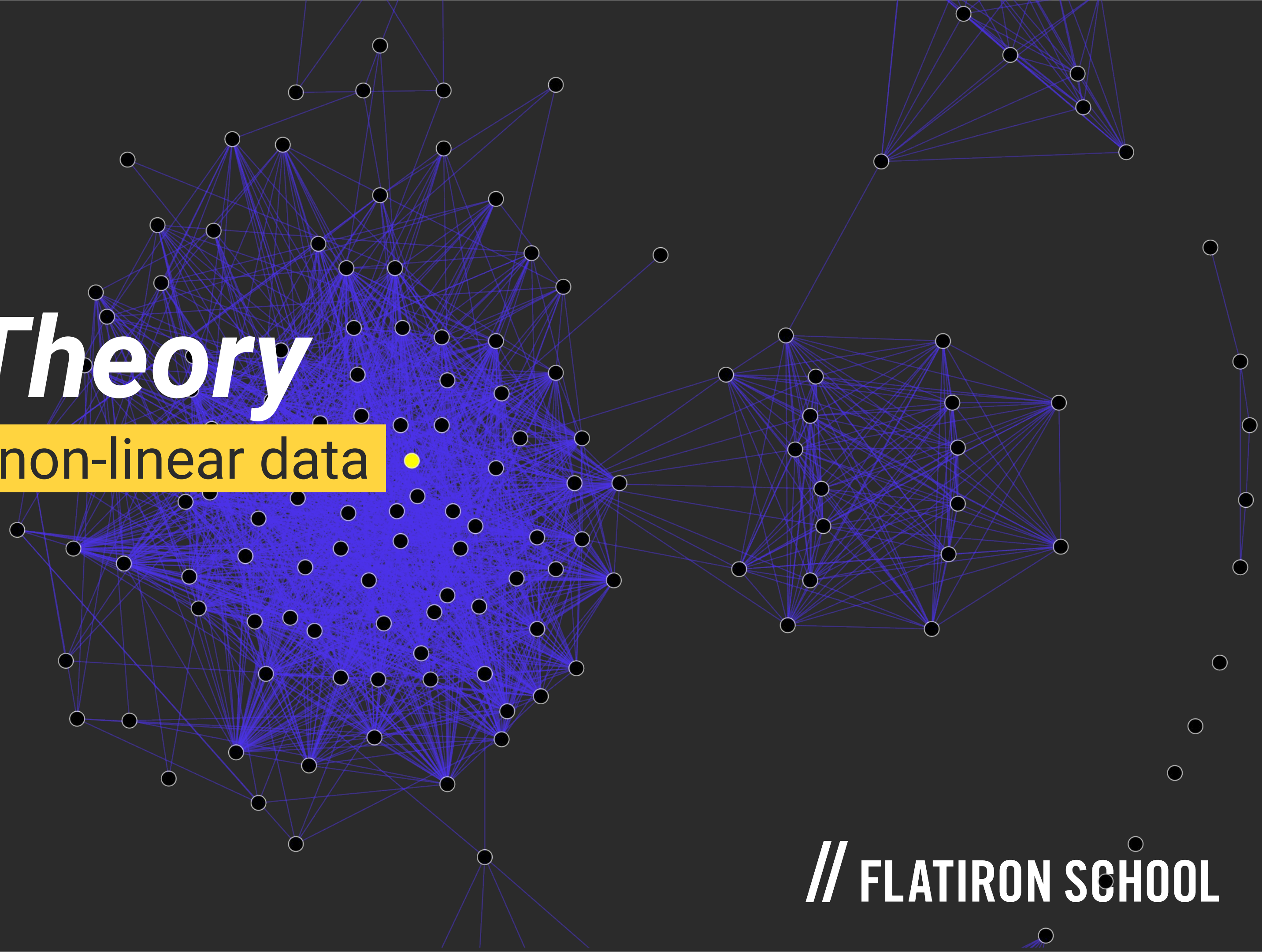
// FLATIRON SCHOOL

Graph Theory
Presented by Wachira Ndaiga
March 19, 2020

Graph Theory

Representing non-linear data



FLATIRON SCHOOL

Outline

Introduction (30 min)

Breakout/TPS (10 mins)

Break (5 mins)

Practical (45 mins)

Learning Outcomes

- Data Structures
- Graph Representations
- Graph Terminology & Analysis
- NetworkX

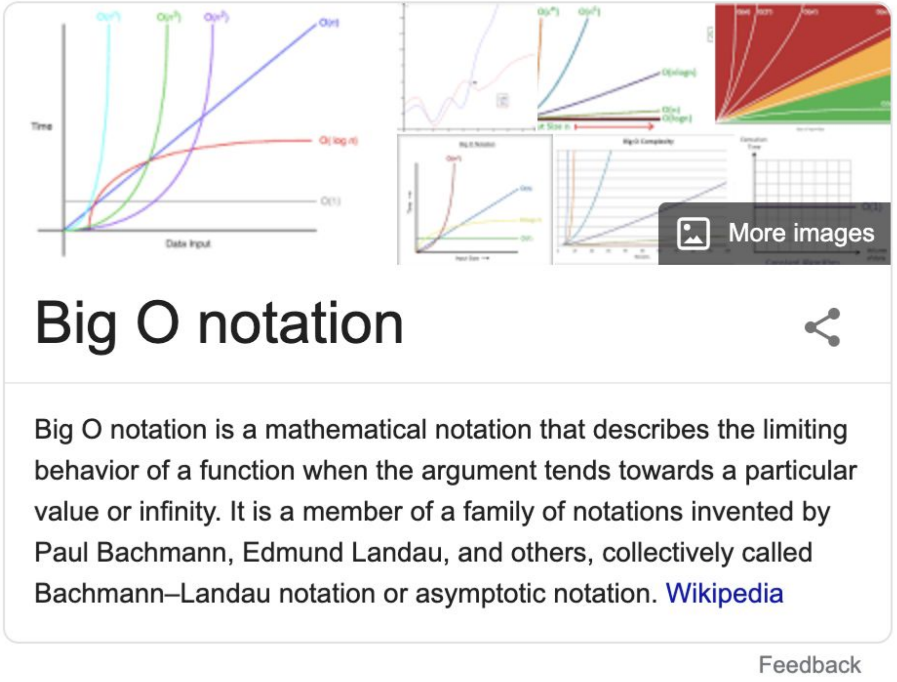
Data Structures

Definition

Data structures are a way of **organizing** and **storing data** so it can be **accessed** and **modified** *efficiently.

Breakout

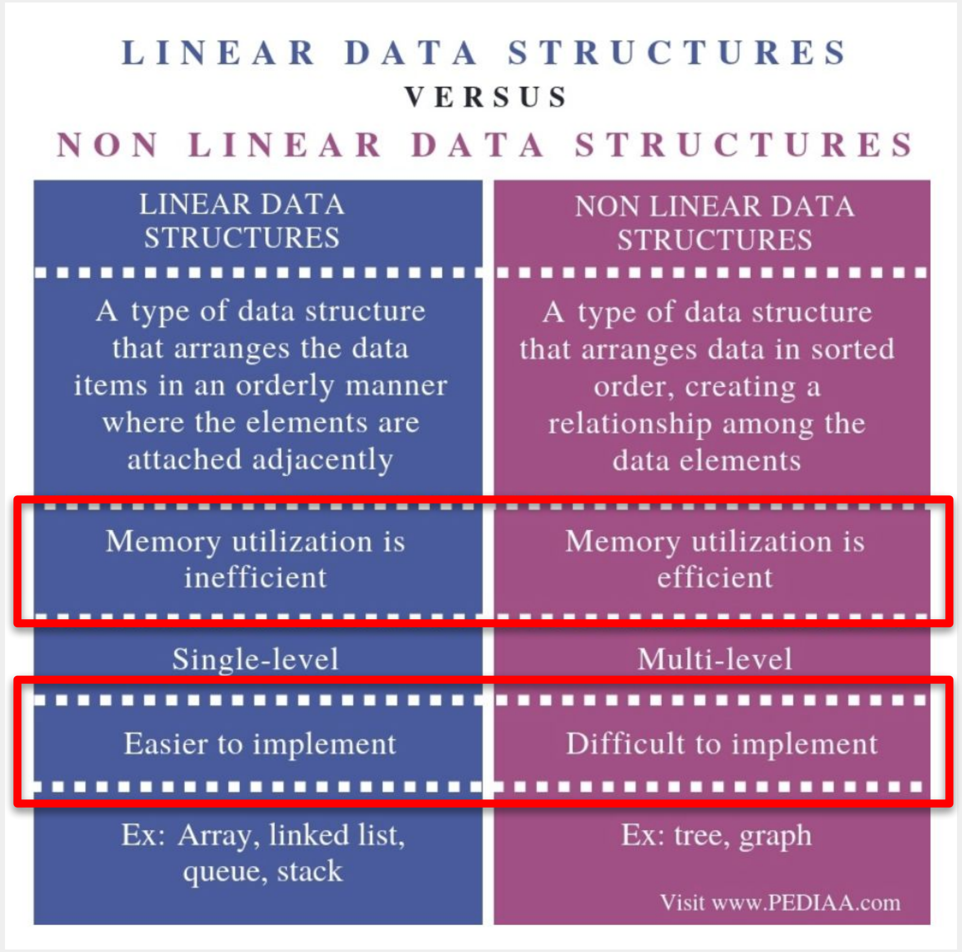
1. What types of data structures exist in python?
2. What limitations have you come across when using ***native*** data structures like lists and dictionaries?
3. Give an example of a custom data structure you might have seen used or heard of before.


More images

Big O notation

Big O notation is a mathematical notation that describes the limiting behavior of a function when the argument tends towards a particular value or infinity. It is a member of a family of notations invented by Paul Bachmann, Edmund Landau, and others, collectively called Bachmann–Landau notation or asymptotic notation. Wikipedia

Feedback


LINEAR DATA STRUCTURES
VERSUS
NON LINEAR DATA STRUCTURES

LINEAR DATA STRUCTURES	NON LINEAR DATA STRUCTURES
A type of data structure that arranges the data items in an orderly manner where the elements are attached adjacently	A type of data structure that arranges data in sorted order, creating a relationship among the data elements
Memory utilization is inefficient	Memory utilization is efficient
Single-level	Multi-level
Easier to implement	Difficult to implement
Ex: Array, linked list, queue, stack	Ex: tree, graph

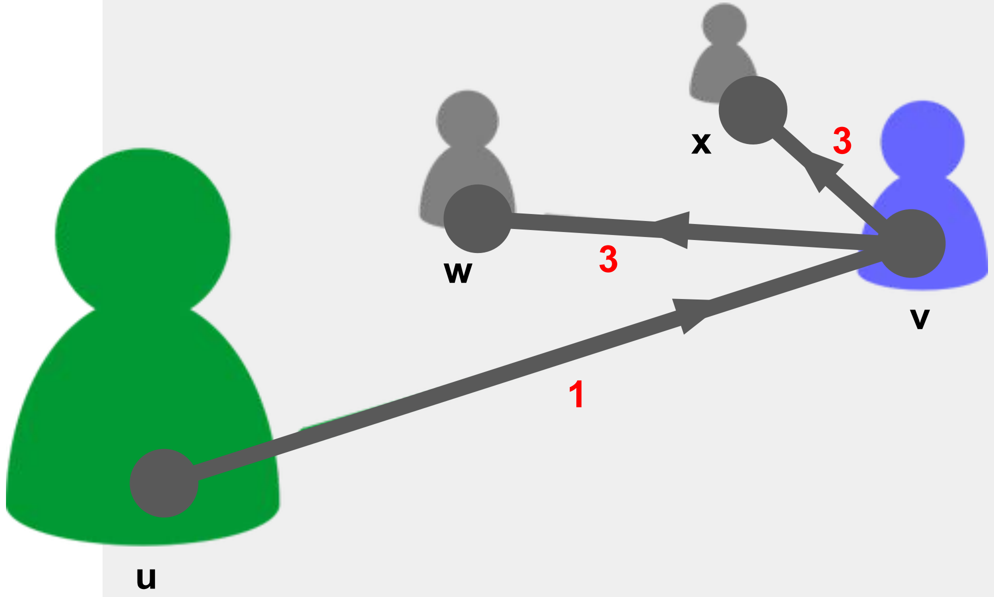
Visit www.PEDIAA.com

Graphs

Definition

Types of Graphs:

1. Undirected Graphs
2. Directed Graphs
3. Unipartite Graphs
4. Bipartite Graphs

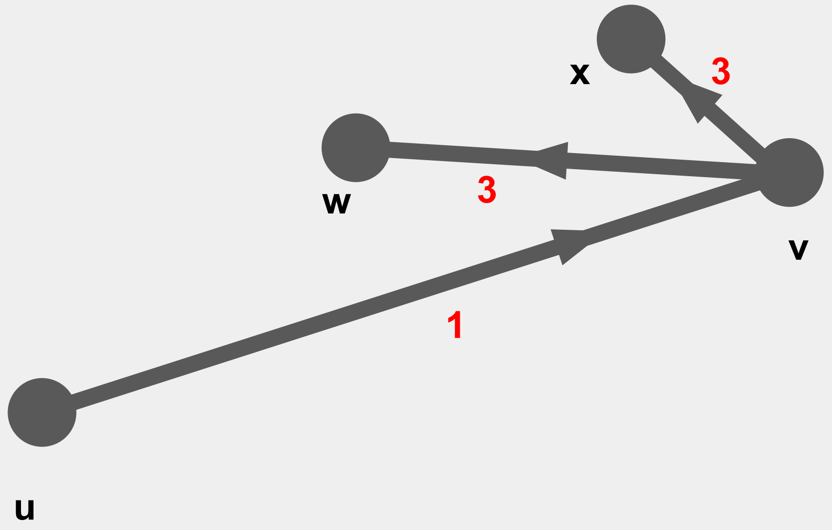


Graphs

Representations

There are a number of ways to represent graphs mathematically:

- **Adjacency Matrix**
- Adjacency List
- Incidence List
- Incidence Matrix

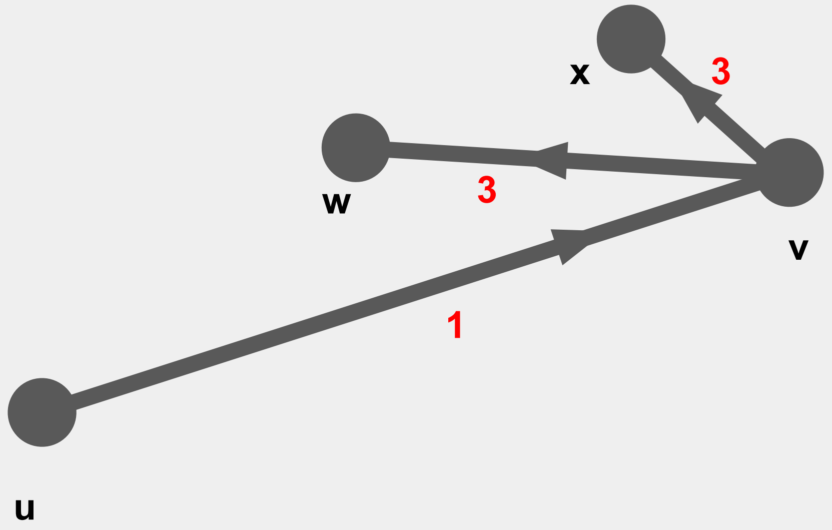


Graphs

Representations

Adjacency Matrix

An adjacency matrix is a **2D** array of **size V x V** where V is the number of vertices in a graph.

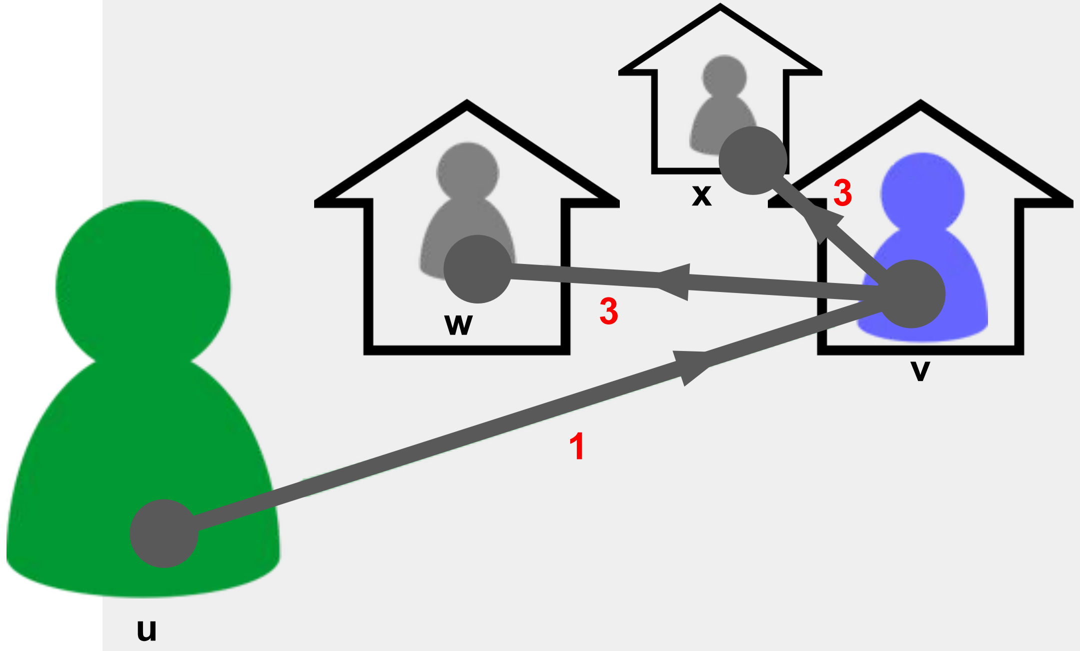


Breakout

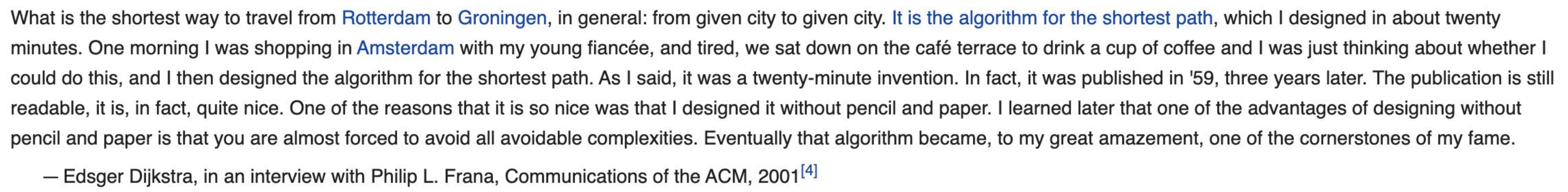
Graphs

Representations

	u	v	w	x
u	0	1	0	0
v	0	0	3	3
w	0	0	0	0
x	0	0	0	0

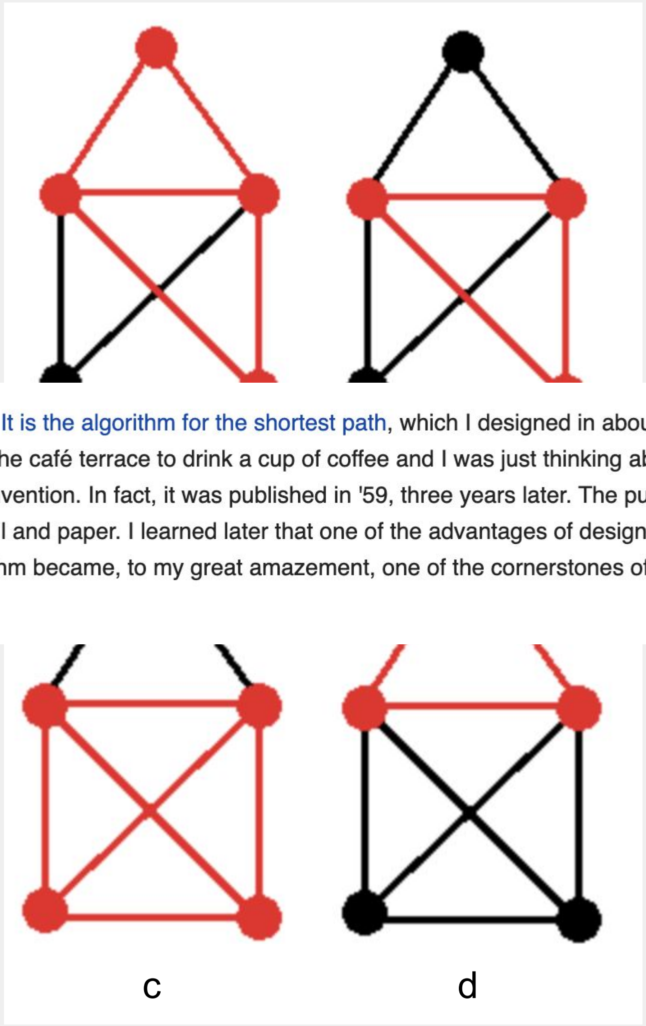


Graphs


What is the shortest way to travel from Rotterdam to Groningen, in general: from given city to given city. It is the algorithm for the shortest path, which I designed in about twenty minutes. One morning I was shopping in Amsterdam with my young fiancée, and tired, we sat down on the café terrace to drink a cup of coffee and I was just thinking about whether I could do this, and I then designed the algorithm for the shortest path. As I said, it was a twenty-minute invention. In fact, it was published in '59, three years later. The publication is still readable, it is, in fact, quite nice. One of the reasons that it is so nice was that I designed it without pencil and paper. I learned later that one of the advantages of designing without pencil and paper is that you are almost forced to avoid all avoidable complexities. Eventually that algorithm became, to my great amazement, one of the cornerstones of my fame.

— Edsger Dijkstra, in an interview with Philip L. Frana, Communications of the ACM, 2001[4]

- Path
- Clique
- Centrality
    - Degree
    - Betweeneness
    - Closeness
- Travelling/Traversal

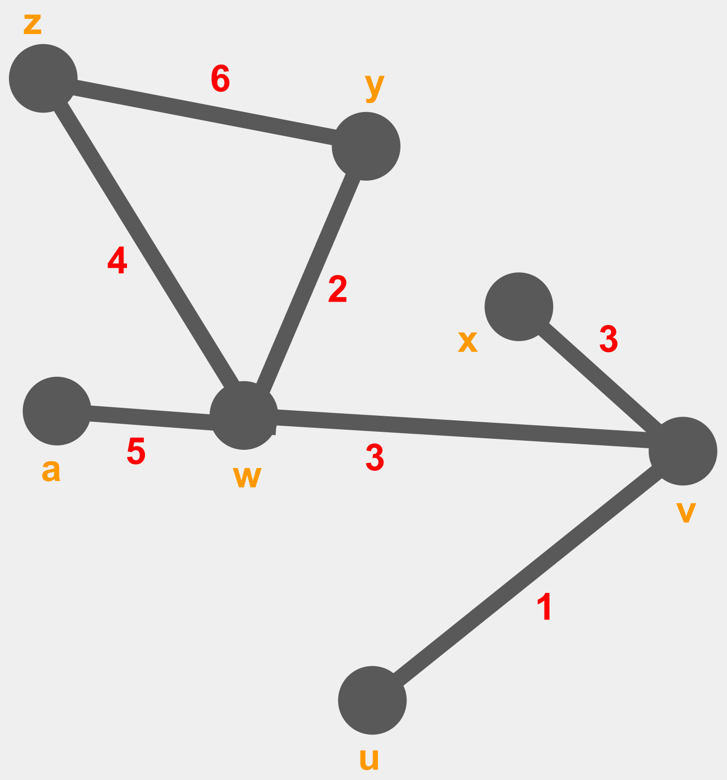


Graphs

Analysis

1. Represent the graph to the right with an Adjacency Matrix.
2. How many cliques are there in the graph?
3. What is the shortest path from source **z** to target **u**?
4. What are the centrality measures for the graph to the right?

Breakout



Recap

1. What domains are represented well with graphs?

2. What are cliques and paths in what interesting ways can we use them?

3. How do we analyse the performance of data structures?

4. What are the popular graph traversal algorithms?

Q&A

Practical