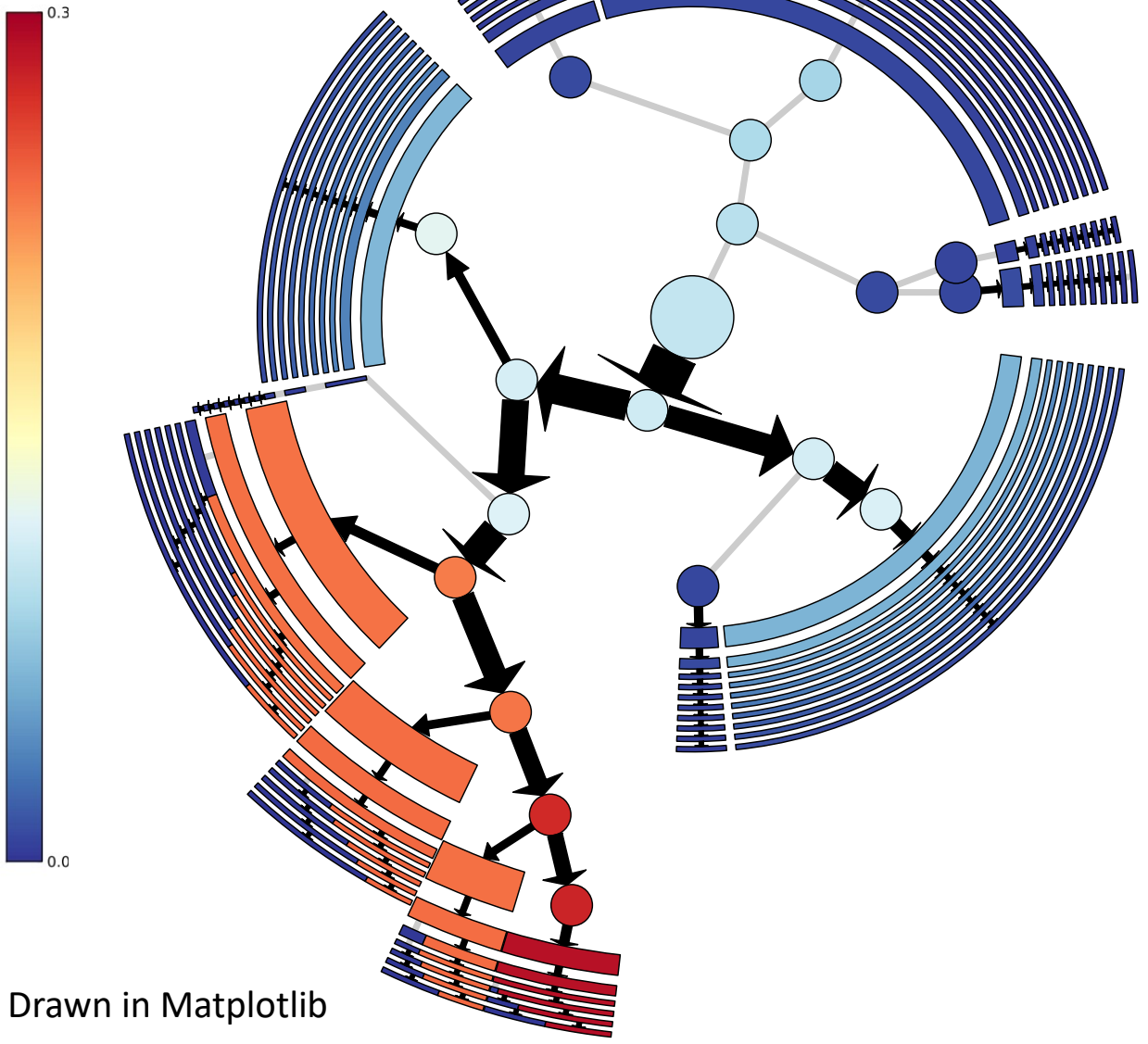


Drawn in Matplotlib

When networks get large, aggregation can help.



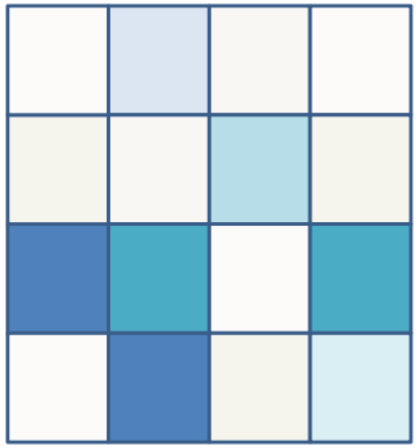
16,384 nodes, force-directed, GraphViz

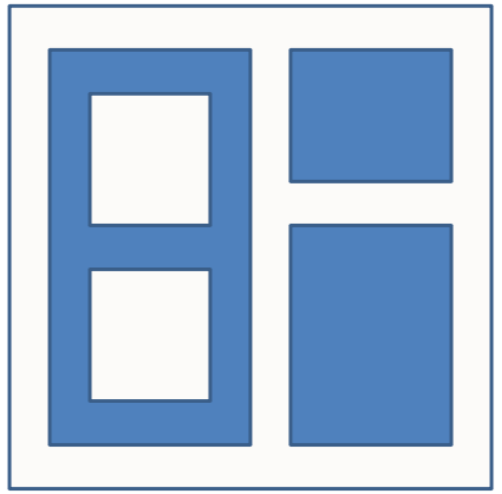


Drawn in Matplotlib

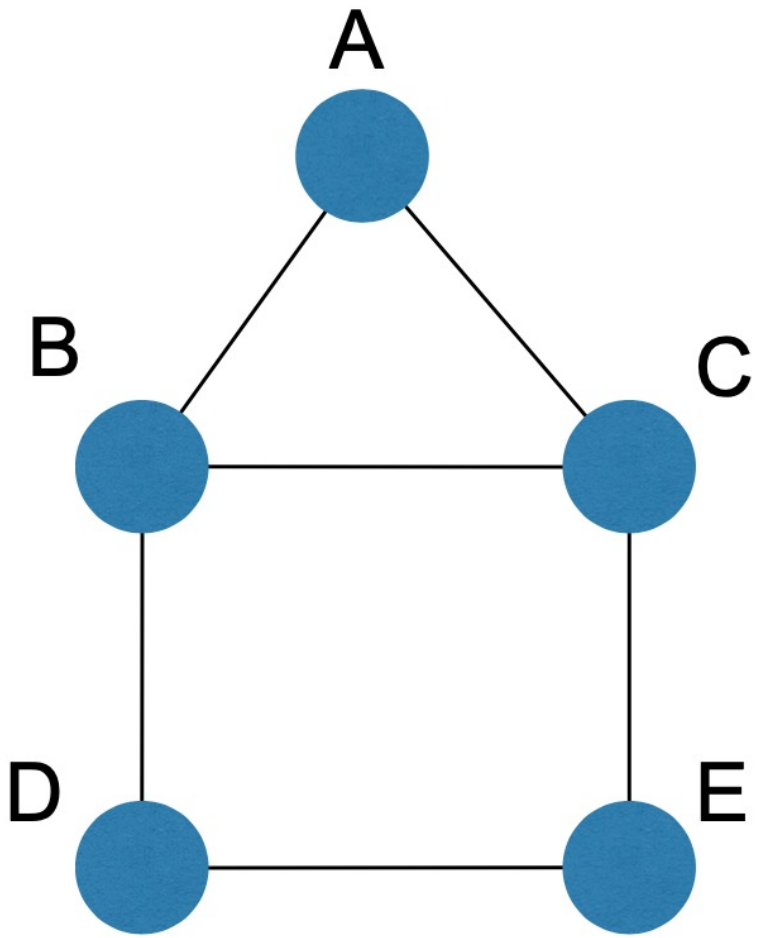
Eliding or filtering can help as well.

Networks don’t have to be drawn as node-link diagrams.



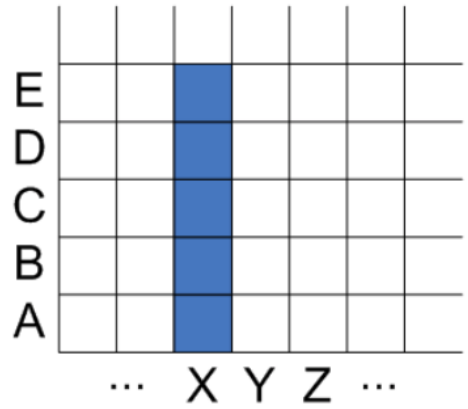


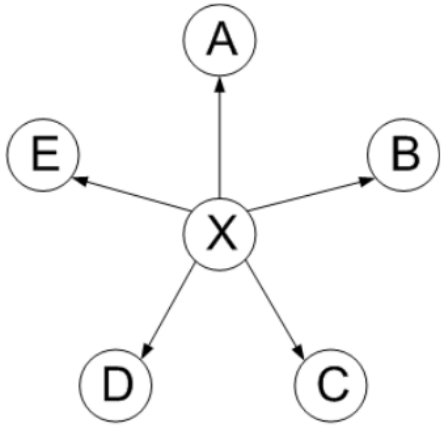
Graphs can be represented as adjacency matrices.

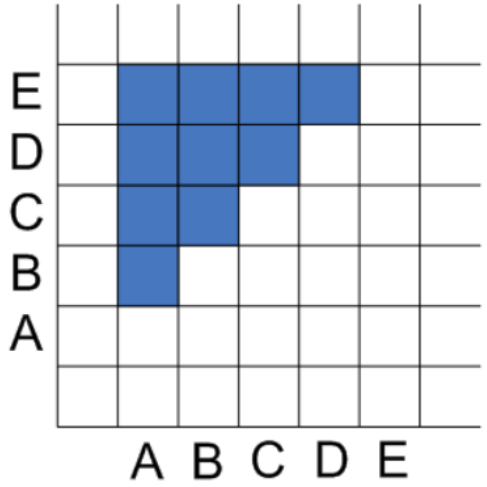


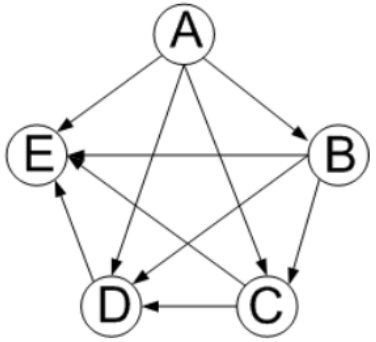
	A	B	C	D	E
A		■	■		
B	■		■	■	
C	■	■			■
D		■			■
E			■	■	

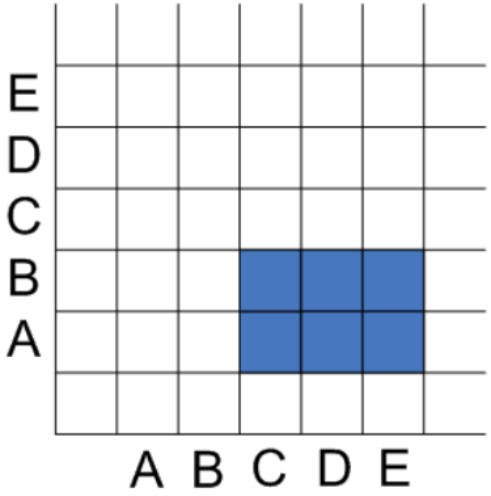
Patterns in directed adjacency matrices

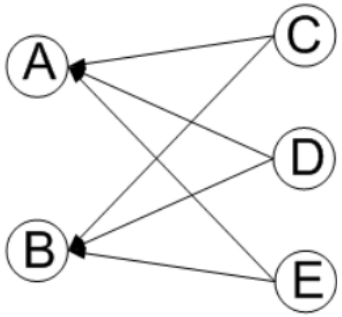


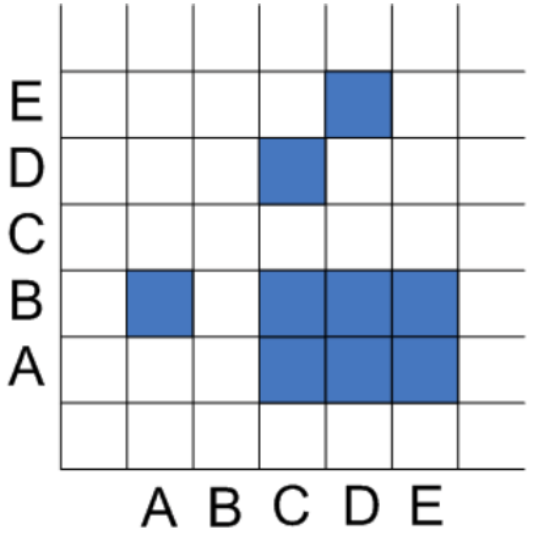


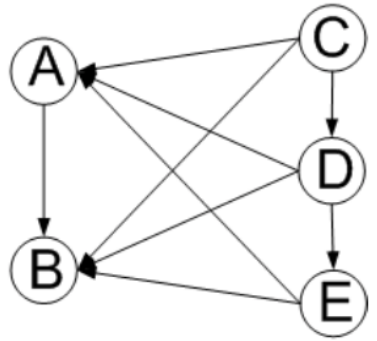






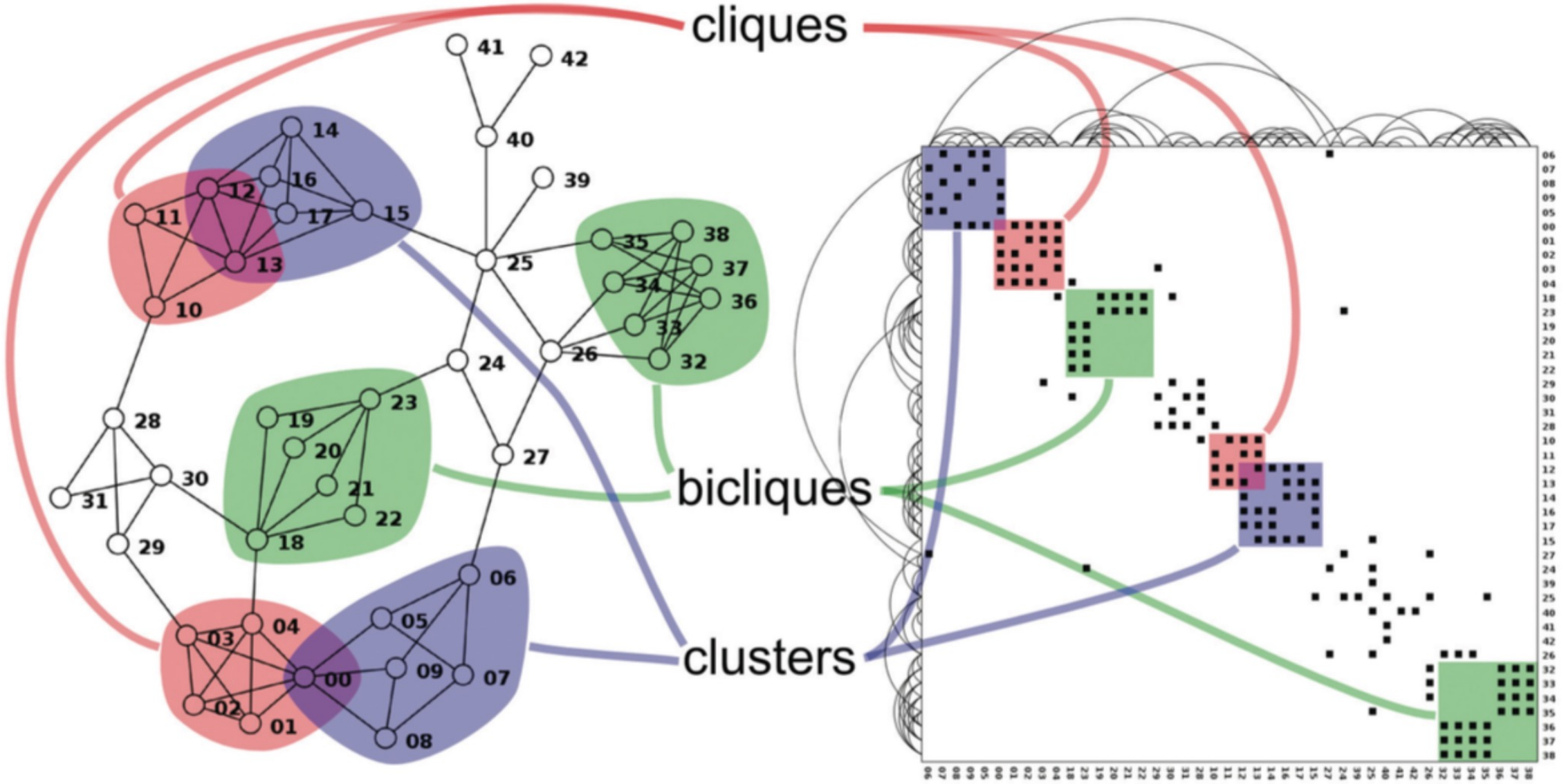






HJ Schulz 2007

Patterns in directed adjacency matrices



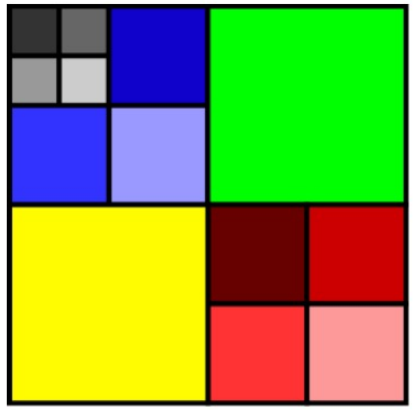
McGuffin 2012

But patterns are only revealed with order…

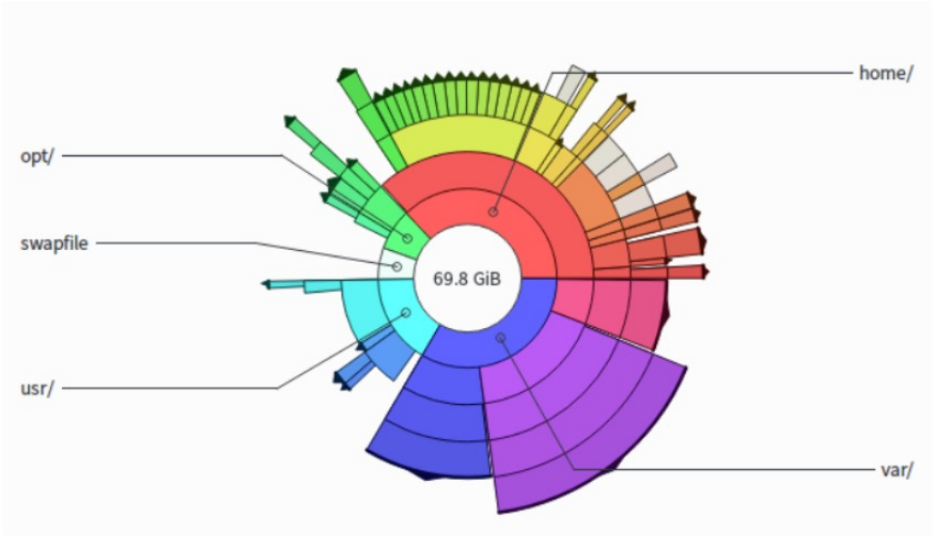
Les Miserable graph

Trees have additional options

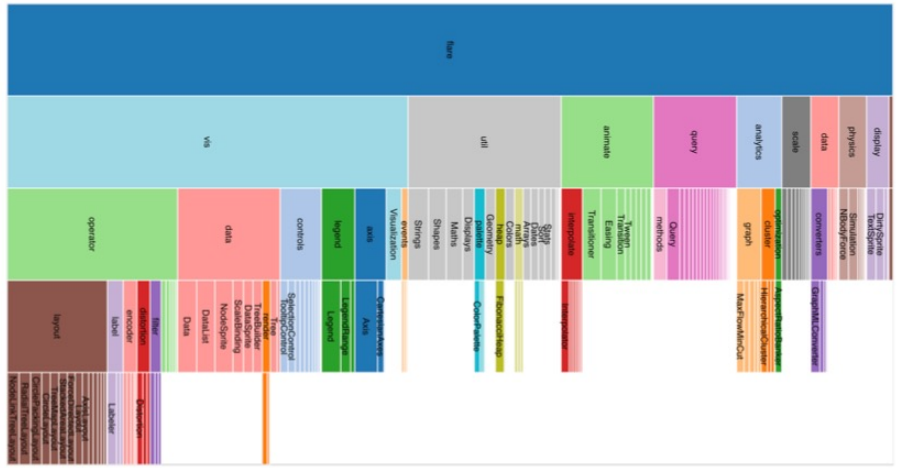
Treemap



Sunburst



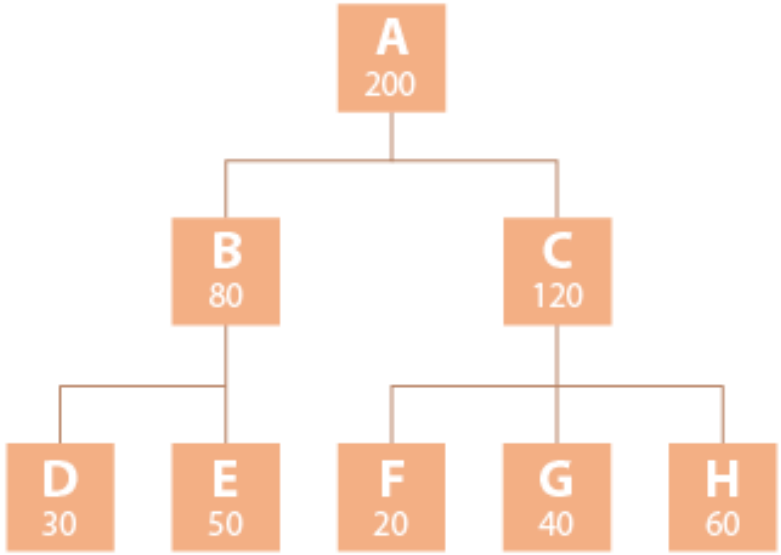
Icicle Plot



Treemaps

Recursively divide space based on some size metric of the nodes.

- What metrics?
  - Size of subtree
  - Attribute of node
- How to divide space?
  - Cluster
  - Pivot
  - Slice & Dice
  - Squarified
  - Strip
  - More…



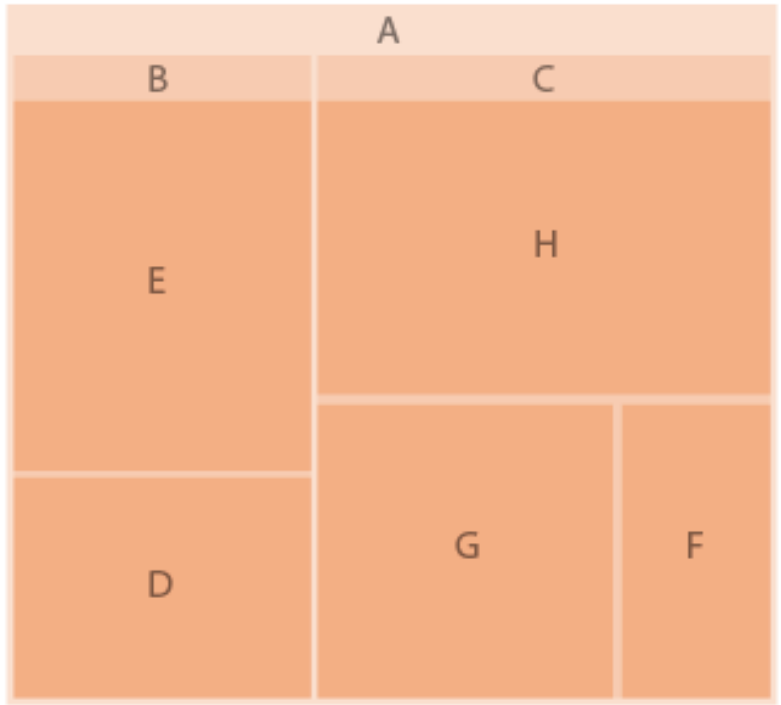
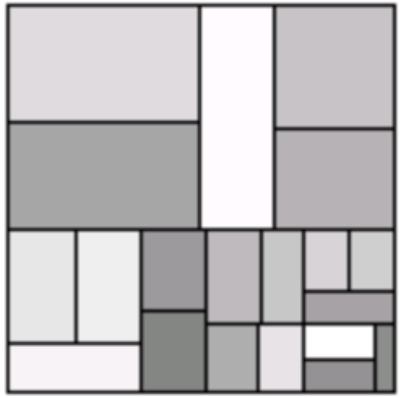
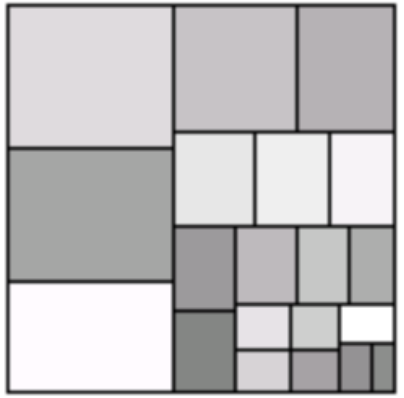


Image from DataVizCatalogue.com

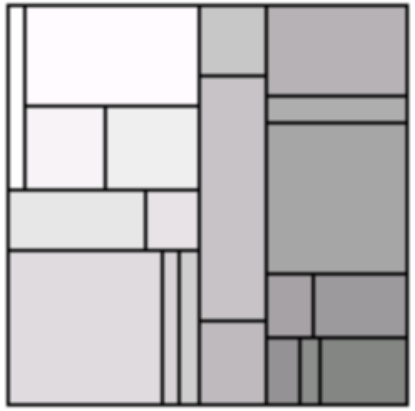
Treemap Layout Algorithms



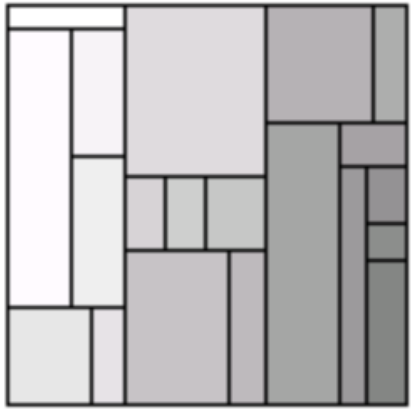
Cluster



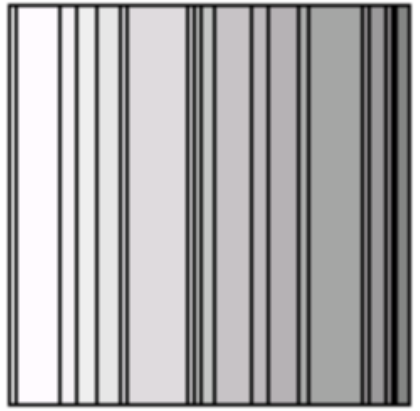
Squarified



Pivot-by-Middle

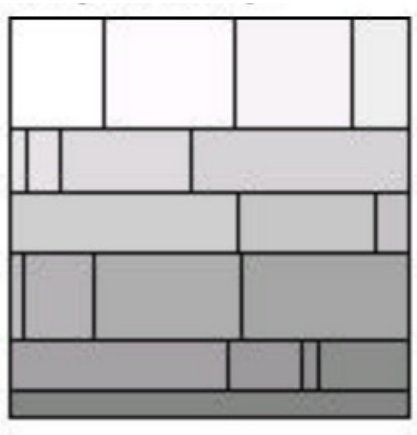


Pivot-by-Size



Slice-and-Dice

Grayscale indicates index order of nodes.

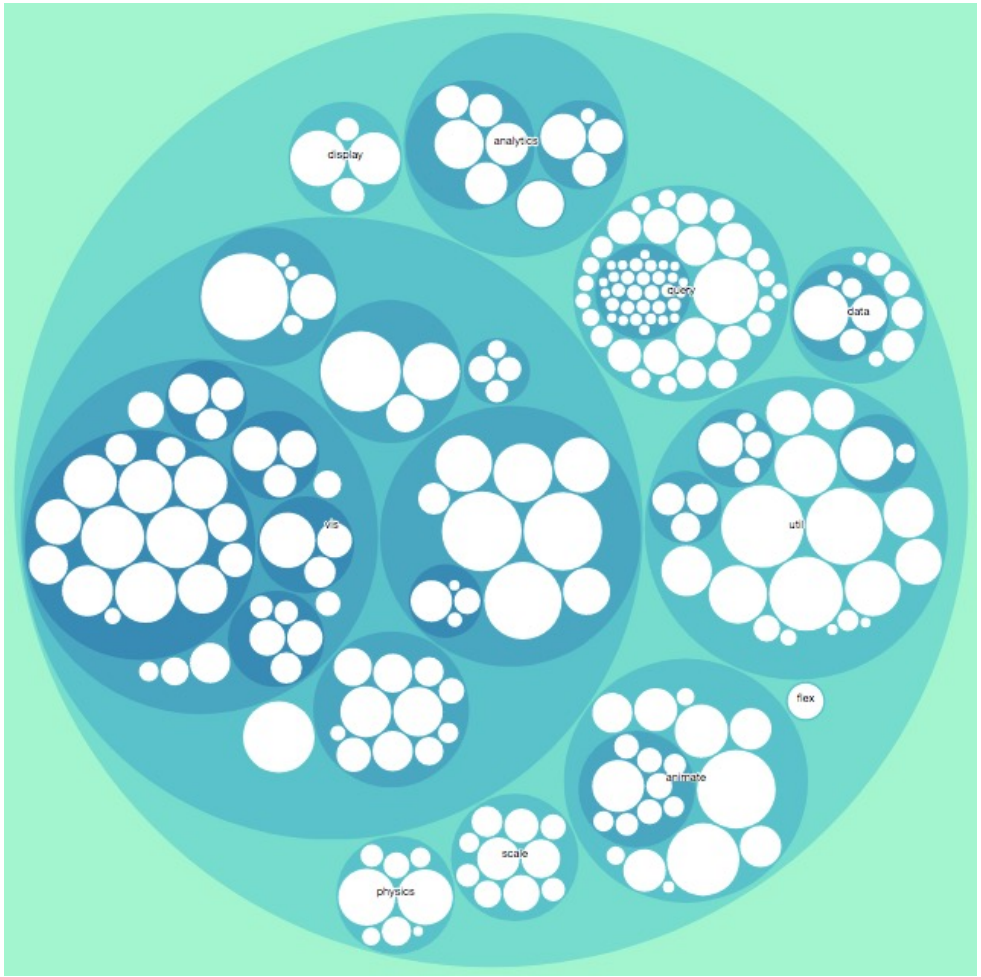


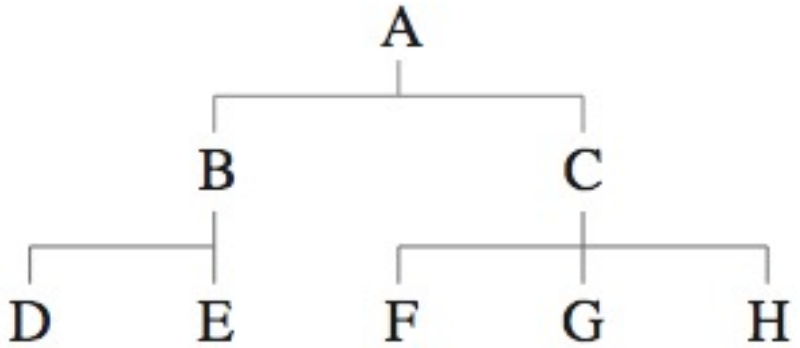
Strip

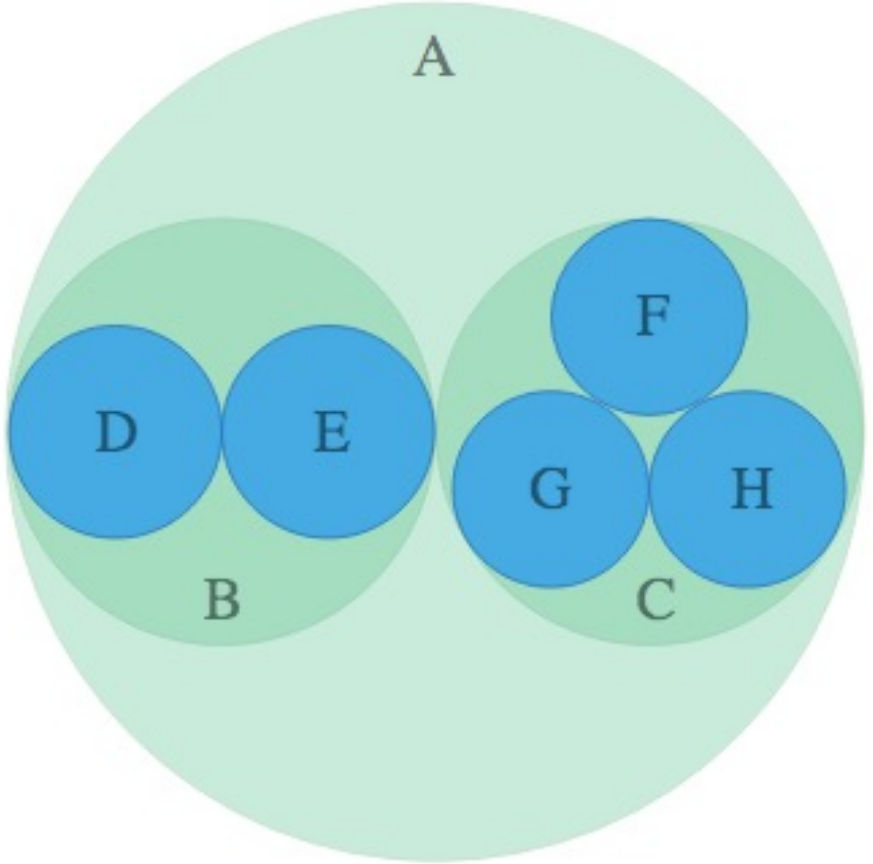
Images from Bederson, Shneiderman, & Wattenberg, “Ordered and Quantum Treemaps”


TECHNOLOGY
INTERNET INFORMATION
GOOGL -0.34%
FB +0.80%
APPLICATION S
ORCL -0.14
ADBE +0.04
CRM -0.13
INTU
SEMICONDUCT
INTC -0.45%
TXN
AVGO +0.97
BUSINESS SOFTWARE & S
MSFT +1.33%
TELECOM S
T
VZ
INFORMATIO
IBM -0.72
ACN
FIS
NVDA -0.28
AMAT
DIVER
CCI
MU
COMMUNIC
CSCO +0.12%
ATVI
EA
SERVICES
CATALOG & MAIL O
AMZN +0.72%
DISCOUN
WMT +2.38%
COST -0.46
TGT
ENTERTAI
DIS
CMCSA +1.38%
FOXA
BUSINESS
BKNG -0.21%
CATV SYS
NFLX +1.11%
RESTAU
MCD +1.47%
MAR
TJX
SBUX +2.03
RAILROA
UNP -0.78
CSX
NSC
DAL
HOME I
HD
LOW
AIR DELI
UPS -0.81
FDX
LUV
BASIC MATERIALS
MAJOR INT
XOM +0.38%
CVX
INDEPE
COP +1.62
EOG
OXY
SPECIAL
PX
LYB
CHEMIC
DWDP -2.04%
SLB +0.12%
HAL
KMI
PSX
VLO
HEALTHCARE
DRUG MANUFACTURE
JNJ -0.38%
MRK +1.01%
ABBV -0.19%
PFE
LLY -0.36
BMY +0.51
BIOTECHN
AMGN +0.67%
GILD +0.78
BIIB
VRTX
CELG
MEDICAL
MDT +0.22
ABT -0.18
SYK
BSX
MEDICA
BDX +0.17
ISRG
BAX
MEDIC
TMO
AGN
ZTS
HCA
HEALTH CARE PLANS
UNH +0.42%
CVS -0.48
AET
CI
FINANCIAL
MONEY CENTER BANKS
JPM -0.56%
WFC +0.12%
BAC -0.68%
C
PNC -0.12
STI
CREDIT SERVICES
V
MA +0.54%
PYPL -0.76
AXP -0.47
DFS
SYF
PROPERTY & CASUAL
BRK-B +0.39%
AIG
INVESTM
MS -1.23
GS -0.69
CME
ASSET
BLK
BK
STT
REGION
USB
REIT -
LIFE I
INSUR
REIT -
AMT +0.74
ICE
AFL
CONSUMER GOODS
ELECTRONIC
AAPL +0.81%
BEVERAGES -
KO -0.10%
PEP +0.23%
CIGARE
PM
MO
TEXTILE
NKE
K
F
PERSONAL
PG +0.04%
CL
EL
KMB
IP
KHC
STZ
JCI
INDUSTRIAL GOODS
DIVERSIFIE
MMM -1.04
HON -1.88
GE +0.44%
DHR
ITW
ETN
IR
AEROSPACE/DEFEN
BA -0.12%
UTX -2.22%
GD
RTN
LMT
CAT -2.37
DE
UTILITIES
ELECTRIC U
NEE -0.13
SO
D
DIVERSI
Use mouse wheel to zoom in and out. Drag zoomed map to pan it.
Double-click a ticker to display detailed information in a new window.
Hover mouse cursor over a ticker to see its main competitors in a stacked view with a 3-month history graph.
-3%
-2%
-1%
0%
+1%
+2%
+3%

Treemaps need not be rectangular







Also known as circle packing, Images from DataVizCatalogue.com and https://bl.ocks.org/mbostock/7607535

Indented Trees

- Parent-child relationships shown via indention
- Trade-off between breadth (one level) and depth (to the leaves)

```
root
├── dir1
├── dir2
│   └── file1
├── dir3
│   ├── file2
│   ├── file3
│   └── dir4
│       └── file4
└── file5
```

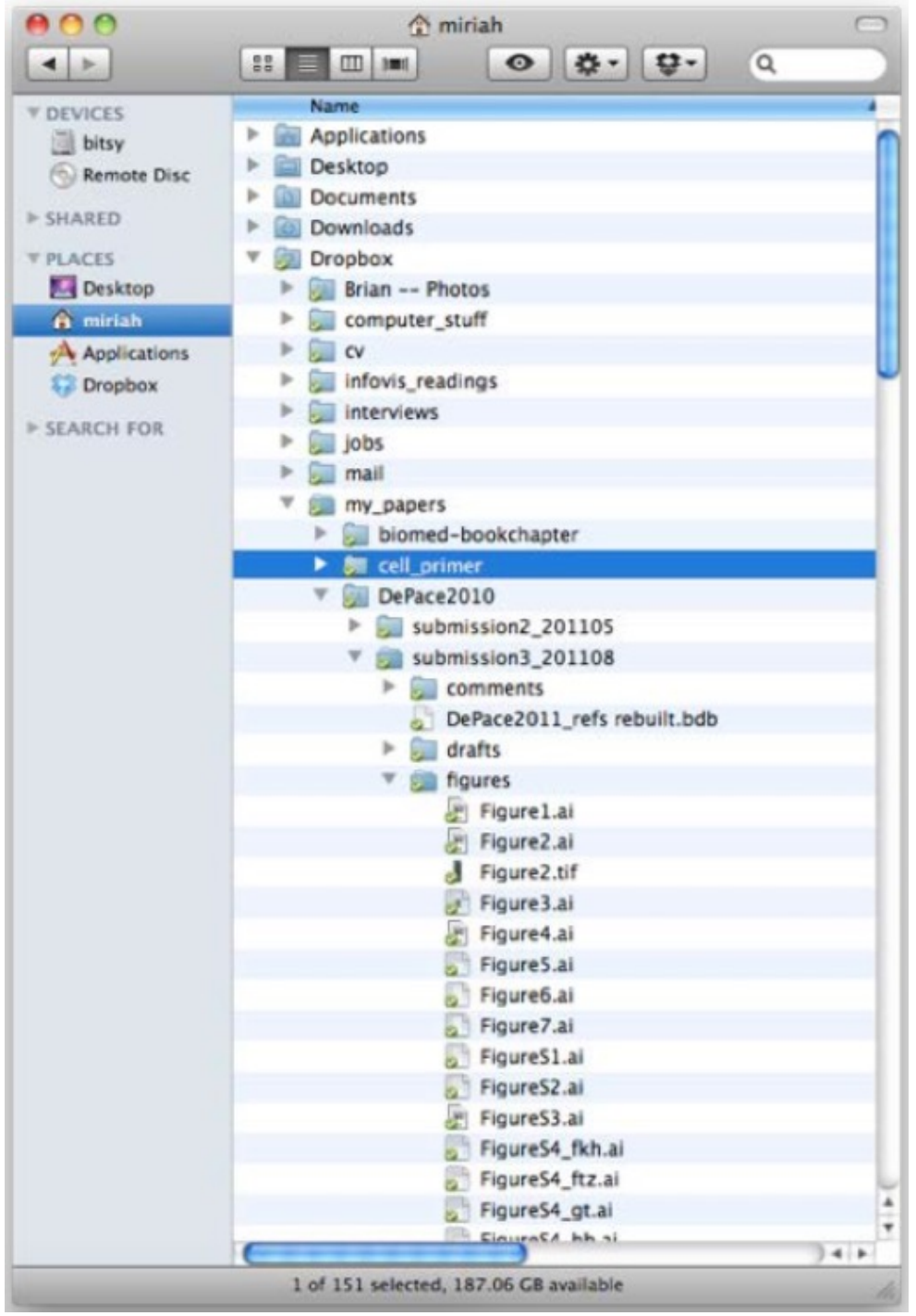


Image from Miriah Meyer & Josh Levine, slides and https://atom.io/packages/ascii-tree

Layered Trees

Parent-child relationships shown via layering, adjacency, & alignment

Layout similar to node-link without edges

Extent of parent (e.g., length, angle) constrains extent of children, similar to enclosure

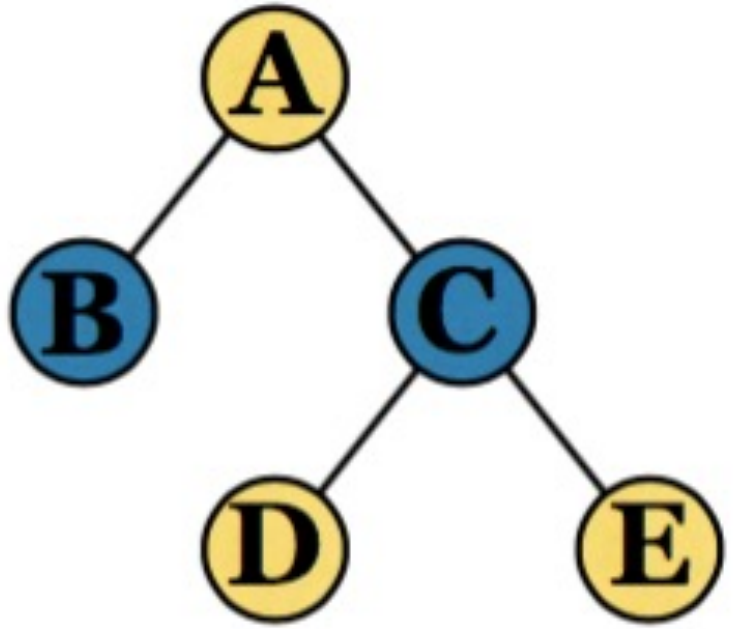
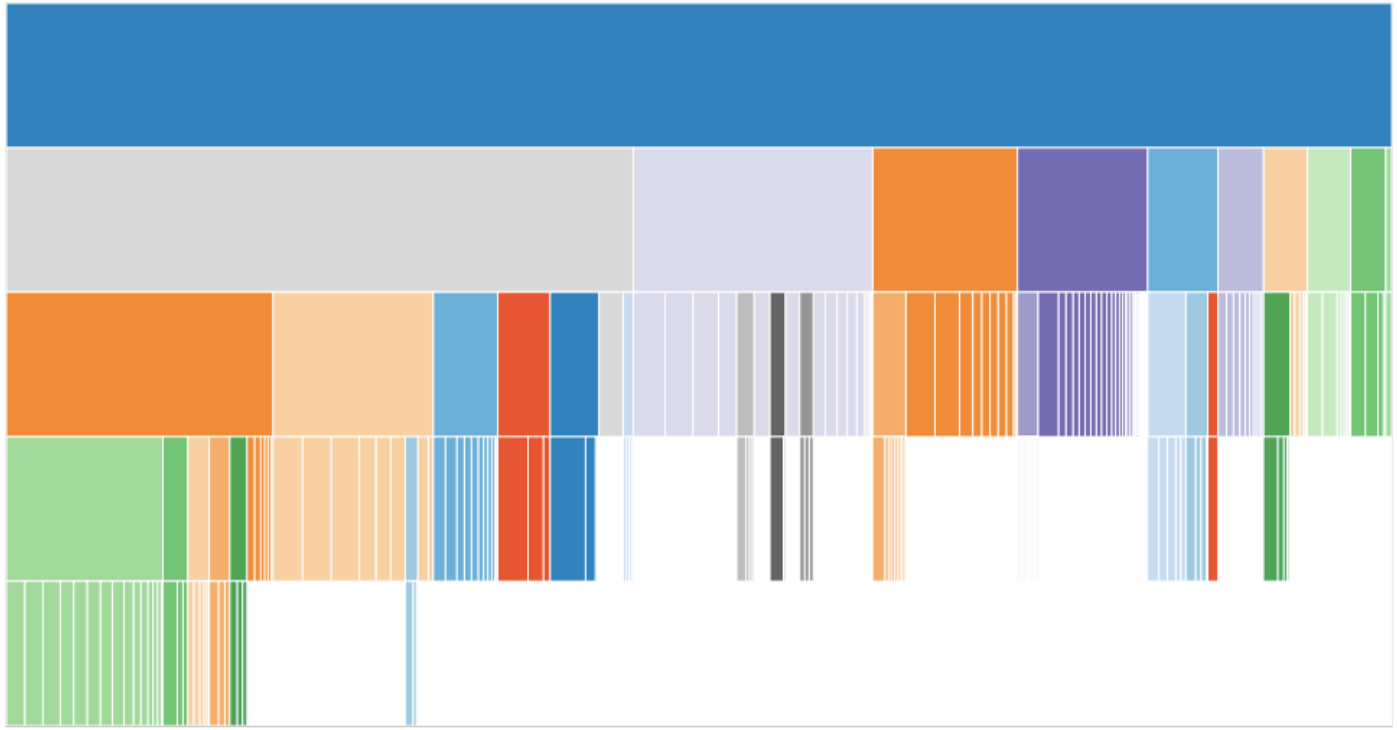
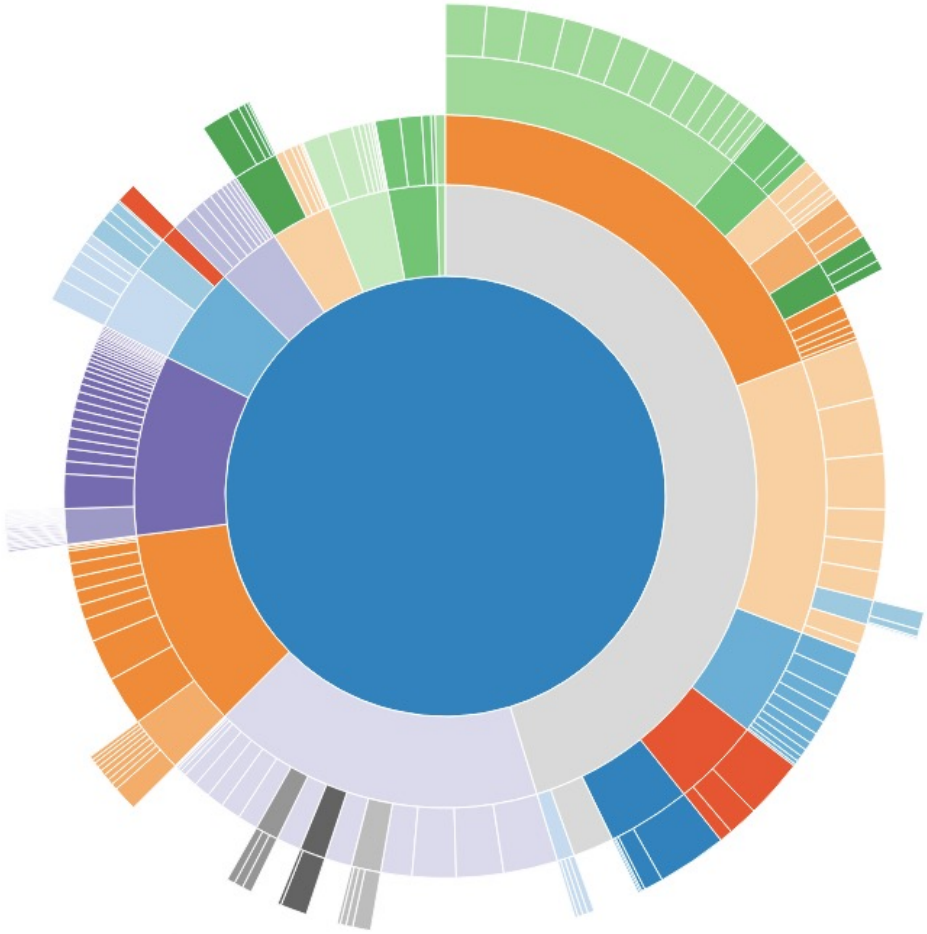


Image from Miriah Meyer & Josh Levine, slides and https://atom.io/packages/ascii-tree

Icicle charts and sunbursts use layering

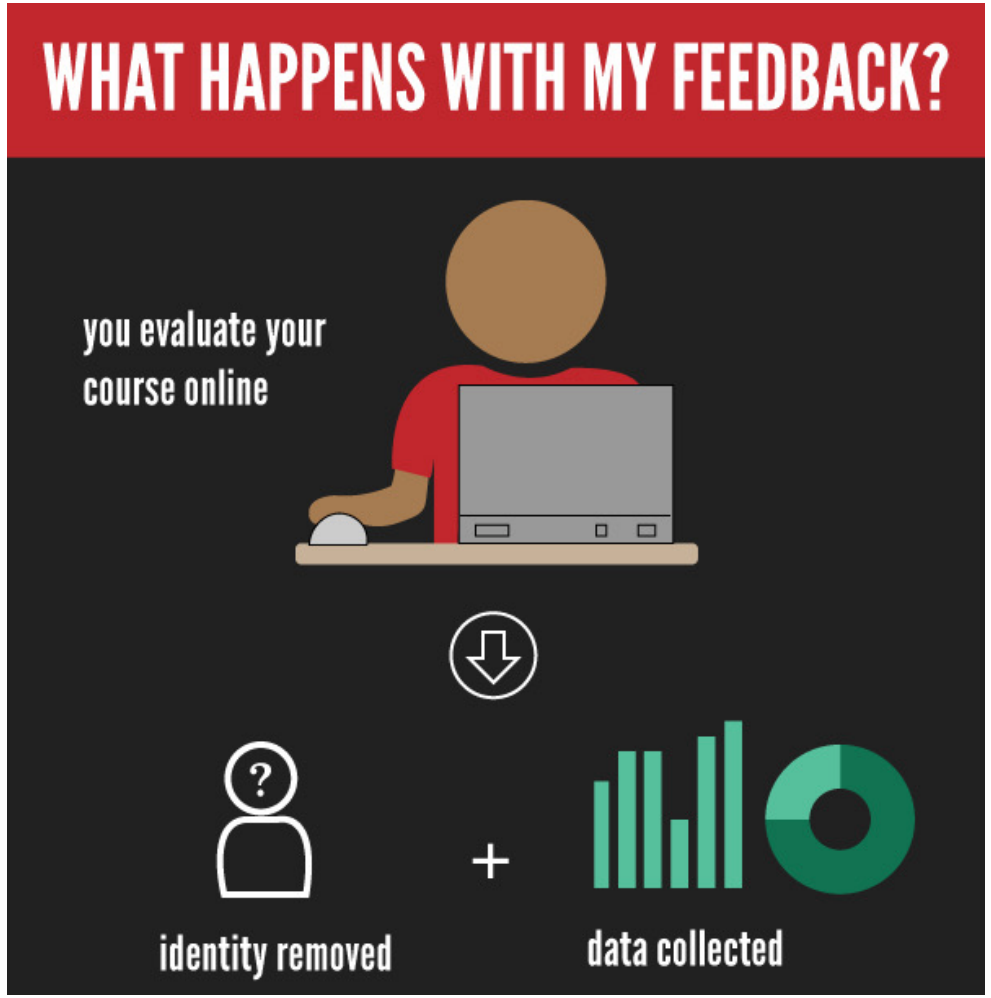


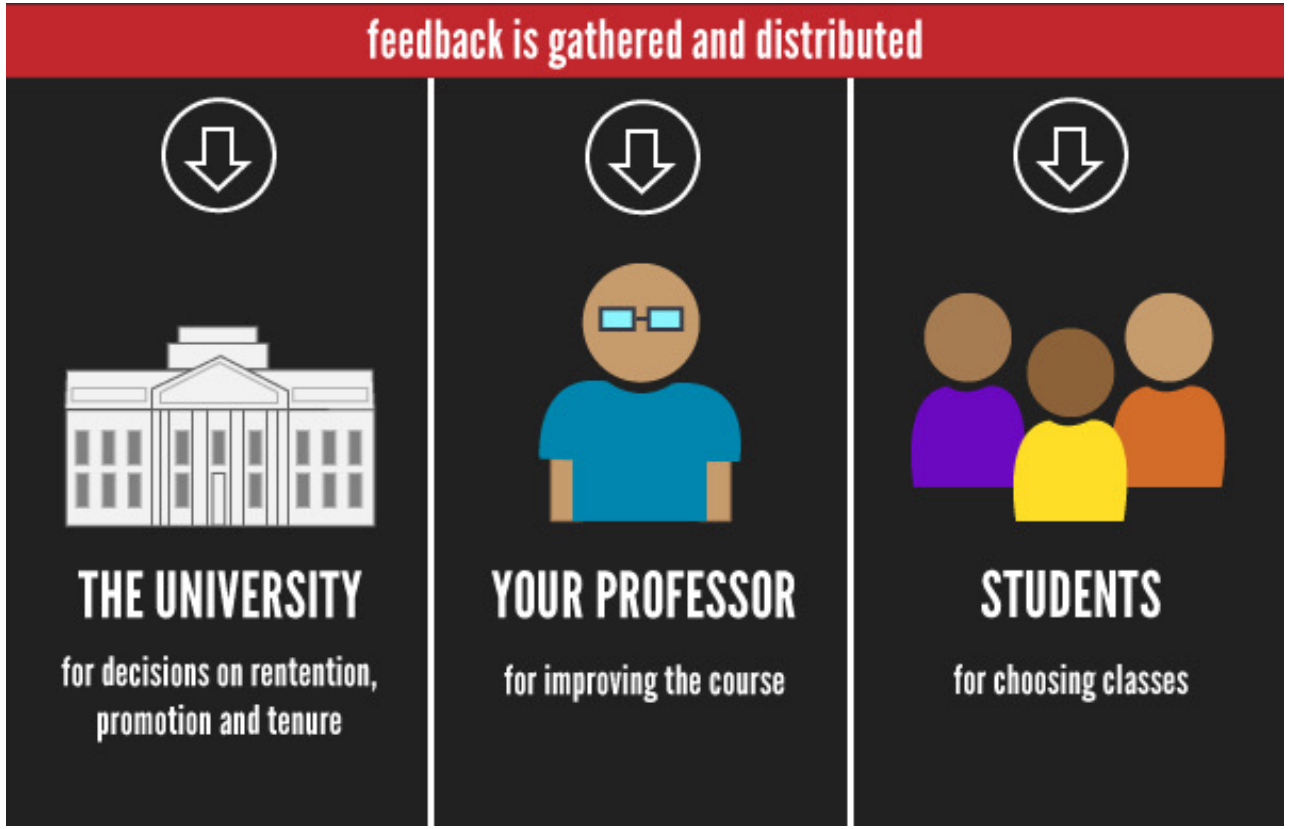


Images fromhttps://bl.ocks.org/mbostock/1005873,
https://bl.ocks.org/mbostock/4348373

Student Course Feedback

https://scf.utah.edu

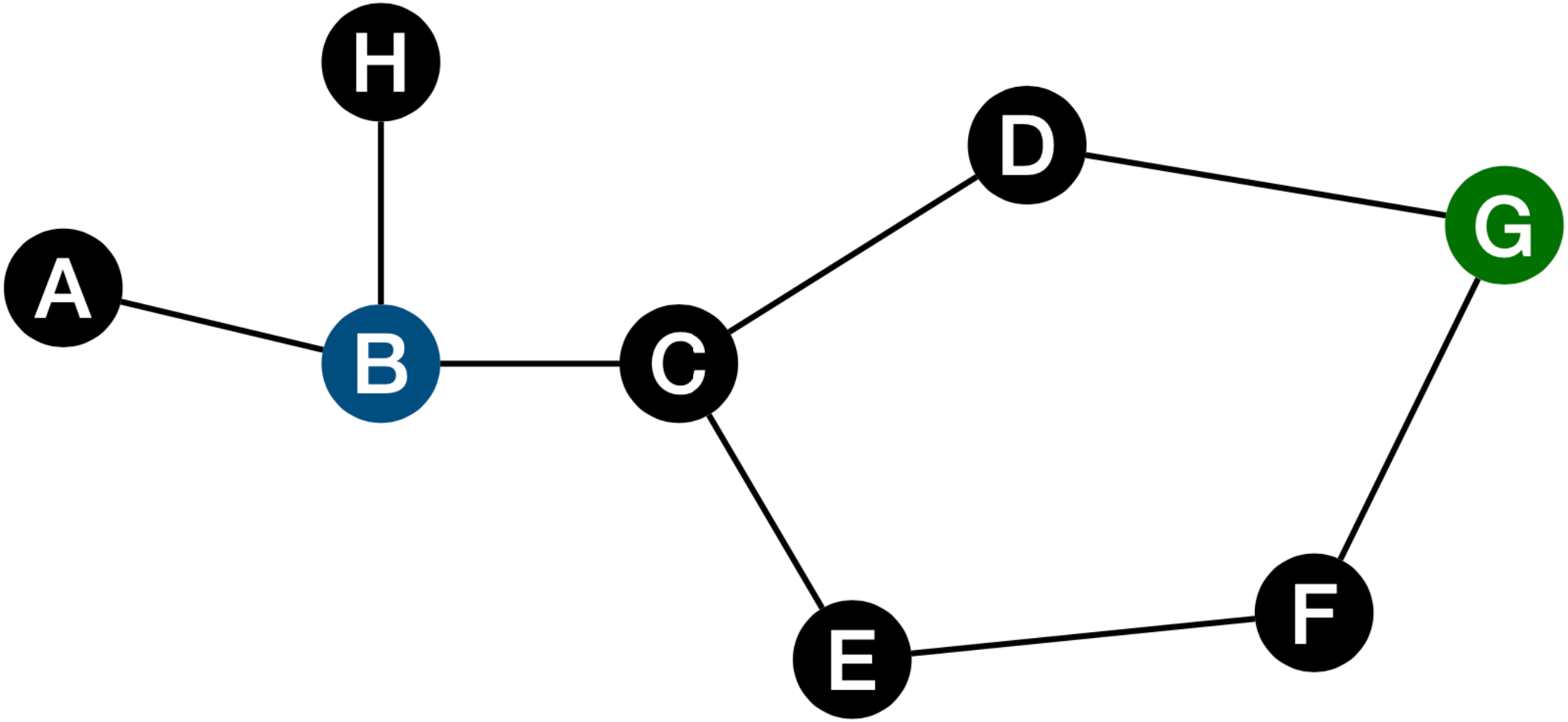




Shortest Path Algorithms

Breadth-First Search Example

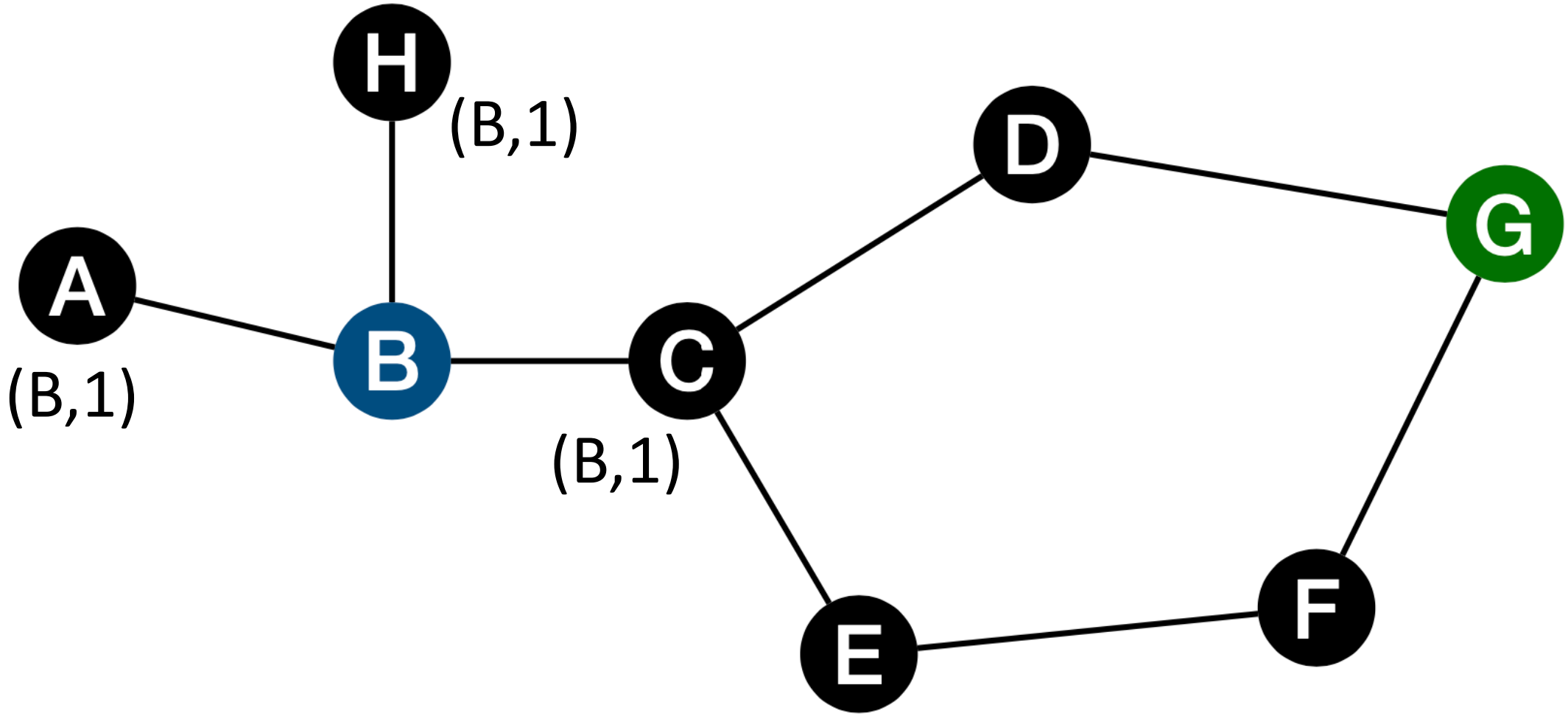
Shortest path from B to G



[B]

Breadth-First Search Example

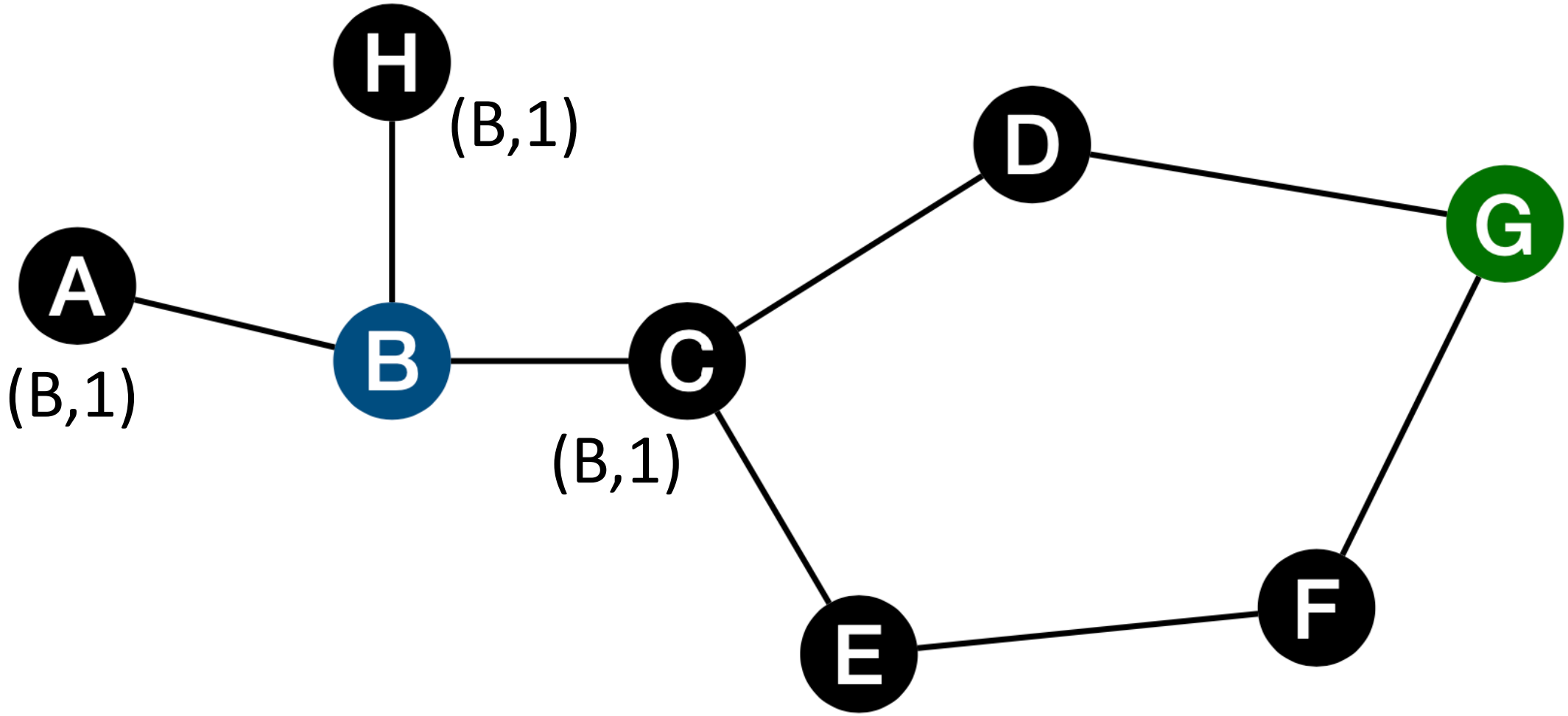
Shortest path from B to G



[~~B~~, A, H, C]

Breadth-First Search Example

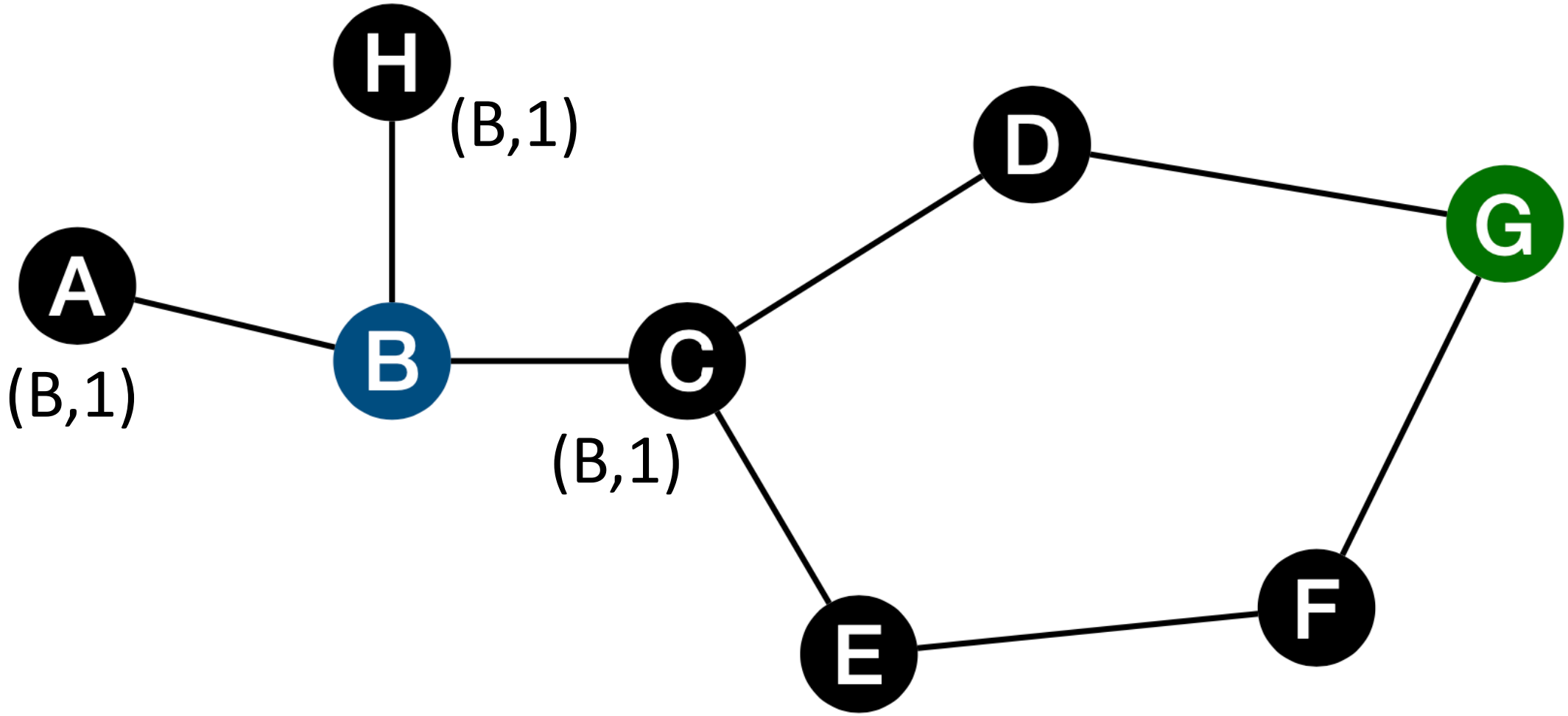
Shortest path from B to G



[~~B~~, ~~A~~, H, C]

Breadth-First Search Example

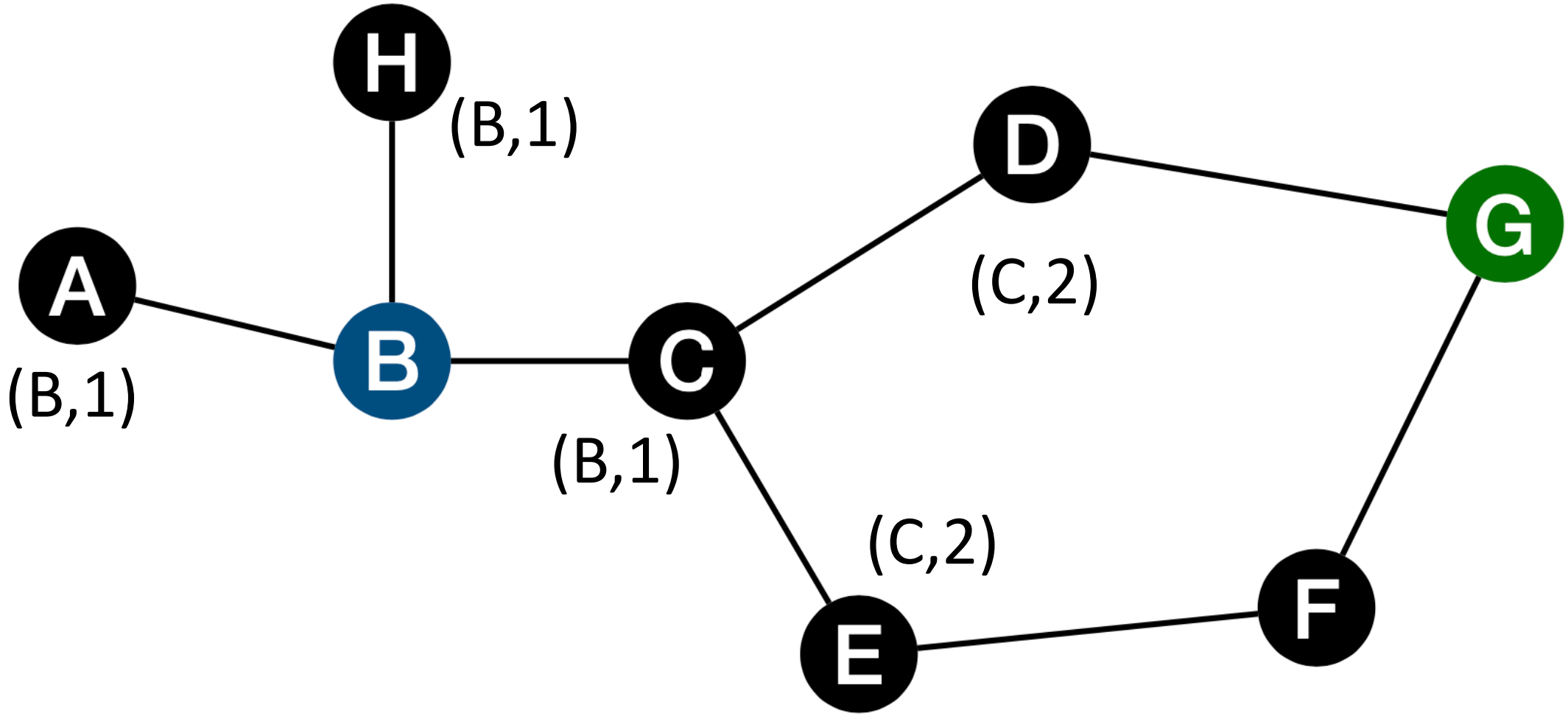
Shortest path from B to G



[~~B~~, ~~A~~, ~~H~~, C]

Breadth-First Search Example

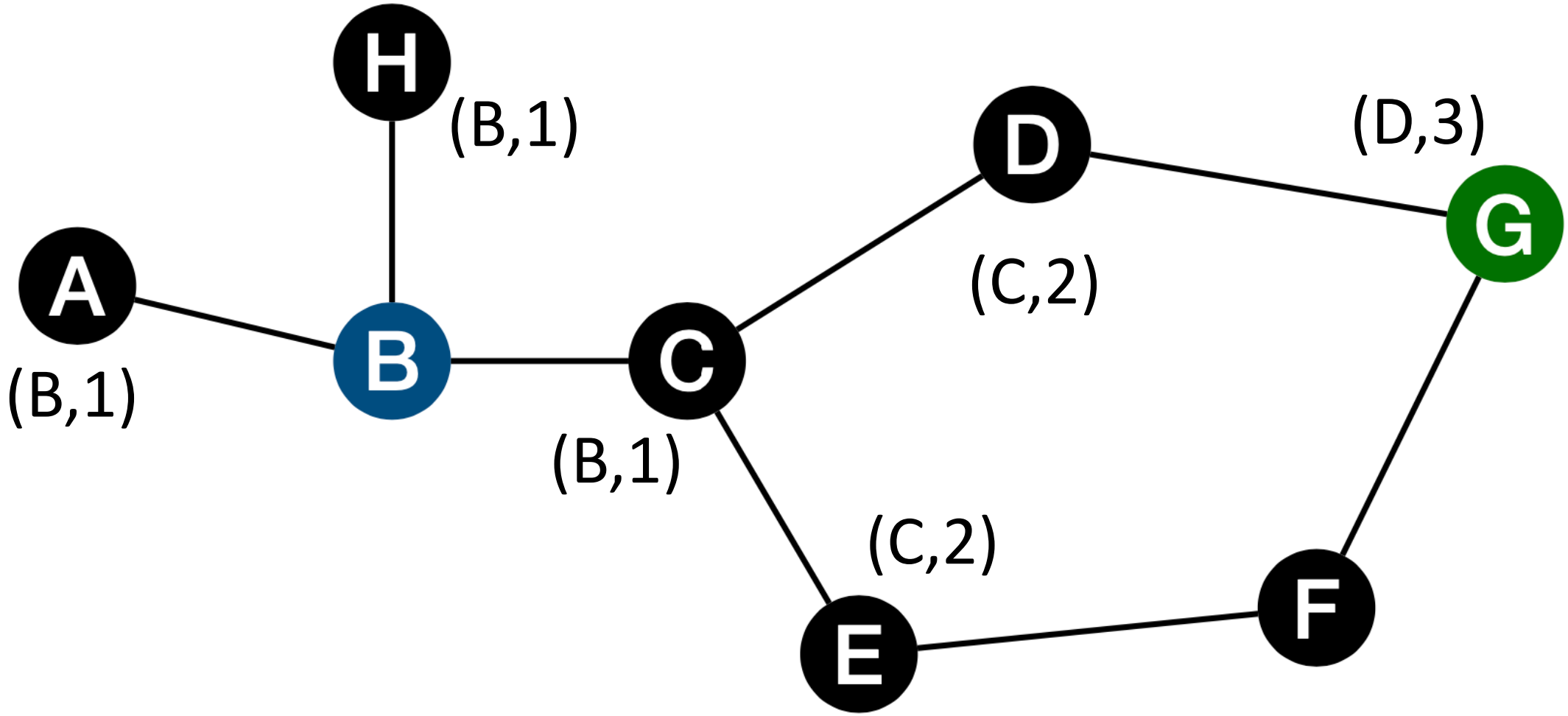
Shortest path from B to G



[~~B~~, ~~A~~, ~~H~~, ~~C~~, D, E]

Breadth-First Search Example

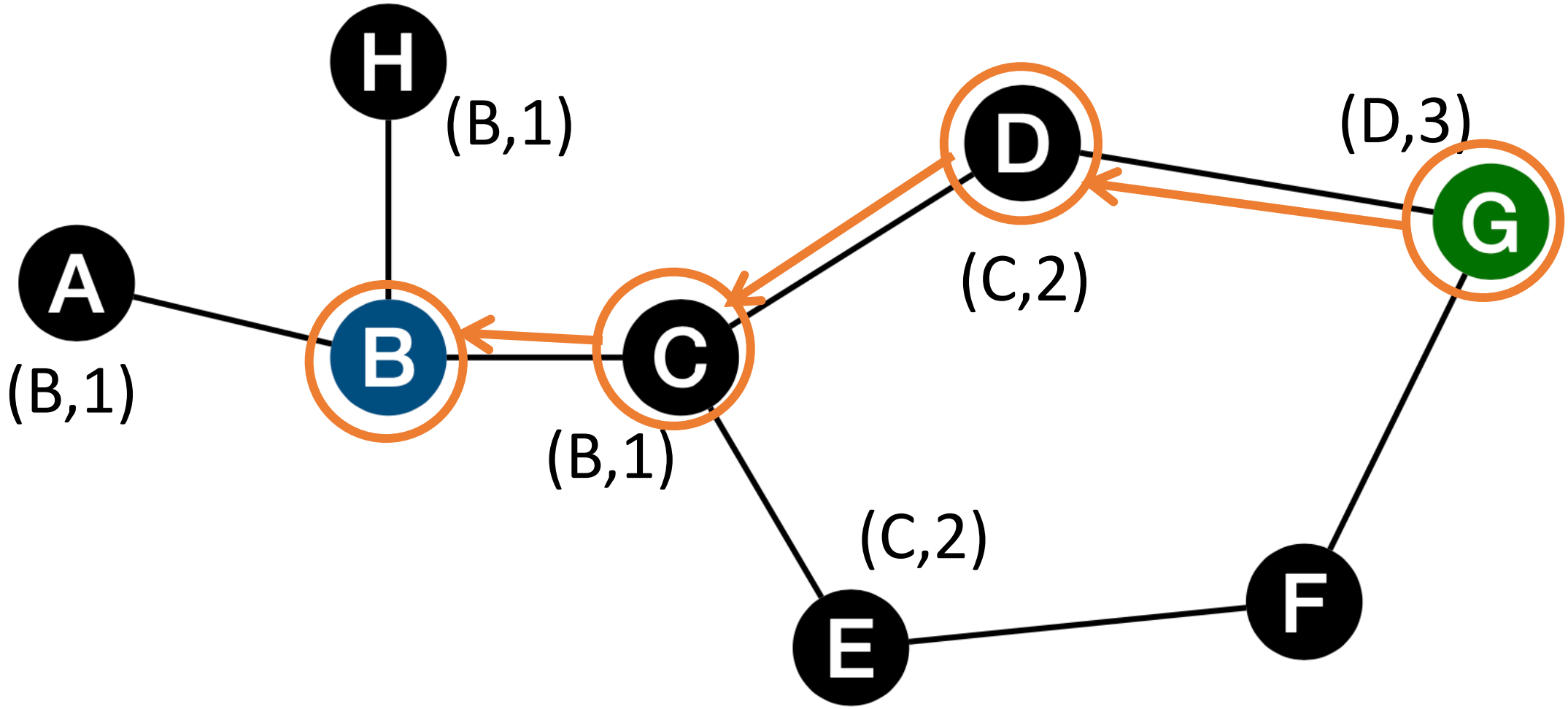
Shortest path from B to G



[~~B~~, ~~A~~, ~~H~~, ~~C~~, ~~D~~, E, **G**]

Breadth-First Search Example

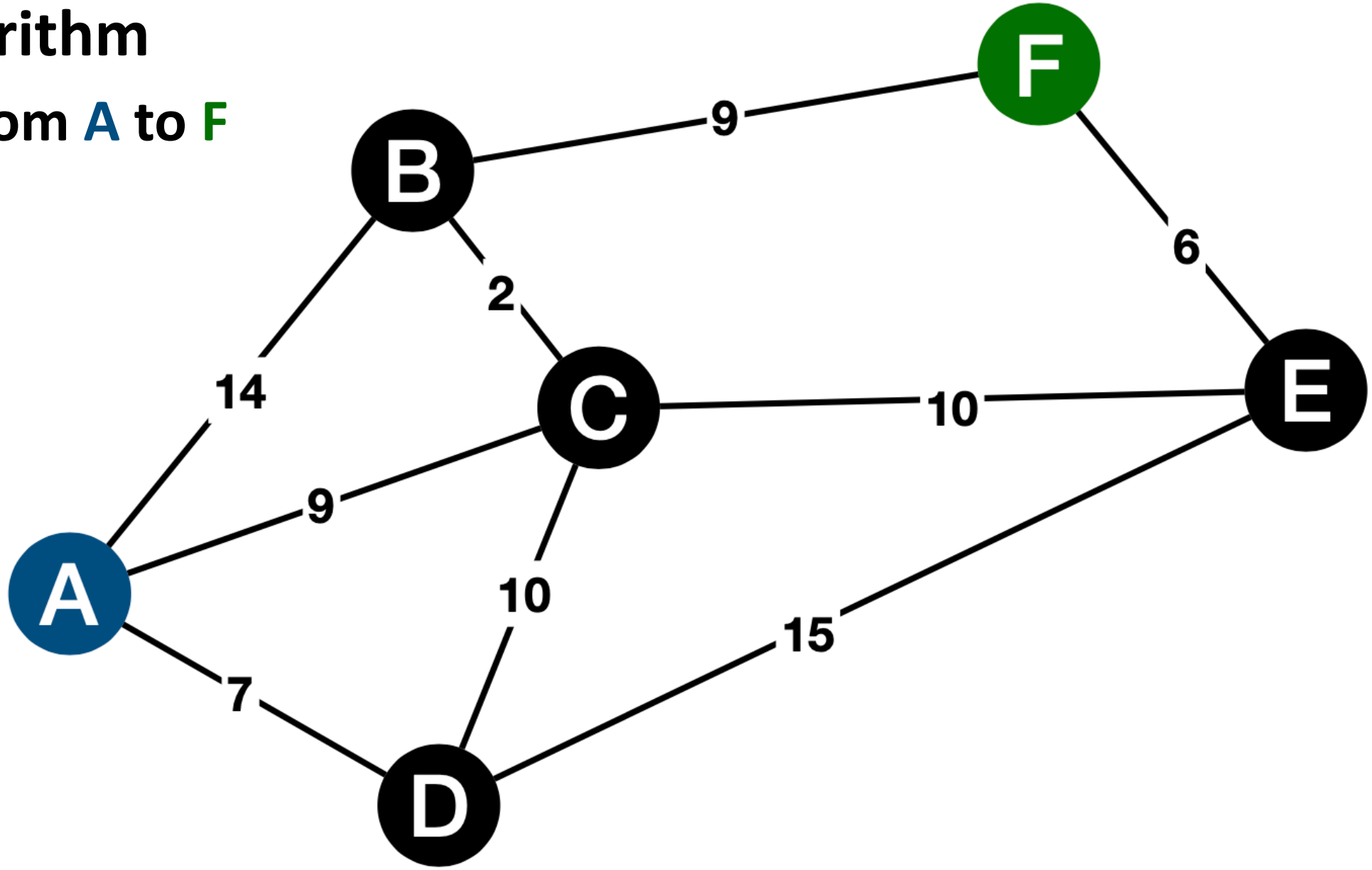
Shortest path from B to G



[~~B~~, ~~A~~, ~~H~~, ~~C~~, ~~D~~, E, **G**]

Dijkstra's Algorithm

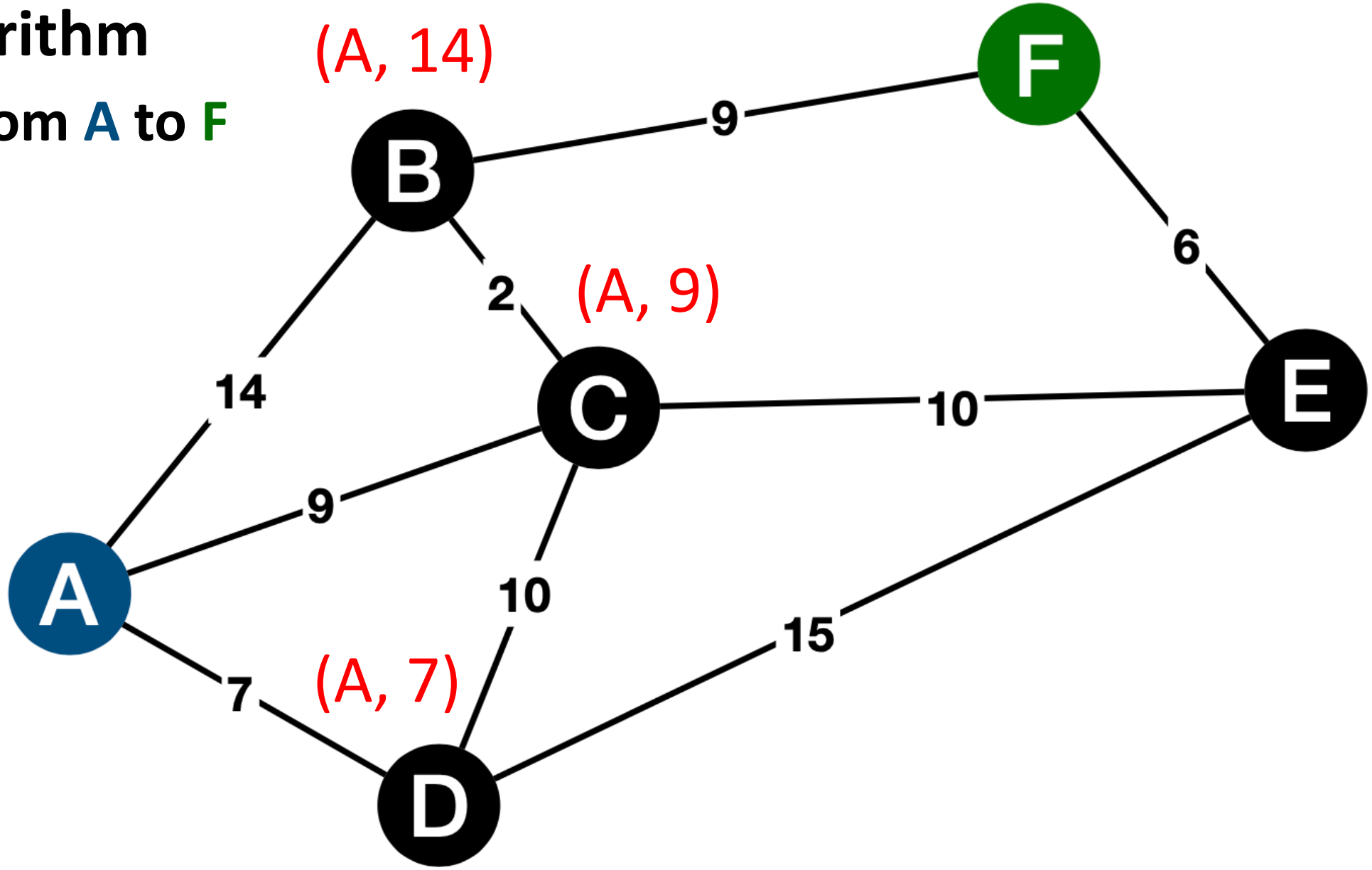
Shortest path from A to F



[A]

Dijkstra's Algorithm

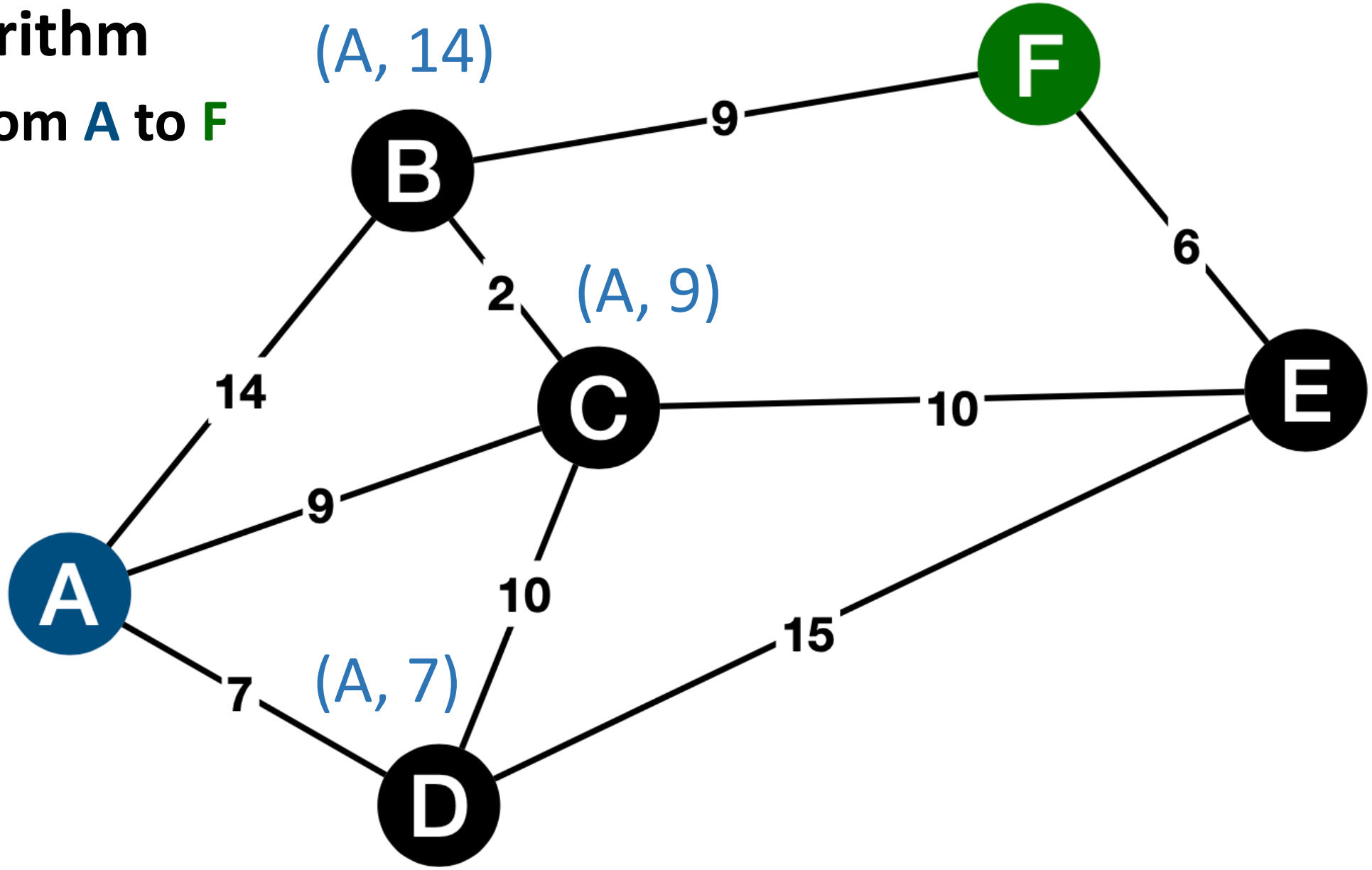
Shortest path from A to F



[**A**, D(7), C(9), B(14)]

Dijkstra's Algorithm

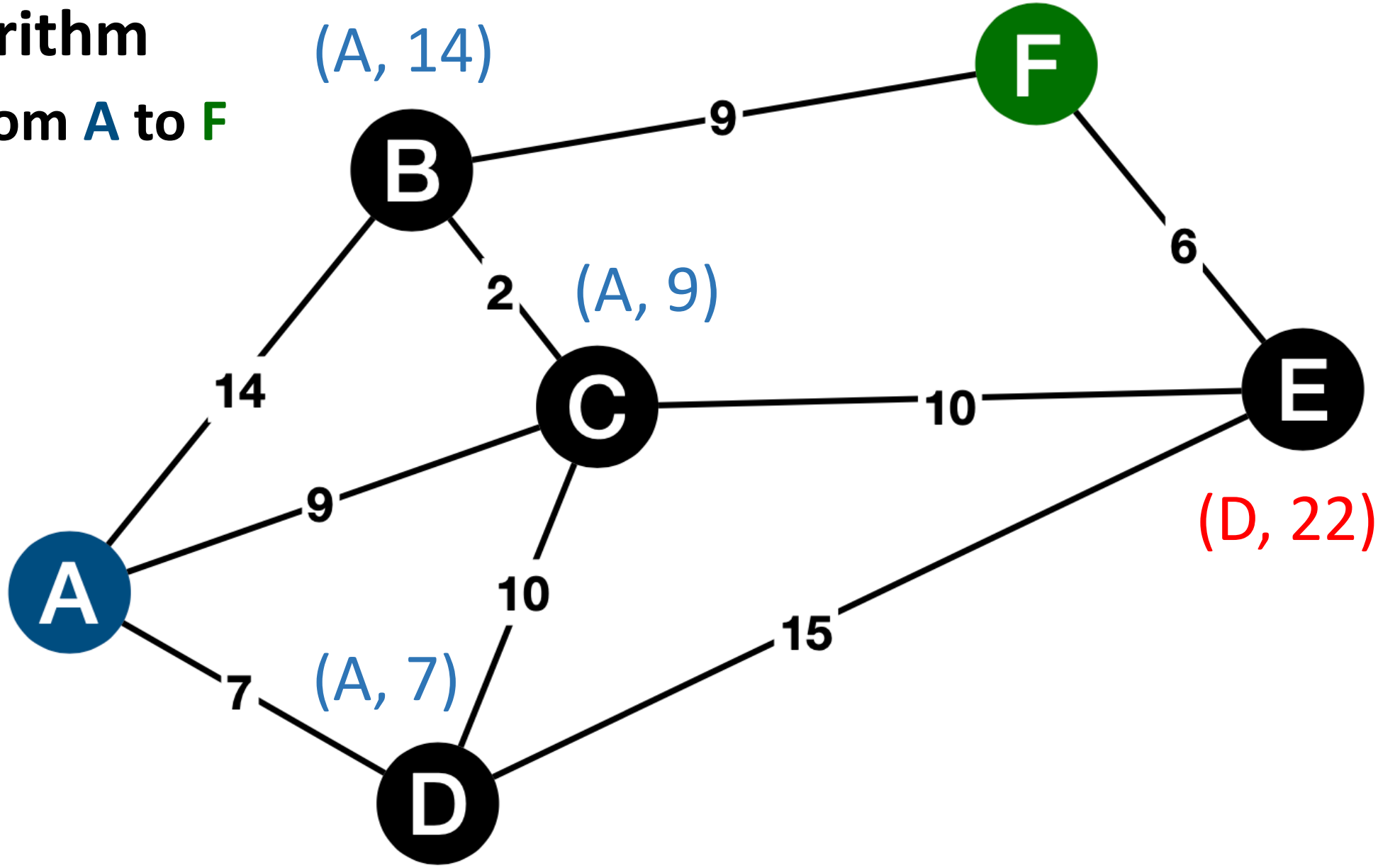
Shortest path from A to F



[~~A~~, D(7), C(9), B(14)]

Dijkstra's Algorithm

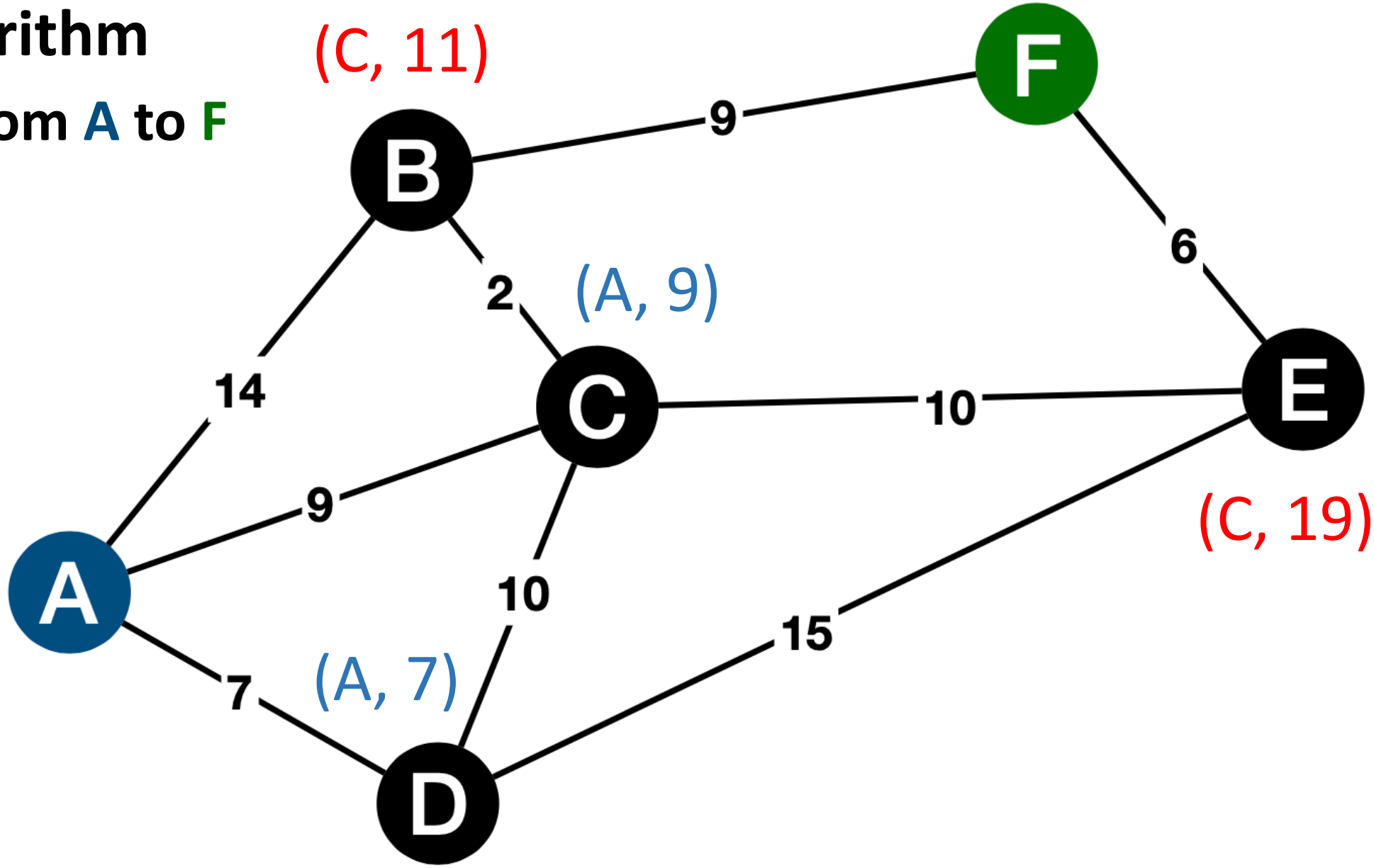
Shortest path from A to F



[~~A~~, **D(7)**, C(9), B(14), E(22)]

Dijkstra's Algorithm

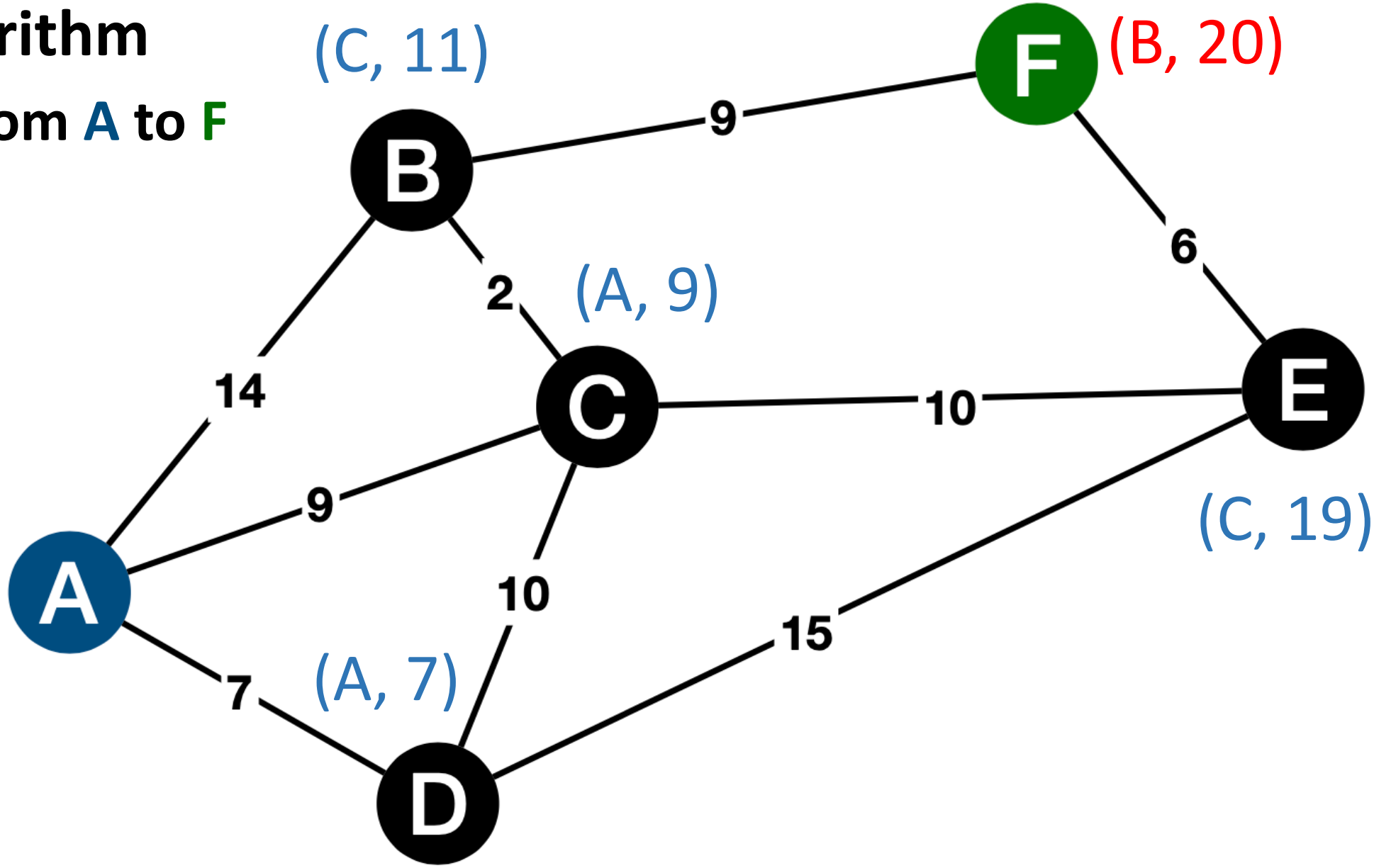
Shortest path from A to F



[~~A~~, ~~D(7)~~, **C(9)**, B(11), E(19)]

Dijkstra's Algorithm

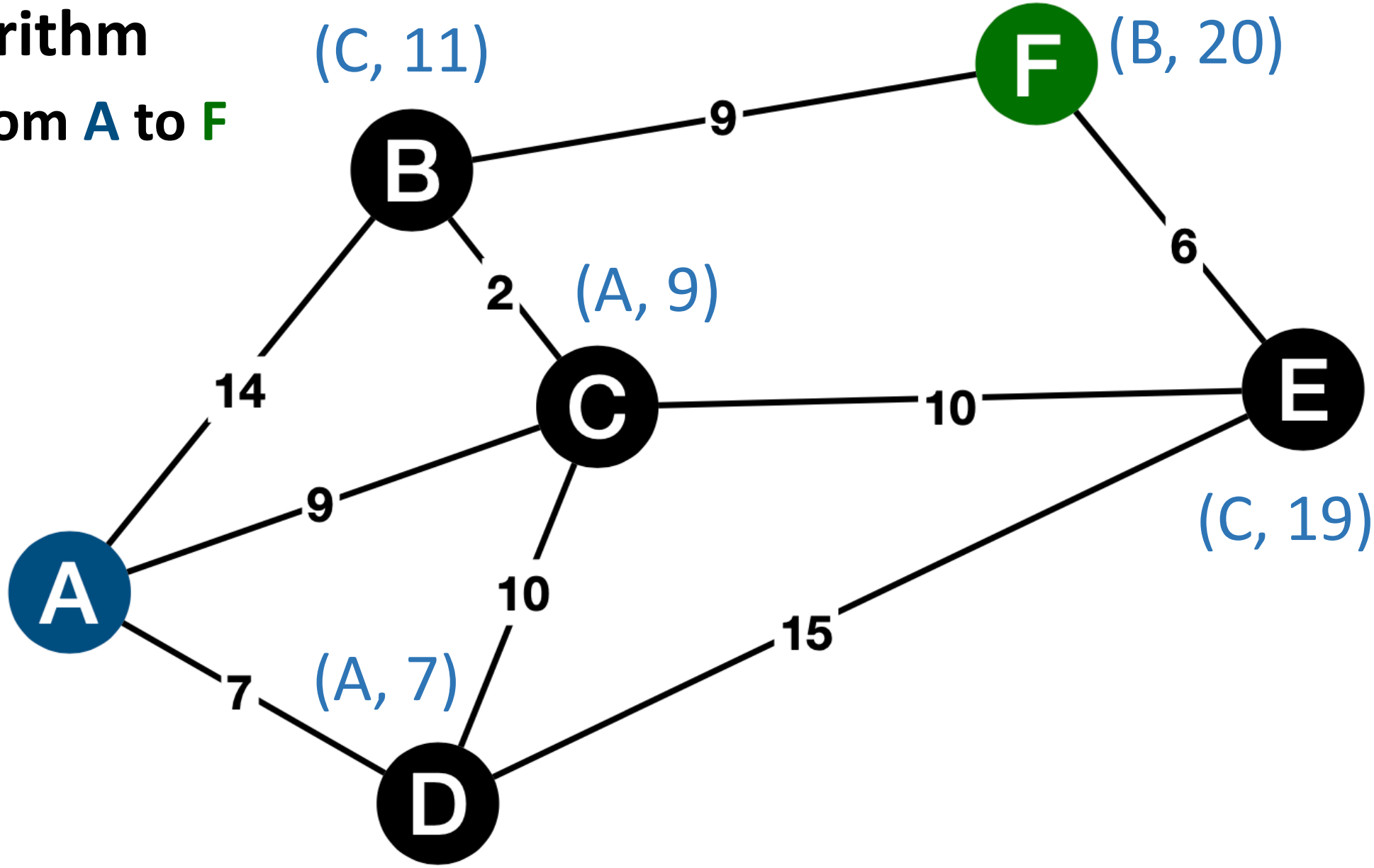
Shortest path from A to F



[~~A~~, ~~D(7)~~, ~~C(9)~~, **B(11)**, E(19), F(20)]

Dijkstra's Algorithm

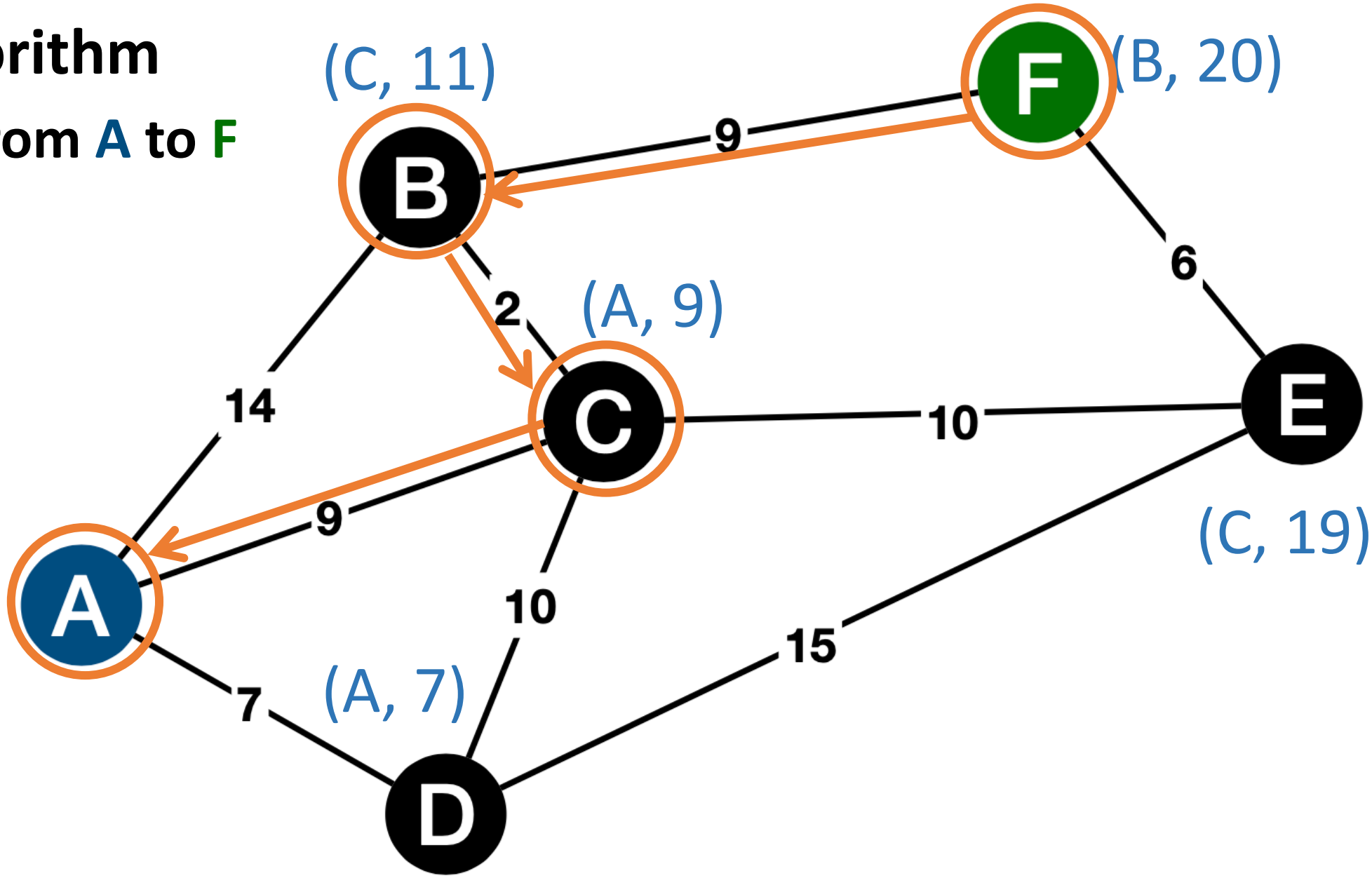
Shortest path from A to F



[~~A~~, ~~D(7)~~, ~~C(9)~~, ~~B(11)~~, **E(19)**, F(20)]

Dijkstra's Algorithm

Shortest path from A to F



[~~A~~, ~~D(7)~~, ~~C(9)~~, ~~B(11)~~, ~~E(19)~~, **F(20)**]