

Graphs

CS 225 - Fall 2025

Mattox Beckman / Based on slides from Brad Solomon

Objectives

- Define vocabulary needed to describe graphs
- Implement a graph using an edge list
- Informal Early Feedback in the Discord

A **graph** G is a set V, E of **vertices** and **edges**

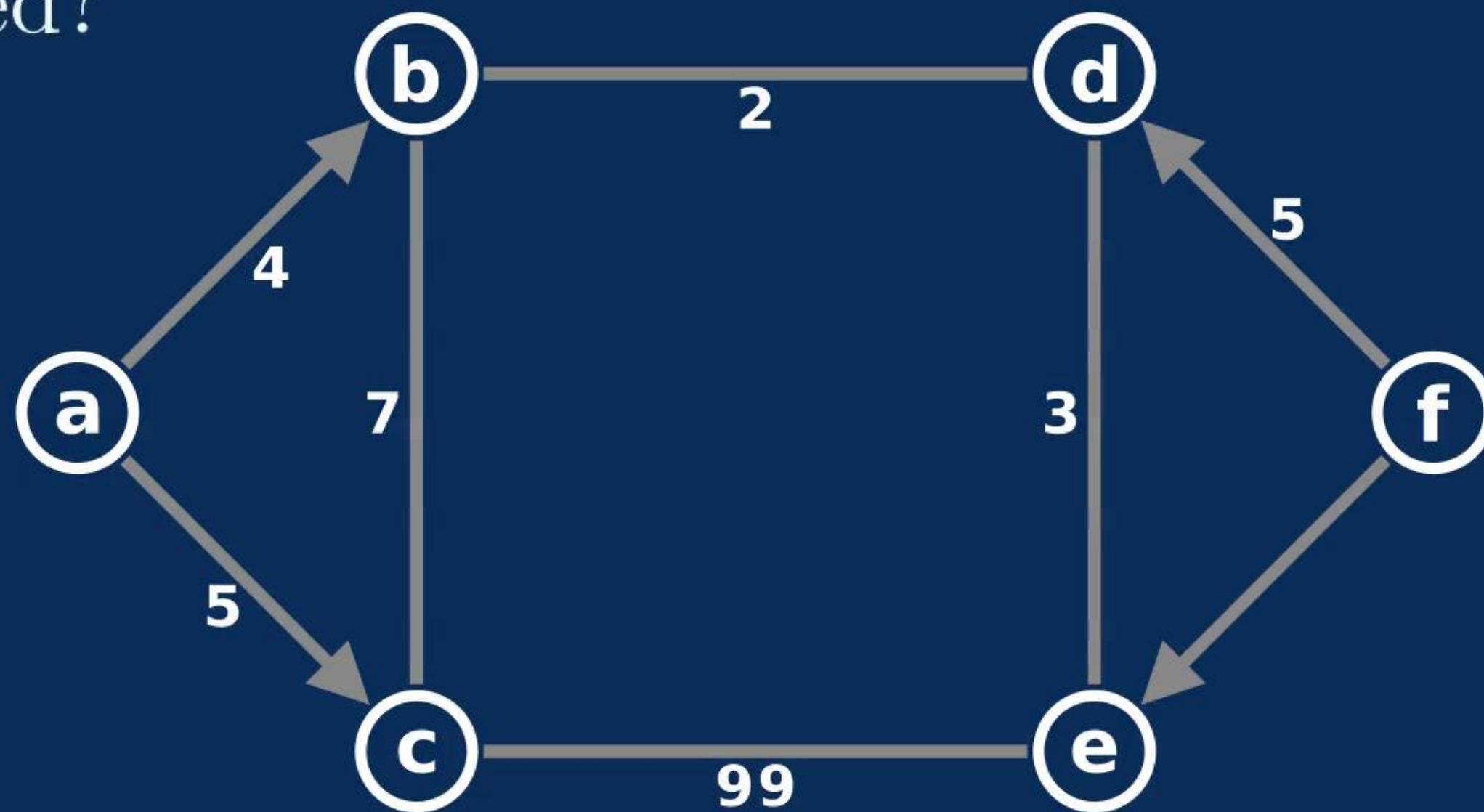
A **graph** G is a set V, E of **vertices** and **edges**

- Vertices also known as *nodes*.
- Can contain data; key-values; weights; etc.

A **graph** G is a set V, E of **vertices** and **edges**

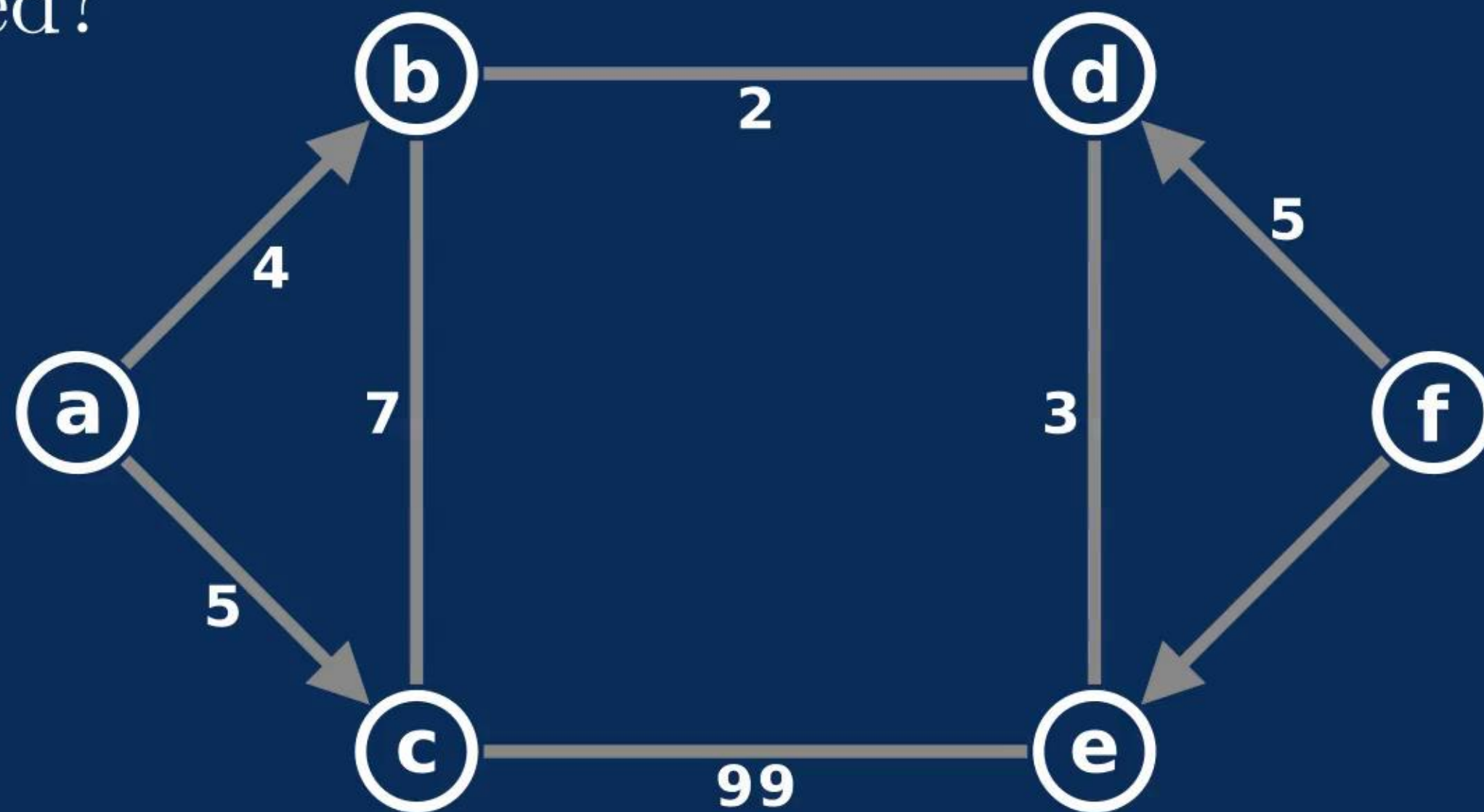
- Vertices also known as *nodes*.
- Can contain data; key-values; weights; etc.
- Edges connect nodes; defined by their endpoints.
- Edges can be directed or undirected, weighted or unweighted

Which edges are weighted?



Which edges are weighted?

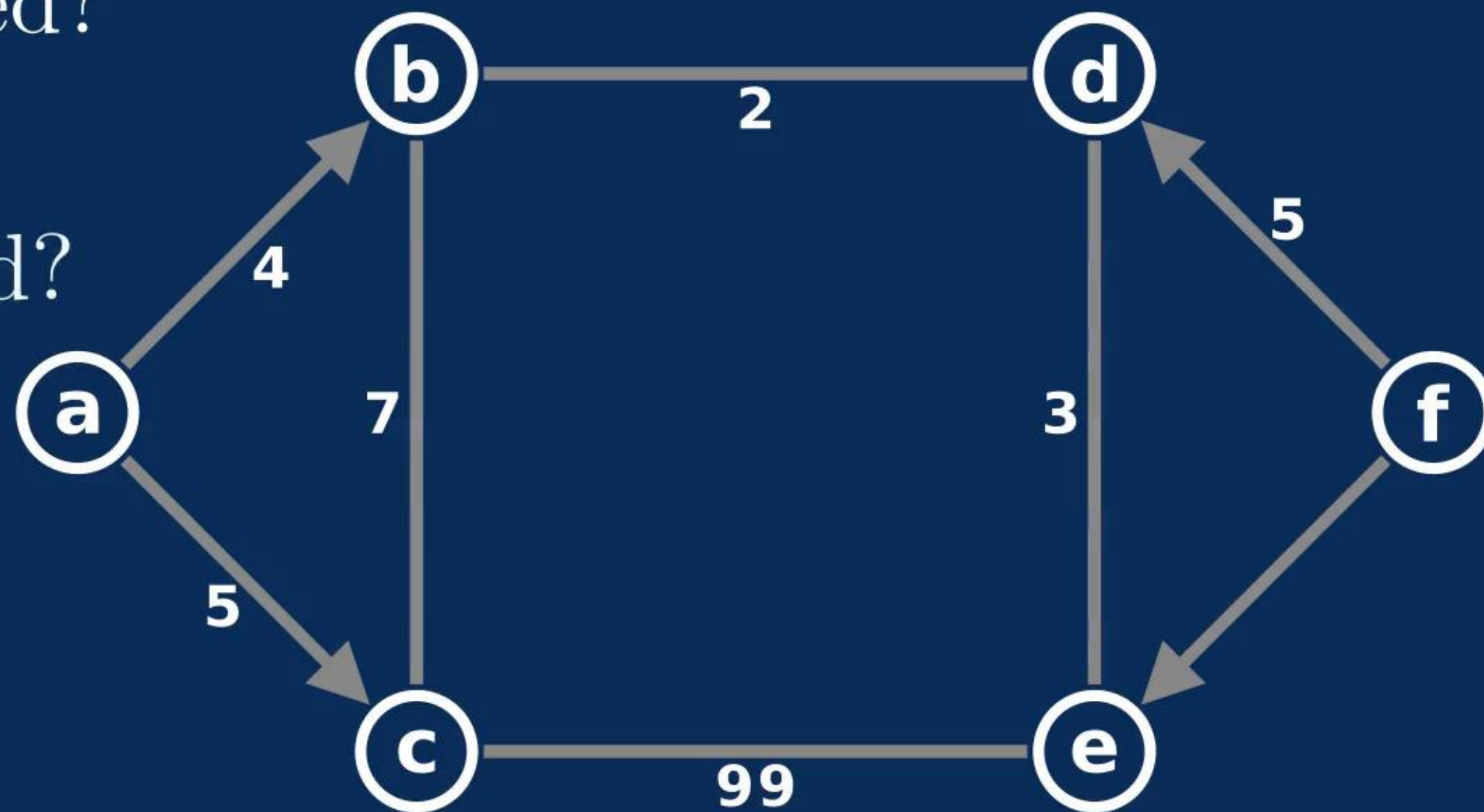
- All but f-e.



Which edges are weighted?

- All but f-e.

Which edges are directed?

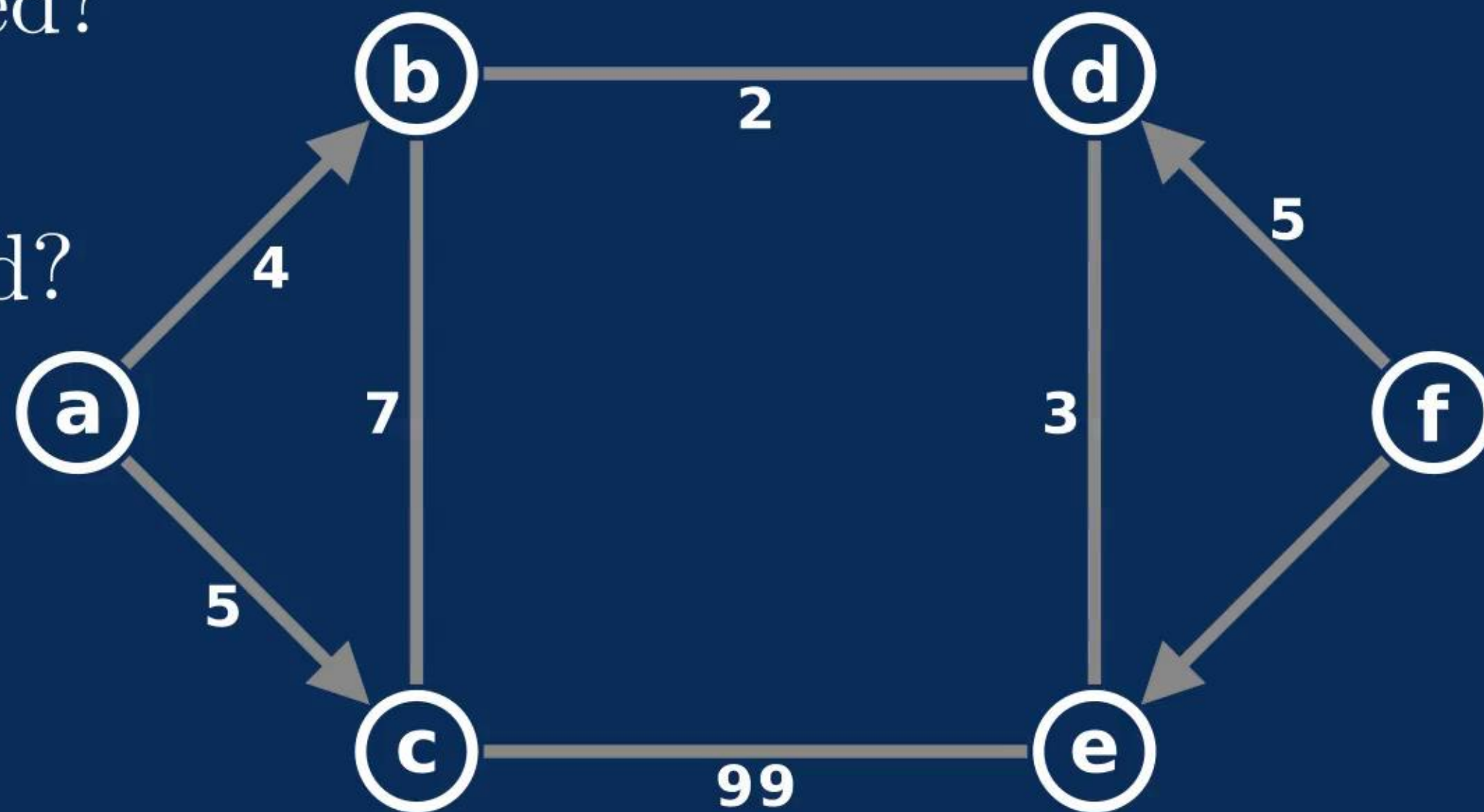


Which edges are weighted?

- All but f-e.

Which edges are directed?

- a-b, a-c, f-d, f-e

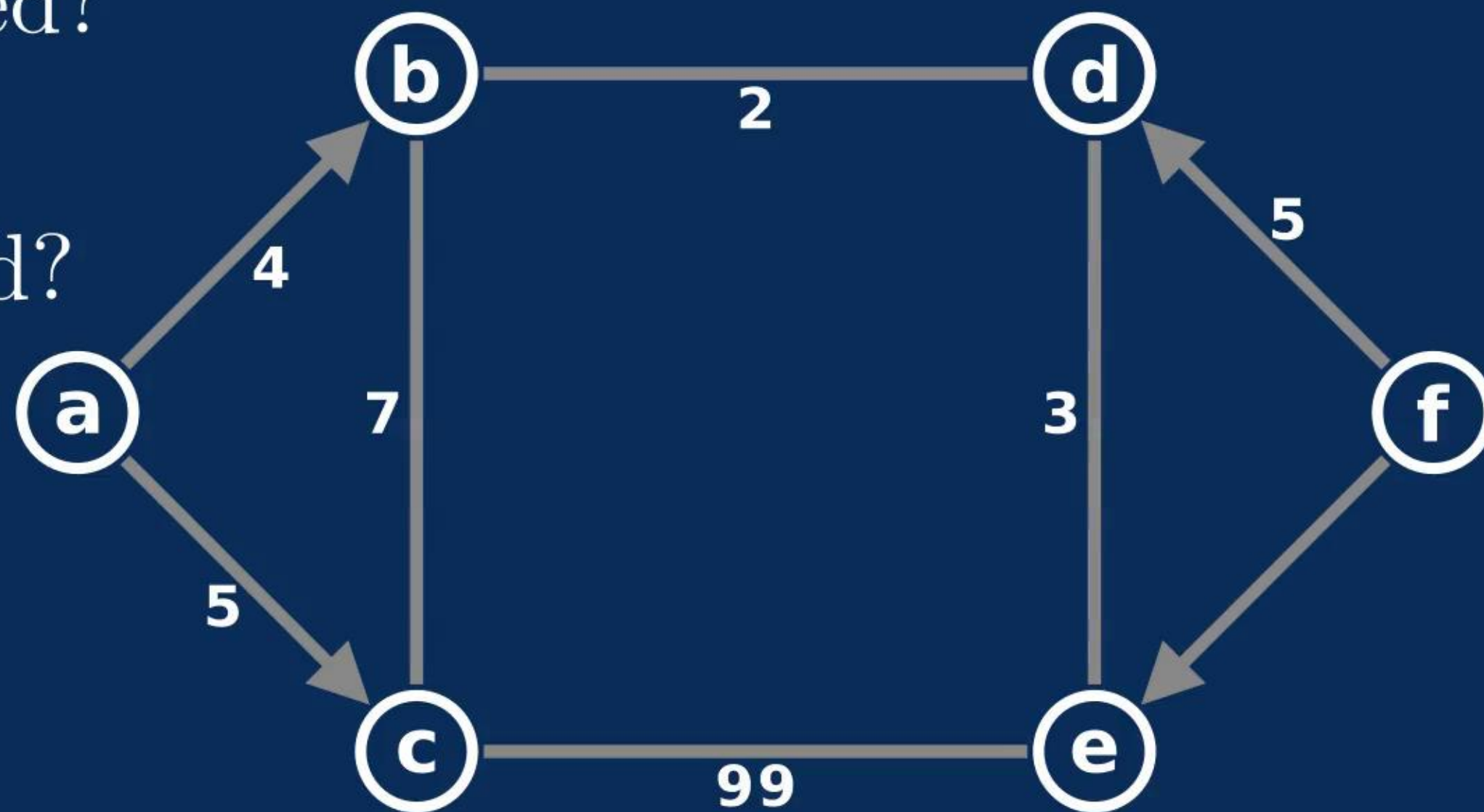


Which edges are weighted?

- All but f-e.

Which edges are directed?

- a-b, a-c, f-d, f-e



For these: what are the vertices and what are the edges?

- Flight routes between cities
- Legal Game moves
- Friendship connections
- Subway station map
- Prerequisite maps for classes

Self Loops are possible

Multiple Edges are possible

We don't usually like those.

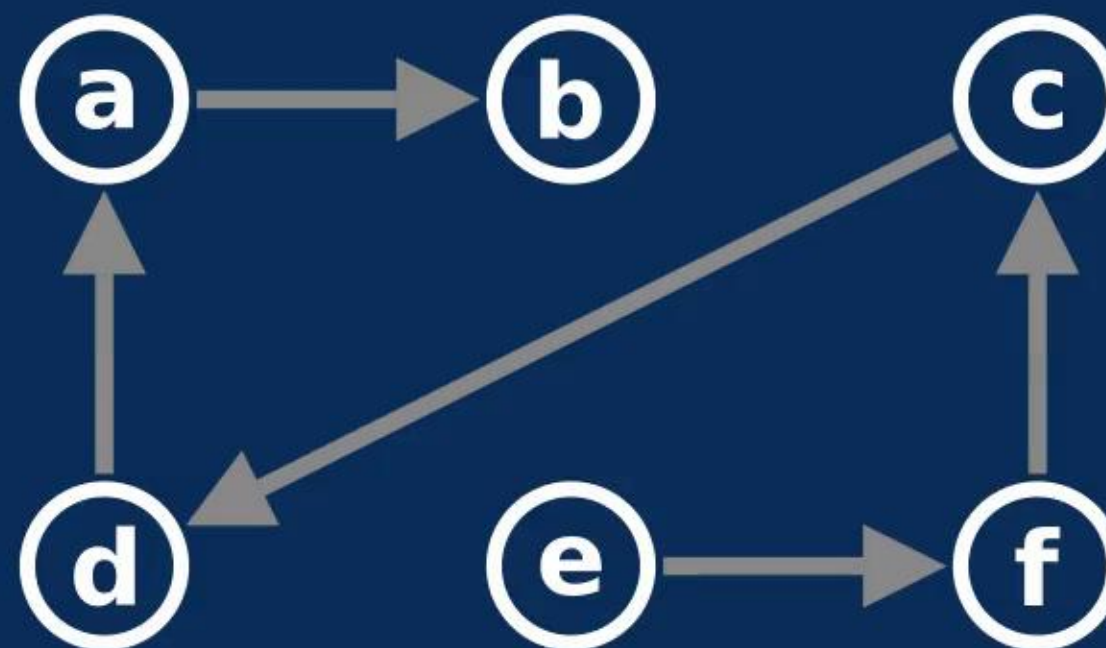


Simple Graph:

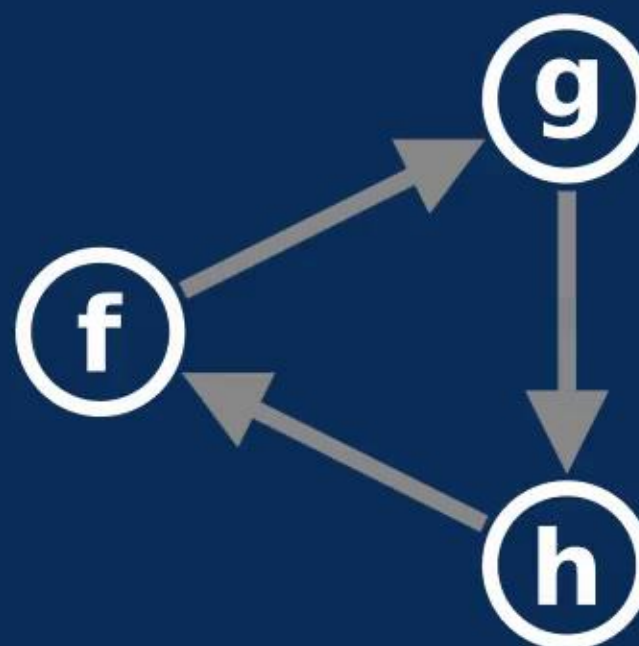
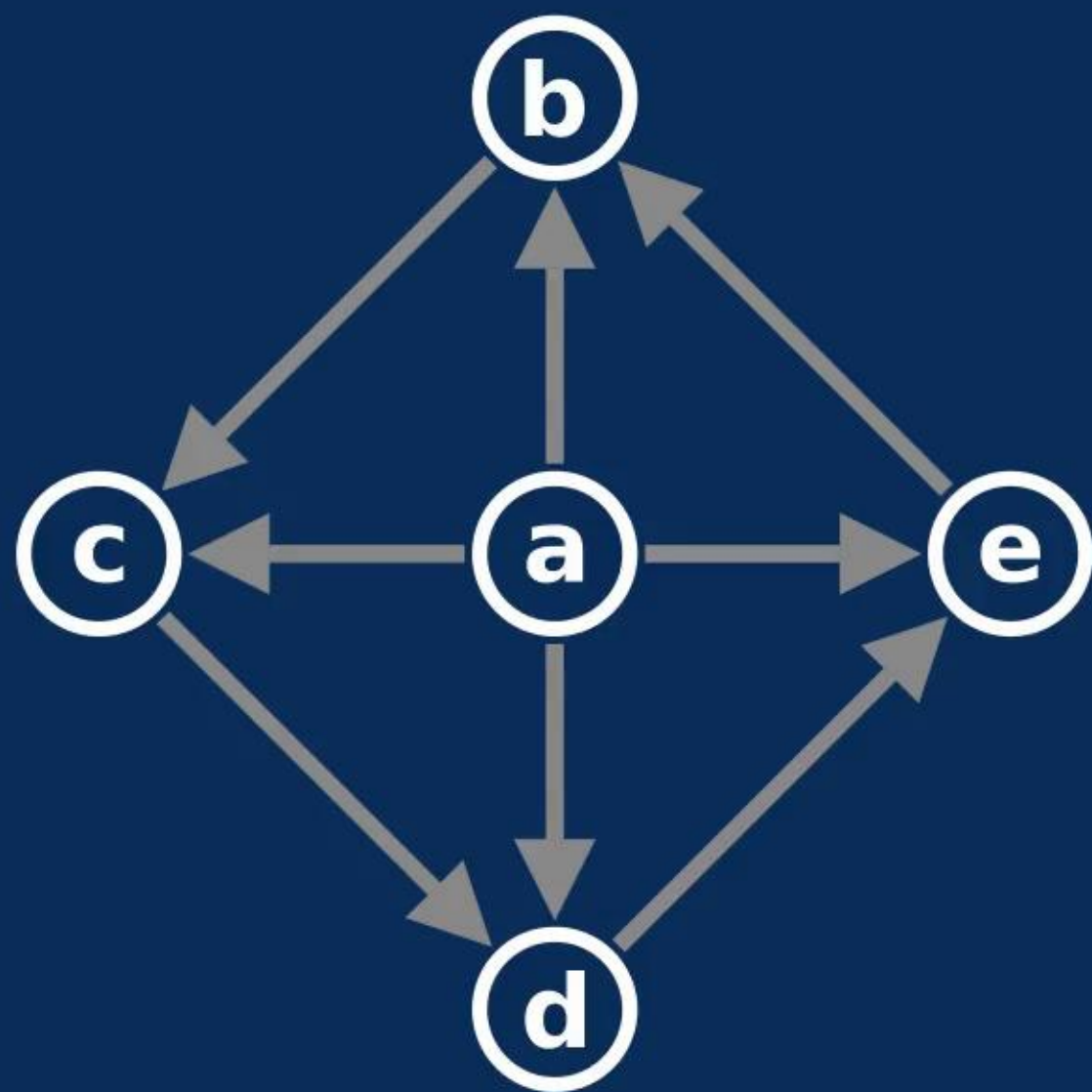
A subgraph G' of G :

- $G' = (V', E')$
- $V' \subset V$
- $E' \subset E$

if $(v_1, v_2) \in E$, then $v_1 \in E$ and $v_2 \in V'$



A graph can be **disconnected**!



Some Mathy Things

Implementation

Query

- `getEdges(Vertex v)`
- `areAdjacent(Vertex v1, Vertex v2)`
- `origin(Edge e)`
- `destination(Edge e)`

Three common ways to implement graphs

- Edge List (vector of edges)
- Adjacency List (vector of lists)
- Adjacency Matrix (2d vector)

How do you pick?

What needs to go in an edge?

What needs to go in an edge?

- Source and Destination Vertices

What needs to go in an edge?

- Source and Destination Vertices
- Possibly a weight

What needs to go in an edge?

- Source and Destination Vertices
- Possibly a weight
- Possibly a key

What needs to go in an edge?

- Source and Destination Vertices
- Possibly a weight
- Possibly a key

What about direction?

