



University  
of Manitoba

# Graphs – Introduction (theory)

MATH 2740 – Mathematics of Data Science – Lecture 14

Julien Arino

[julien.arino@umanitoba.ca](mailto:julien.arino@umanitoba.ca)

Department of Mathematics @ University of Manitoba

Fall 202X

The University of Manitoba campuses are located on original lands of Anishinaabeg, Ininew, Anisininew, Dakota and Dene peoples, and on the National Homeland of the Red River Métis. We respect the Treaties that were made on these territories, we acknowledge the harms and mistakes of the past, and we dedicate ourselves to move forward in partnership with Indigenous communities in a spirit of Reconciliation and collaboration.

# Outline

Why use graphs/networks?

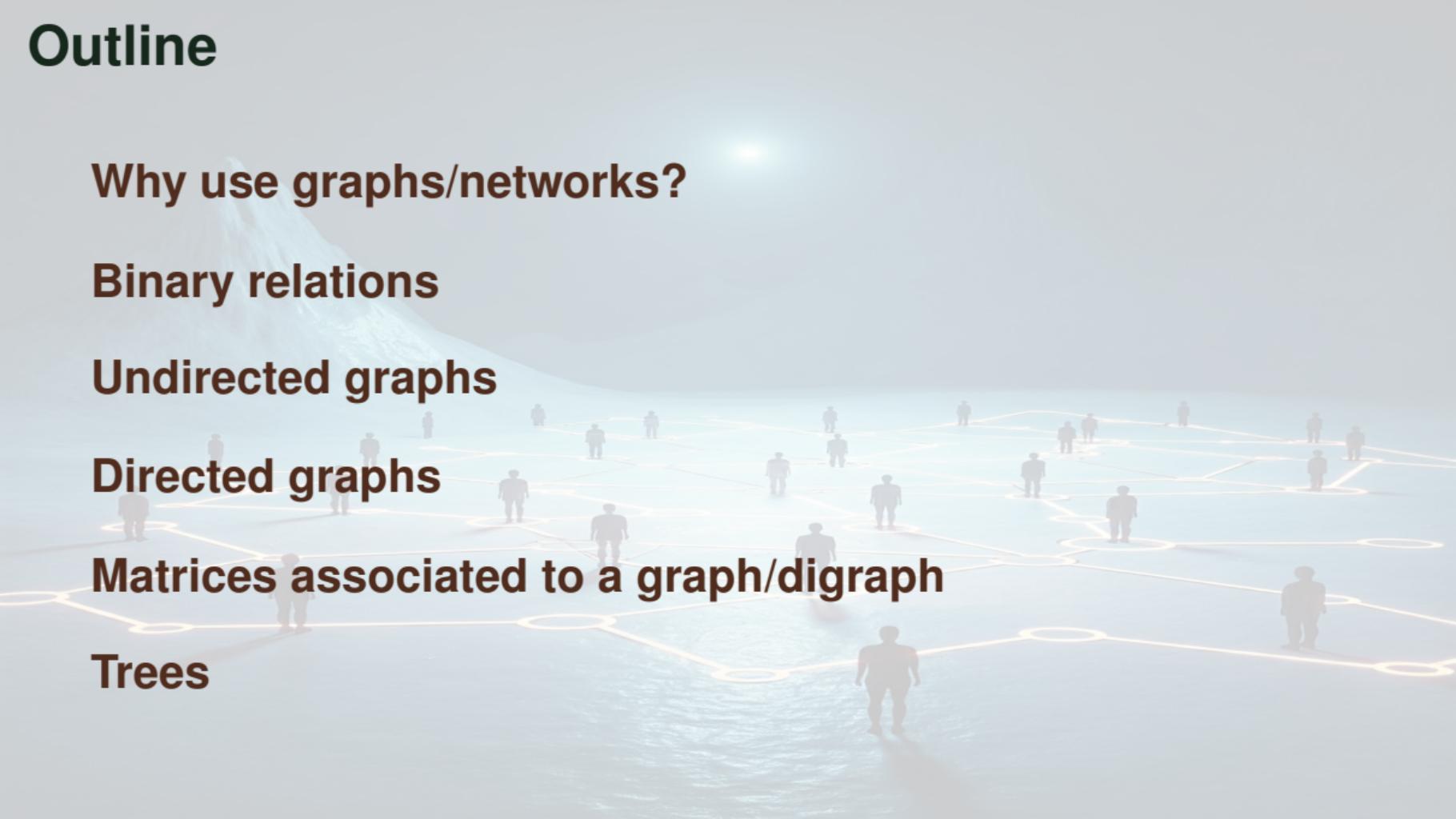
Binary relations

Undirected graphs

Directed graphs

Matrices associated to a graph/digraph

Trees



# Why use graphs/networks?

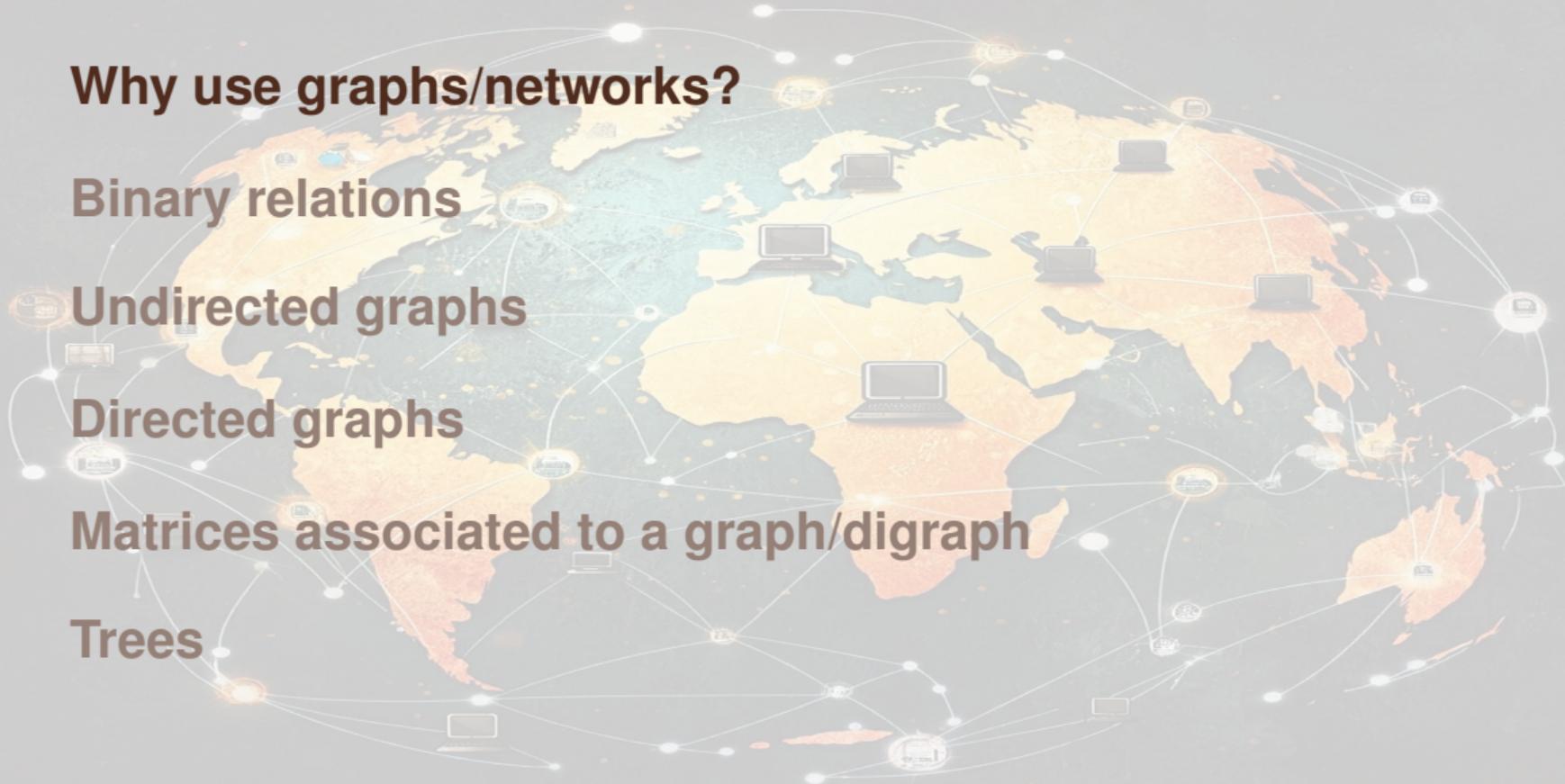
Binary relations

Undirected graphs

Directed graphs

Matrices associated to a graph/digraph

Trees



# Graphs versus networks

Mostly a terminology difference:

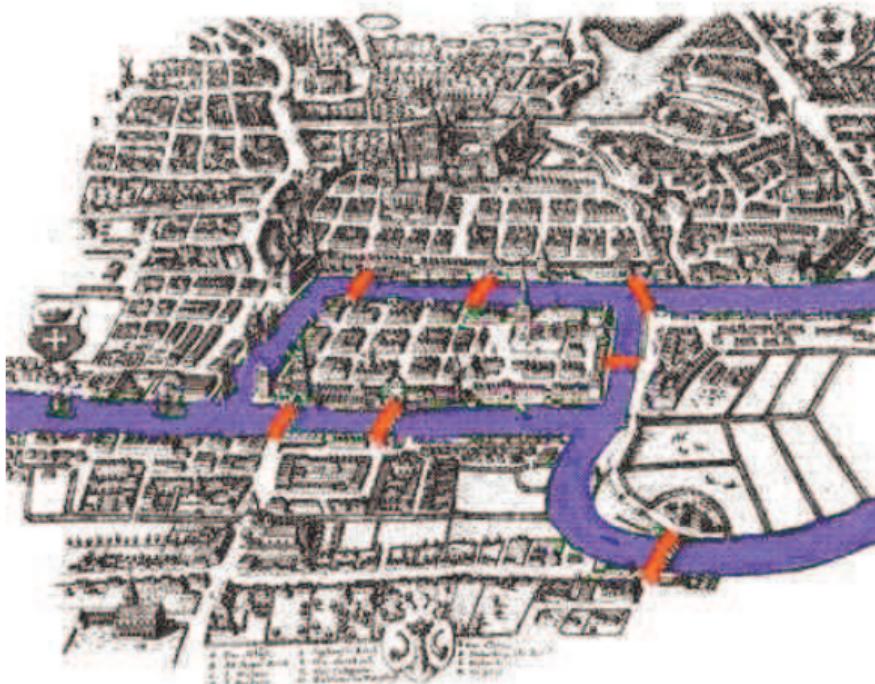
- ▶ graphs in the mathematical world
- ▶ networks elsewhere

I will mostly say *graphs* (this is a math course) but might oscillate

Beware: language is not consistent, so make sure you read the definitions at the start of whatever source you are using

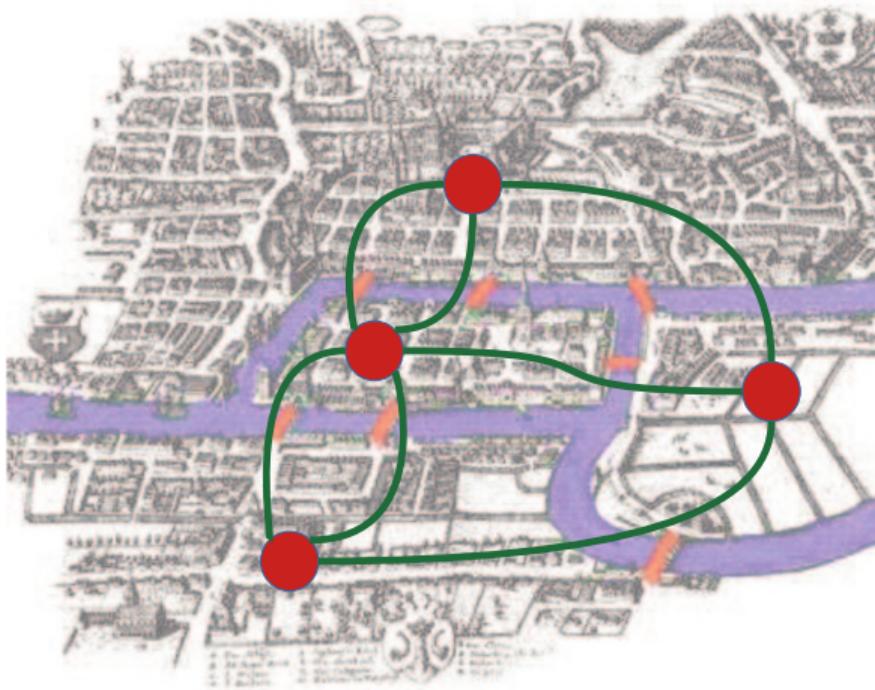
# The genesis of graphs – Euler's bridges of Königsberg

Cross the 7 bridges in a single walk without recrossing any of them?



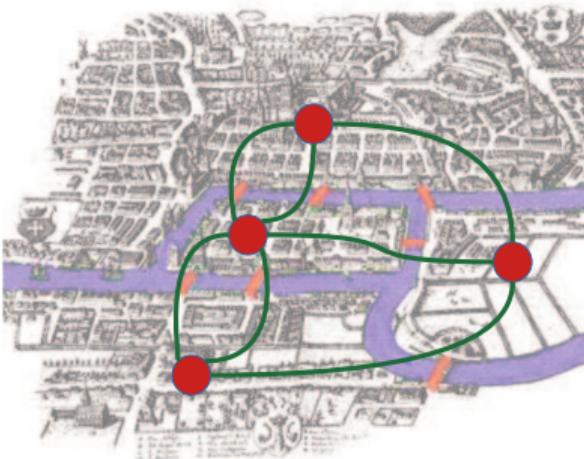
# The genesis of graphs – Euler's bridges of Königsberg

Cross the 7 bridges in a single walk without recrossing any of them?



# The genesis of graphs – Euler's bridges of Königsberg

Cross the 7 bridges in a single walk without recrossing any of them?



## Mathematical problem

Is it possible to find a *trail* containing all *edges* of the graph?

## Finding a cycle with all vertices

Salesperson must visit some cities (vertices) for their job. Can they plan a round trip using trains enabling them to visit each specified city exactly once?



- ▶ 2 vertices are connected iff a line connects the cities and does not pass through any other city

### Mathematical problem

Is it possible to find a cycle containing all graph vertices?

# How far is it to “train” through $n$ cities?

What is the minimal length of train travel needed to visit  $n$  cities (vertices)?



- ▶ all cities are connected; each edge has a value assigned to it (the distance)

## Mathematical problem

What is the minimal spanning tree associated to the graph?

## Graphs/networks encode relations

Graphs are used in a variety of contexts because they encode *relations* between objects

Many objects in the world have relations... so graphs are quite easy to find

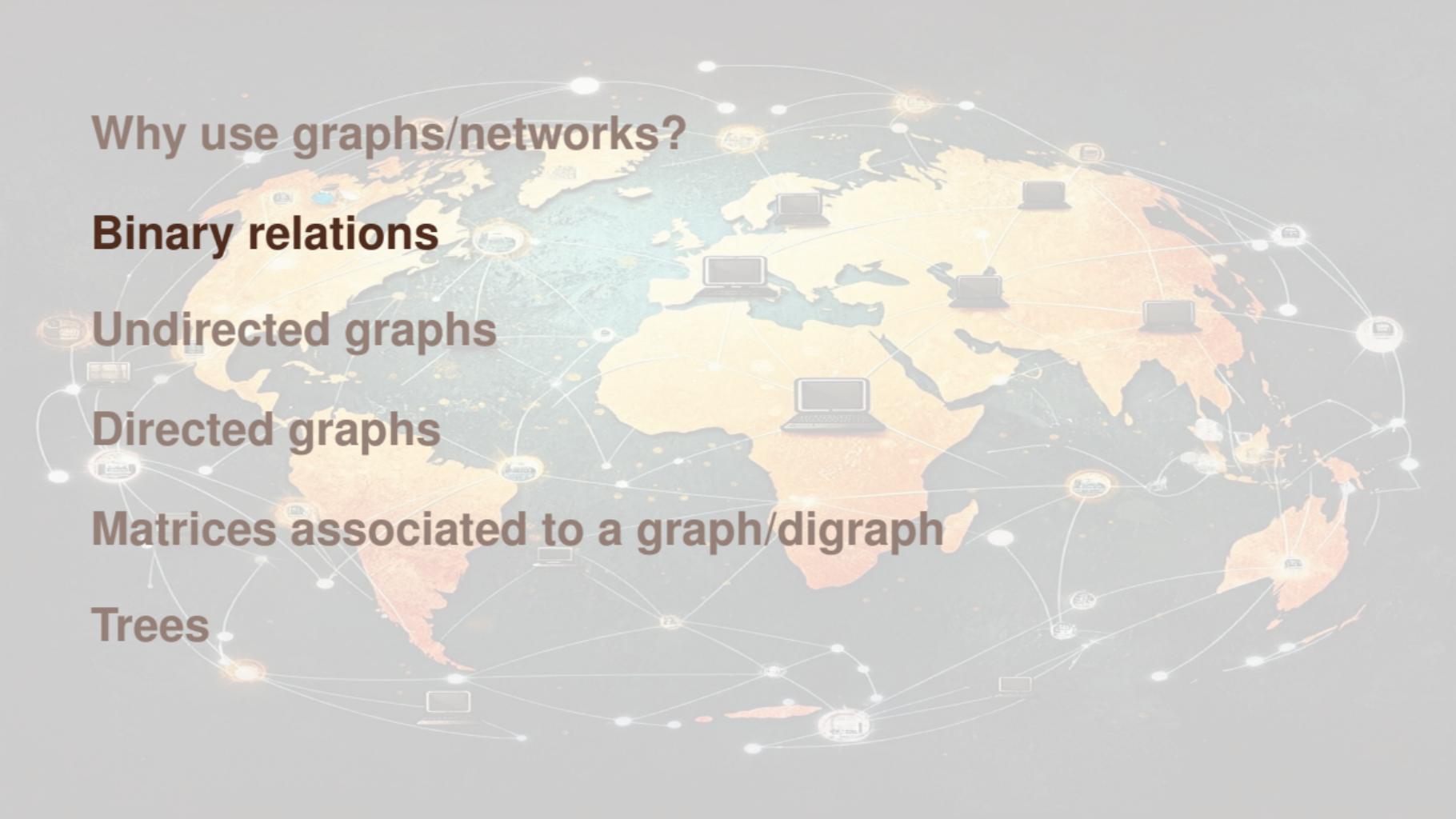
We will see many examples later, for now we cover the mathematical background

## Graphs vs digraphs vs multigraphs vs multidigraphs vs ...

Name-wise and notation-wise, this domain is a bit of a mess

- ▶ The vertex set  $V$  is essentially the only constant
- ▶ *Undirected graph*  $G = (V, E)$ , where  $E$  are the *edges*
- ▶ *Undirected multigraph*  $G_M = (V, E)$
- ▶ *Directed graph* (or *digraph*)  $G = (V, A)$ , where  $A$  are the *arcs*
- ▶ *Directed multigraph* (or *multidigraph*)  $G_M = (V, A)$
- ▶ Any of the above is called a *graph* and is denoted  $G = (V, X)$ , when we seek generality

And just to confuse the whole thing more: we often say *graph* for *unoriented graph*



Why use graphs/networks?

Binary relations

Undirected graphs

Directed graphs

Matrices associated to a graph/digraph

Trees

# Binary relation

## Definition 1 (Binary relation)

- ▶ A **binary relation** is an arbitrary association of elements of one set with elements of another (maybe the same) set
- ▶ A binary relation over the sets  $X$  and  $Y$  is defined as a subset of the Cartesian product  $X \times Y = \{(x, y) | x \in X, y \in Y\}$
- ▶  $(x, y) \in R$  is read “ $x$  is  $R$ -related to  $y$ ” and is denoted  $xRy$
- ▶ If  $(x, y) \notin R$ , we write “not  $xRy$ ” or  $x\not R y$

## Definition 2 (Properties of binary relations)

A binary relation  $R$  over a set  $X$  is

- ▶ **Reflexive** if  $\forall x \in X, xRx$
- ▶ **Irreflexive** if there does not exist  $x \in X$  such that  $xRx$
- ▶ **Symmetric** if  $xRy \Rightarrow yRx$
- ▶ **Asymmetric** if  $xRy \Rightarrow y \not Rx$
- ▶ **Antisymmetric** if  $xRy$  and  $yRx \Rightarrow x = y$
- ▶ **Transitive** if  $xRy$  and  $yRz \Rightarrow xRz$
- ▶ **Total** (or **complete**) if  $\forall x, y \in X, xRy$  or  $yRx$

### Definition 3 (Equivalence relation)

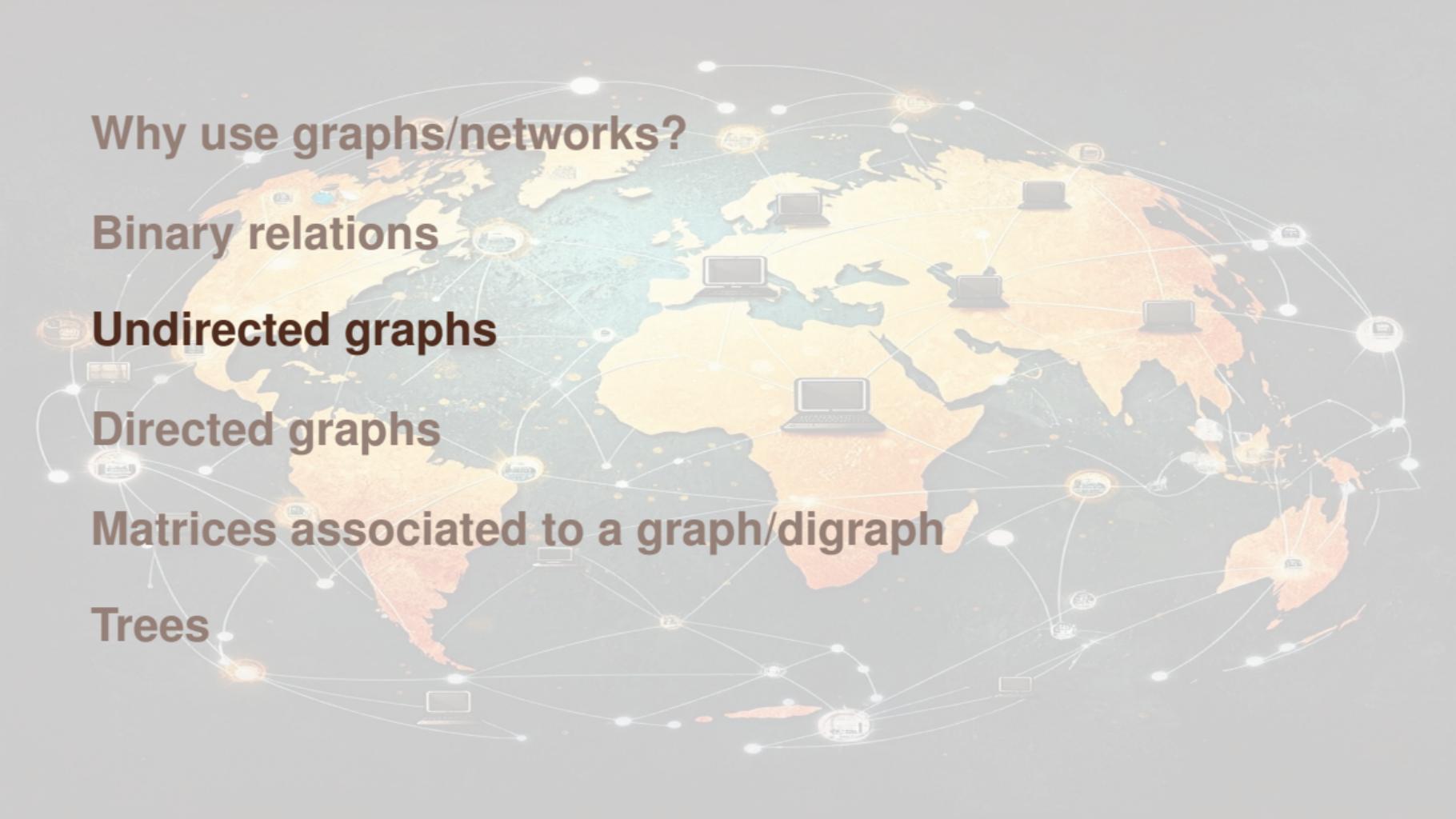
A relation that is reflexive ( $\forall x \in X, xRx$ ), symmetric ( $xRy \Rightarrow yRx$ ) and transitive ( $xRy$  and  $yRz \Rightarrow xRz$ ) is an **equivalence relation**

### Definition 4 (Partial order)

A relation that is reflexive ( $\forall x \in X, xRx$ ), antisymmetric ( $xRy$  and  $yRx \Rightarrow x = y$ ) and transitive ( $xRy$  and  $yRz \Rightarrow xRz$ ) is a **partial order**

### Definition 5 (Total order)

A partial order that is total ( $\forall x, y \in X, xRy$  or  $yRx$ ) is a **total order**



**Why use graphs/networks?**

**Binary relations**

**Undirected graphs**

**Directed graphs**

**Matrices associated to a graph/digraph**

**Trees**

## The `igraph` library

Throughout these slides, we use the package `igraph`

I illustrate the functions that can be used to study some of the mathematical notions I introduce

I use mostly examples from the `igraph` documentation

# **Undirected graphs**

**Undirected graph**

**Degree of a vertex**

**Isomorphic graphs**

**Subgraphs, unions of graphs**

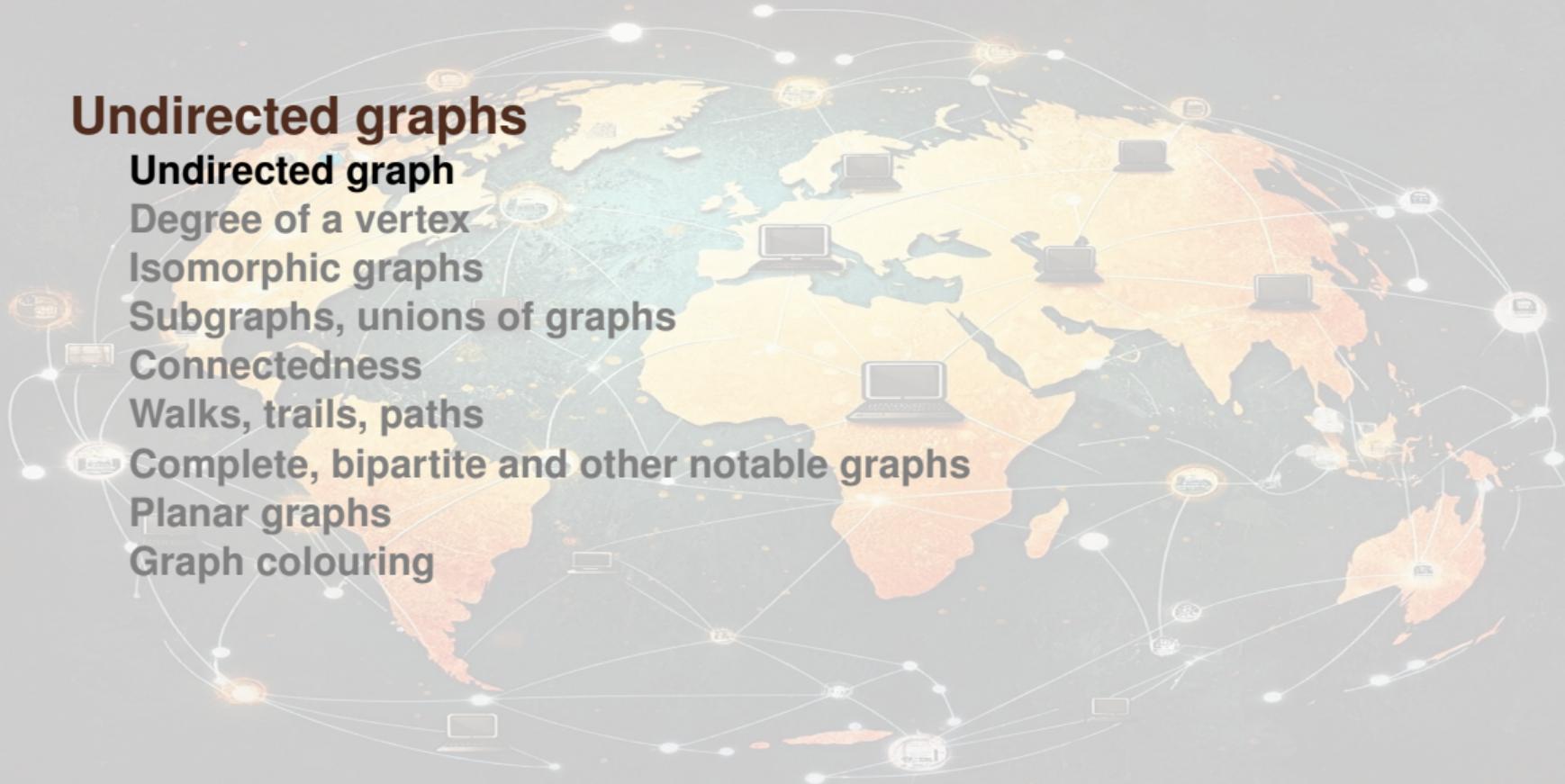
**Connectedness**

**Walks, trails, paths**

**Complete, bipartite and other notable graphs**

**Planar graphs**

**Graph colouring**



# Graph

Intuitively: a graph is a set of points, and a set of relations between the points

The points are called the *vertices* of the graph and the relations are the *edges* of the graph

We can also think of the relations as being one directional, in which case the relations are the *arcs* of the digraph (a contraction of “directed graph”)

# Graph, vertex and edge

## Definition 6 (Graph)

An **undirected graph** is a pair  $G = (V, E)$  of sets such that

- ▶  $V$  is a set of points:  $V = \{v_1, \dots, v_p\}$
- ▶  $E$  is a set of 2-element subsets of  $V$ :  $E = \{\{v_i, v_j\}, \{v_i, v_k\}, \dots, \{v_n, v_p\}\}$  or  
 $E = \{v_i v_j, v_i v_k, \dots, v_n v_p\}$

## Definition 7 (Vertex)

The elements of  $V$  are the **vertices** (or nodes, or points) of the graph  $G$ .  $V$  (or  $V(G)$ ) is the vertex set of the graph  $G$

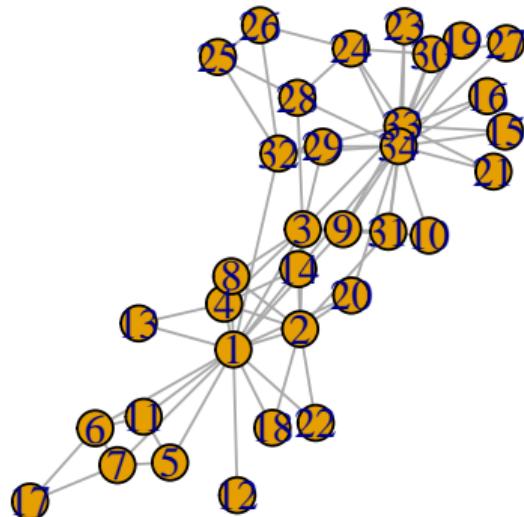
## Definition 8 (Edge)

The elements of  $E$  are the **edges** (or lines) of the graph  $G$ .  $E$  (or  $E(G)$ ) is the edge set of the graph  $G$

## Setting up graphs in igraph

```
G <- make_empty_graph()
G <- make_graph(edges = c(1, 2, 1, 5), n = 10,
                 directed = FALSE)
G <- make_graph("Zachary")
plot(G, main = "Data on a Karate club")
```

## Data on a Karate club



## Order and Size

### Definition 9 (Order of a graph)

The number of vertices in  $G$  is the **order** of  $G$ . Using the notation  $|V(G)|$  for the *cardinality* of  $V(G)$ ,

$$|V(G)| = \text{order of } G$$

### Definition 10 (Size of a graph)

The number of edges in  $G$  is the **size** of  $G$ ,

$$|E(G)| = \text{size of } G$$

- ▶ A graph having order  $p$  and size  $q$  is called a  $(p, q)$ -graph
- ▶ A graph is finite if  $|V(G)| < \infty$

## Some simple measures

**v(G)**

```
## + 34/34 vertices, from dfcdd51:  
## [1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23  
## [26] 26 27 28 29 30 31 32 33 34
```

**head(E(G))**

```
## + 6/78 edges from dfcdd51:  
## [1] 1--2 1--3 1--4 1--5 1--6 1--7
```

**gorder(G)**

```
## [1] 34
```

**gsizes(G)**

```
## [1] 78
```

## Incident – Adjacent

### Definition 11 (Incident)

- ▶ A vertex  $v$  is **incident** with an edge  $e$  if  $v \in e$ ; then  $e$  is an edge at  $v$
- ▶ If  $e = uv \in E(G)$ , then  $u$  and  $v$  are each incident with  $e$
- ▶ The two vertices incident with an edge are its ends
- ▶ An edge  $e = uv$  is incident with both vertices  $u$  and  $v$

### Definition 12 (Adjacent)

- ▶ Two vertices  $u$  and  $v$  are **adjacent** in a graph  $G$  if  $uv \in E(G)$
- ▶ If  $uv$  and  $uw$  are distinct edges (i.e.  $v \neq w$ ) of a graph  $G$ , then  $uv$  and  $uw$  are adjacent edges

## Incident vertices

```
incident(G, 1)

## + 16/78 edges from dfcdd51:
## [1] 1-- 2 1-- 3 1-- 4 1-- 5 1-- 6 1-- 7 1-- 8 1-- 9 1--11 1--12 1--13 1-
## [13] 1--18 1--20 1--22 1--32

incident_edges(G, c(1, 2))

## [[1]]
## + 16/78 edges from dfcdd51:
## [1] 1-- 2 1-- 3 1-- 4 1-- 5 1-- 6 1-- 7 1-- 8 1-- 9 1--11 1--12 1--13 1-
## [13] 1--18 1--20 1--22 1--32
##
## [[2]]
## + 9/78 edges from dfcdd51:
## [1] 1-- 2 2-- 3 2-- 4 2-- 8 2--14 2--18 2--20 2--22 2--31
```

## Adjacent vertices

```
adjacent_vertices(G, v = 1)

## [[1]]
## + 16/34 vertices, from dfcdd51:
## [1]  2  3  4  5  6  7  8  9 11 12 13 14 18 20 22 32

adjacent_vertices(G, v = c(1, 2))

## [[1]]
## + 16/34 vertices, from dfcdd51:
## [1]  2  3  4  5  6  7  8  9 11 12 13 14 18 20 22 32
##
## [[2]]
## + 9/34 vertices, from dfcdd51:
## [1]  1  3  4  8 14 18 20 22 31
```

### Definition 13 (Multiple edge)

**Multiple edges** are two or more edges connecting the same two vertices within a multigraph

### Definition 14 (Loop)

A **loop** is an edge with both the same ends; e.g.  $\{u, u\}$  is a loop

### Definition 15 (Simple graph)

A **simple graph** is a graph which contains no loops or multiple edges

### Definition 16 (Multigraph)

A **multigraph** is a graph which can contain multiple edges or loops

## Testing for these properties

```
any_multiple(G)  
## [1] FALSE  
  
any_loop(G)  
## [1] FALSE  
  
is_simple(G)  
## [1] TRUE
```

## Graph and binary relations

A simple graph  $G$  can be defined in term of a vertex set  $V$  and a binary relation over  $V$  that is

- ▶ irreflexive ( $\forall u \in V, u \not R u$ )
- ▶ symmetric ( $\forall u, v \in V, u R v \implies v R u$ )

The set of edges  $E(G)$  is the set of symmetric pairs in  $R$

If  $R$  is not irreflexive, the graph is not simple

# Undirected graphs

Undirected graph

**Degree of a vertex**

Isomorphic graphs

Subgraphs, unions of graphs

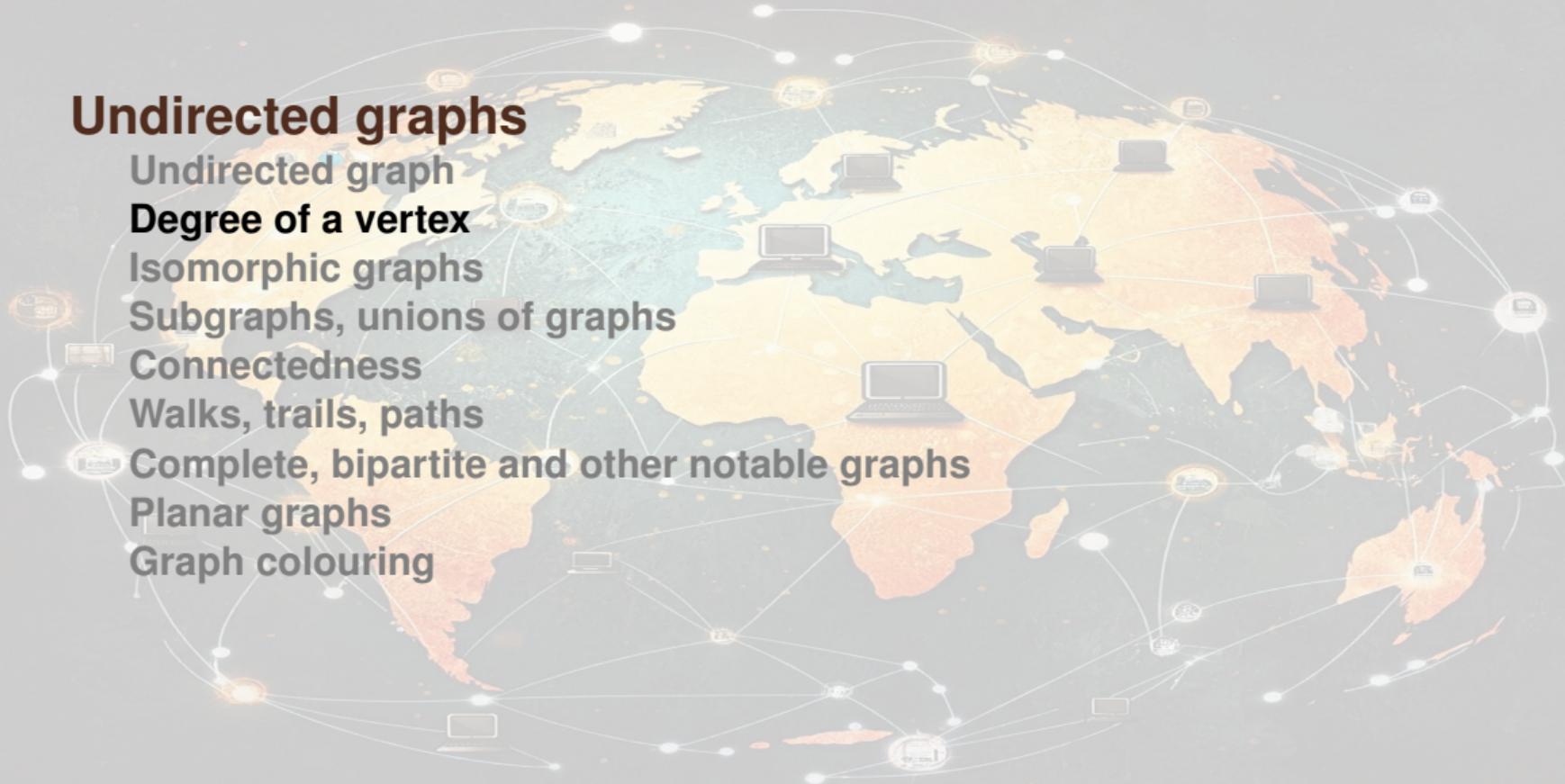
Connectedness

Walks, trails, paths

Complete, bipartite and other notable graphs

Planar graphs

Graph colouring



## Definition 17 (Degree of a vertex)

Let  $v$  be a vertex of  $G = (V, E)$ .

- ▶ The number of edges of  $G$  incident with  $v$  is the **degree** of  $v$  in  $G$
- ▶ The number of edges of  $G$  at  $v$  is the **degree** of  $v$  in  $G$
- ▶ The degree of  $v$  in  $G$  is noted  $d_G(v)$  or  $\deg_G(v)$

## Theorem 18

Let  $G$  be a  $(p, q)$ -graph with vertices  $v_1, \dots, v_p$ , then

$$\sum_{i=1}^p d_G(v_i) = 2q$$

# Degree

```
degree(G)

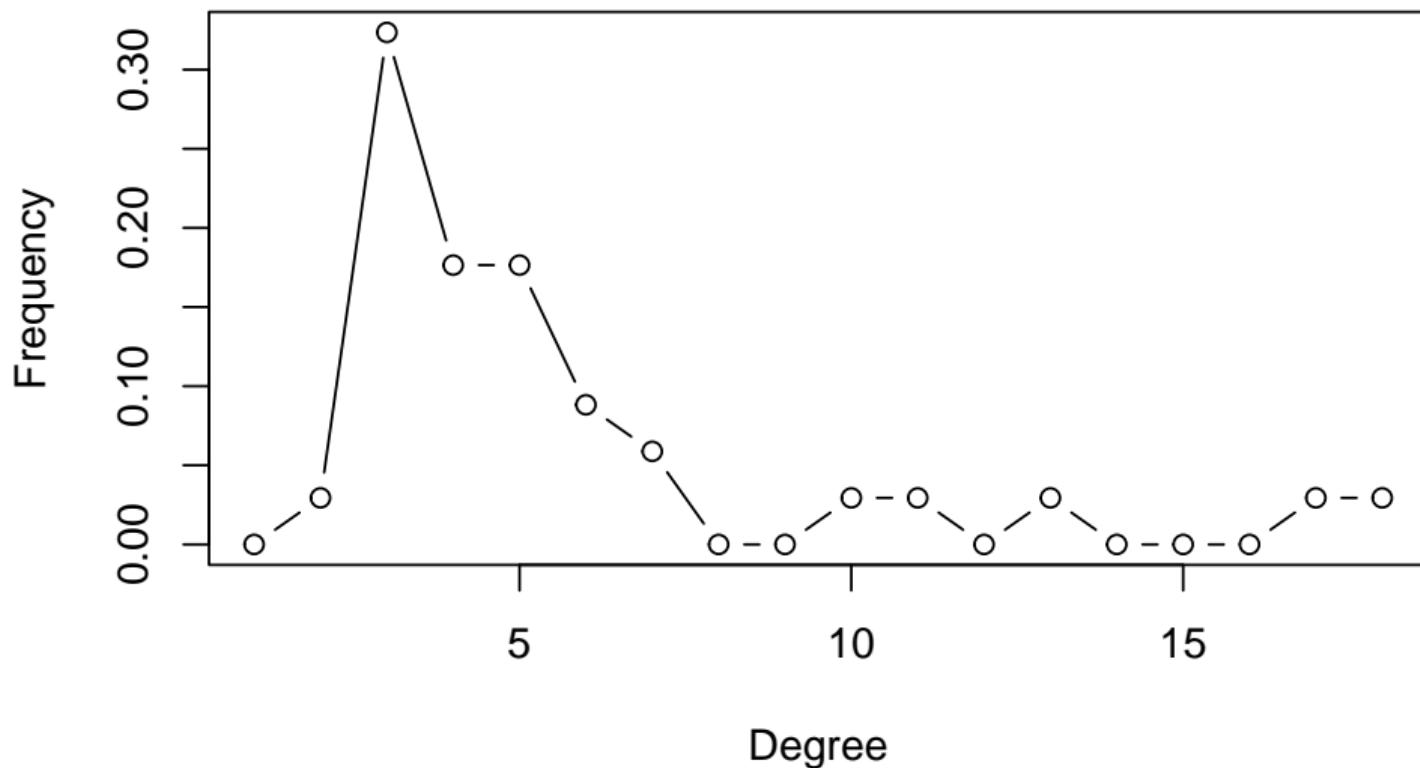
## [1] 16 9 10 6 3 4 4 4 5 2 3 1 2 5 2 2 2 2 2 3 2 2

degree_distribution(G)

## [1] 0.00000000 0.02941176 0.32352941 0.17647059 0.17647059 0.08823529
## [7] 0.05882353 0.00000000 0.00000000 0.02941176 0.02941176 0.00000000
## [13] 0.02941176 0.00000000 0.00000000 0.00000000 0.02941176 0.02941176

plot(degree_distribution(G),
      type = "b",
      xlab = "Degree", ylab = "Frequency",
      main = "Degree distribution of the Karate graph")
```

## Degree distribution of the Karate graph



Definition 19 (Odd vertex)

A vertex is an **odd vertex** if its degree is odd

Definition 20 (Even vertex)

A vertex is called **even vertex** if its degree is even

Theorem 21

*Every graph contains an even number of odd vertices*

## Illustration of recent results

Theorem 18 states that a  $(p, q)$ -graph has  $\sum_{i=1}^p d_G(v_i) = 2q$

```
sum(degree(G)) == 2*length(E(G))  
## [1] TRUE
```

Theorem 21 states that every graph contains an even number of odd vertices

```
sum(pracma::mod(degree(G), 2))  
## [1] 12
```

(`mode(x, 2)` returns 1 if  $x$  is odd so `sum` counts how many are odd)

# Regular graph

## Definition 22 (Regular graph)

If all the vertices of  $G$  have the same degree  $k$ , then the graph  $G$  is  $k$ -regular

```
length(unique(degree(G))) == 1  
## [1] FALSE
```

# **Undirected graphs**

Undirected graph

Degree of a vertex

**Isomorphic graphs**

Subgraphs, unions of graphs

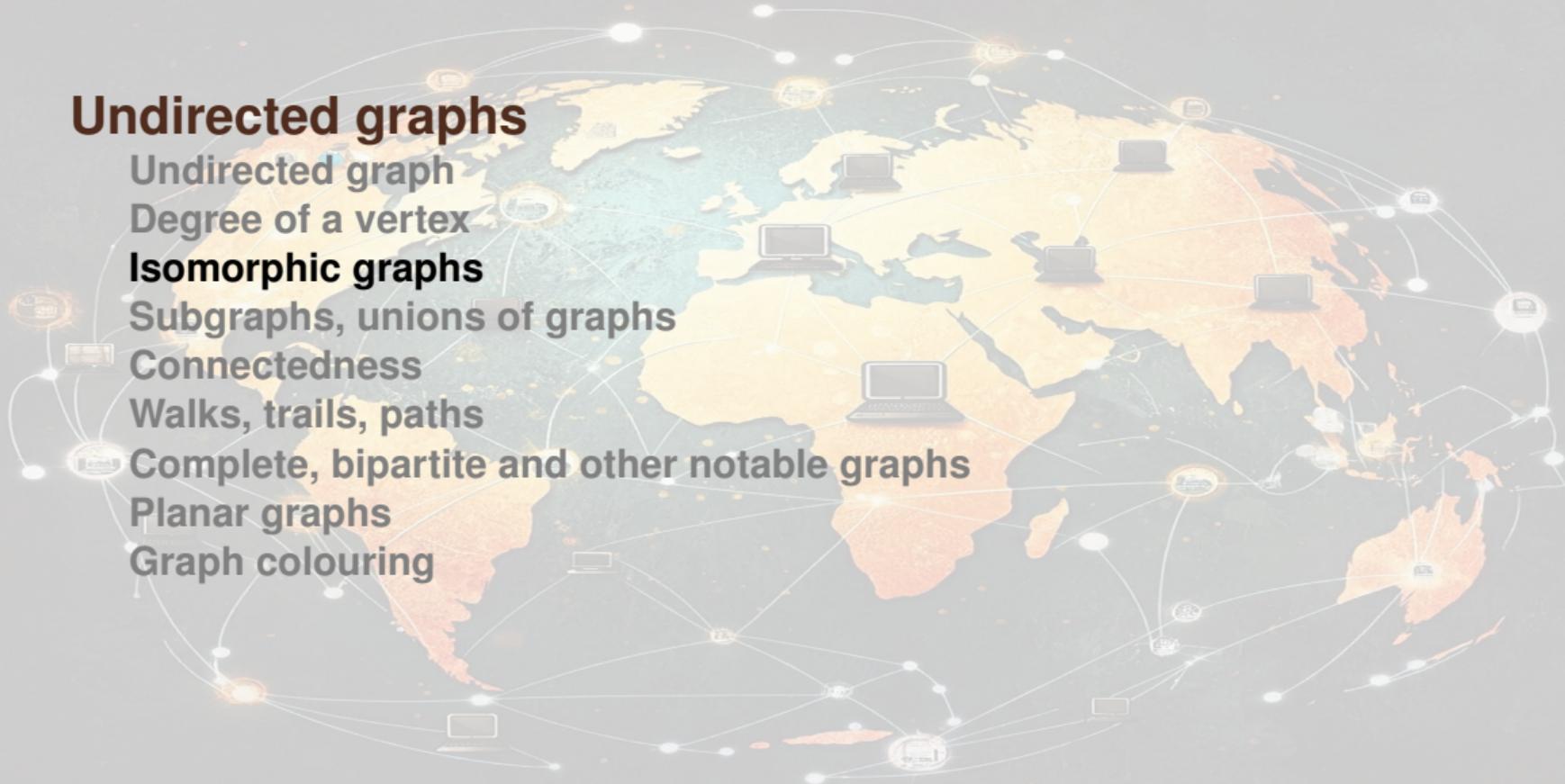
Connectedness

Walks, trails, paths

Complete, bipartite and other notable graphs

Planar graphs

Graph colouring



## Isomorphic graphs

### Definition 23 (Isomorphic graphs)

Let  $G_1 = (V(G_1), E(G_1))$  and  $G_2 = (V(G_2), E(G_2))$  be two graphs.  $G_1$  and  $G_2$  are **isomorphic** if there exists an isomorphism  $\phi$  from  $G_1$  to  $G_2$ , that is defined as an injective mapping  $\phi : V(G_1) \rightarrow V(G_2)$  such that two vertices  $u_1$  and  $v_1$  are adjacent in  $G_1 \iff$  the vertices  $\phi(u_1)$  and  $\phi(v_1)$  are adjacent in  $G_2$

If  $\phi$  is an isomorphism from  $G_1$  to  $G_2$ , then the inverse mapping  $\phi^{-1}$  from  $V(G_2)$  to  $V(G_1)$  also satisfies the definition of an isomorphism. As a consequence, if  $G_1$  and  $G_2$  are isomorphic graphs, then

- ▶  $G_1$  is isomorphic to  $G_2$
- ▶  $G_2$  is isomorphic to  $G_1$

### Theorem 24

*The relation “is isomorphic to” is an equivalence relation on the set of all graphs*

### Theorem 25

*If  $G_1$  and  $G_2$  are isomorphic graphs, then the degrees of vertices of  $G_1$  are exactly the degrees of vertices of  $G_2$*

## Testing isomorphicity

Create two isomorphic graphs by permuting the vertices of the first, then test if they are isomorphic

```
g1 <- sample_pa(30, m = 2, directed = FALSE)
g2 <- permute(g1, sample(vcount(g1)))
# should be TRUE
isomorphic(g1, g2)
## [1] TRUE
```

# **Undirected graphs**

Undirected graph

Degree of a vertex

Isomorphic graphs

**Subgraphs, unions of graphs**

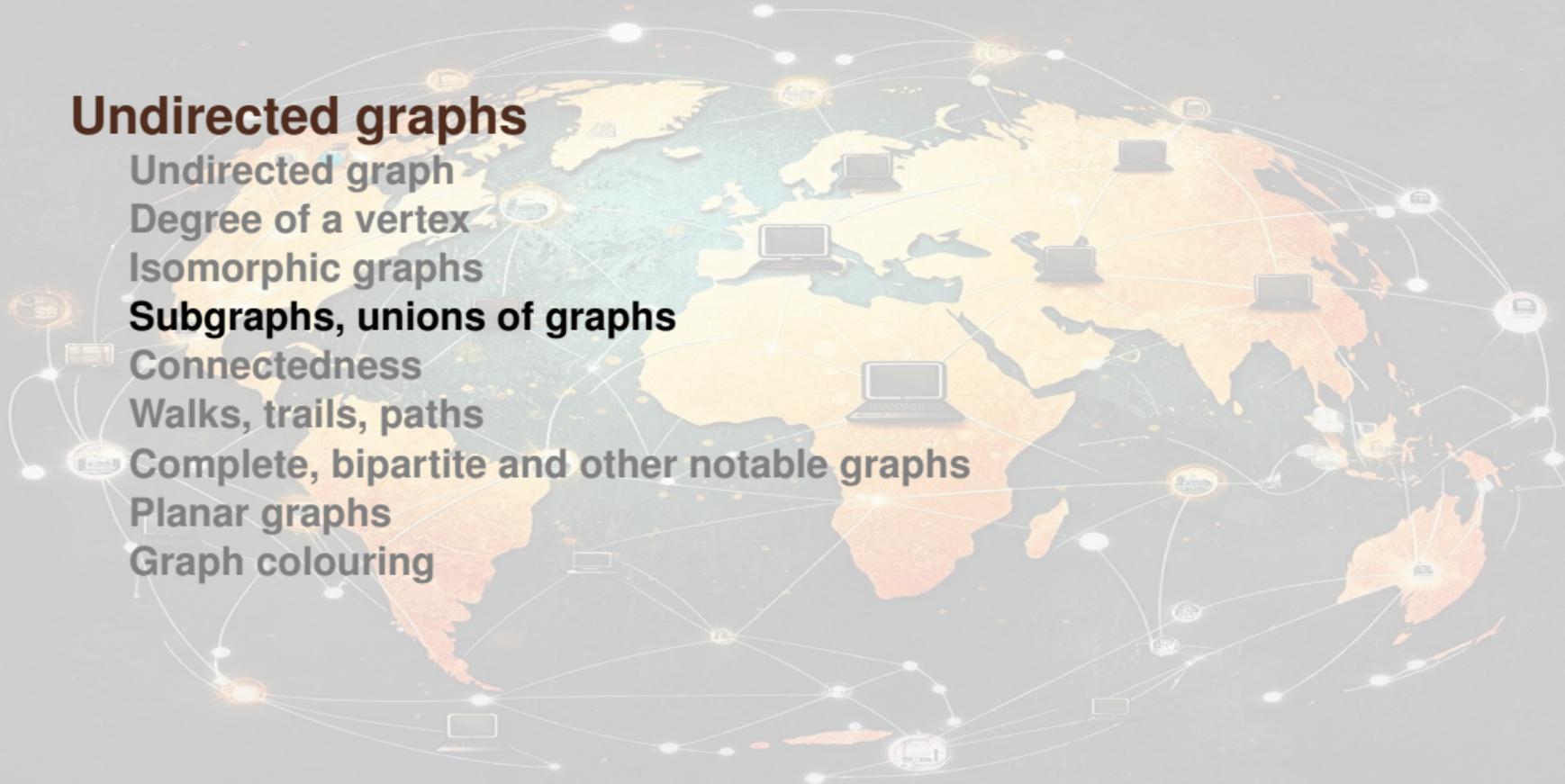
Connectedness

Walks, trails, paths

Complete, bipartite and other notable graphs

Planar graphs

Graph colouring



# Subgraph

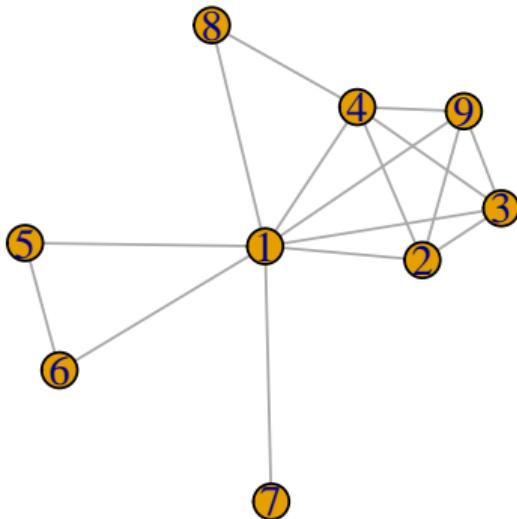
## Definition 26 (Subgraph)

Let  $G = (V, E)$  be a graph. A graph  $H = (V(H), E(H))$  is a **subgraph** of  $G$  if  $V(H) \subseteq V$  and  $E(H) \subseteq E$

## Extracting a subgraph

```
G_sub = induced_subgraph(G, c(1:5, 11:14))
plot(G_sub, main = "A subgraph of the Karate graph")
```

## A subgraph of the Karate graph



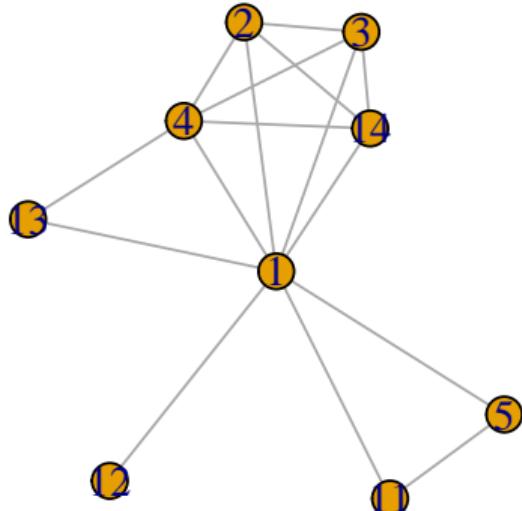
## Naming vertices

Note that vertices (and as a consequence, edges) will be relabelled, so `G_sub` has vertices labelled 1 to 9

You can name vertices if you want to avoid this

```
V(G)$name = 1:length(V(G))
G_sub = induced_subgraph(G, c(1:5, 11:14))
plot(G_sub, labels = V(G)$name,
      main = "Subgraph of the Karate graph preserving names")
```

## Subgraph of the Karate graph preserving names



Let  $G_1 = (V_1, E_1)$  and  $G_2 = (V_2, E_2)$  be two graphs

**Definition 27 (Union of  $G_1$  and  $G_2$ )**

$$G_1 \cup G_2 = (V_1 \cup V_2, E_1 \cup E_2)$$

**Definition 28 (Intersection of  $G_1$  and  $G_2$ )**

$$G_1 \cap G_2 = (V_1 \cap V_2, E_1 \cap E_2)$$

**Definition 29 (Disjoint graphs)**

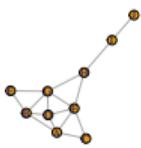
If  $G_1 \cap G_2 = (\emptyset, \emptyset) = \emptyset$  (empty graph) then  $G_1$  and  $G_2$  are **disjoint**

**Definition 30 (Complement of  $G_1$ )**

The **complement**  $\bar{G}_1$  of  $G_1$  is the graph on  $V_1$ , with the edge set  
 $E(\bar{G}_1) = [V_1]^2 \setminus E_1$  ( $e \in E(\bar{G}_1) \iff e \notin E_1$ )

# Union

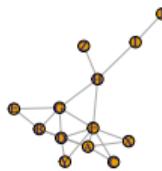
```
net1 <- graph_from_literal(  
  D - A:B:F:G, A - C - F - A, B - E - G - B, A - B, F - G,  
  H - F:G, H - I - J  
)  
net2 <- graph_from_literal(D - A:F:Y, B - A - X - F - H - Z, F - Y)  
print_all(net1 %u% net2)  
  
## IGRAPH c5771e3 UN-- 13 21 --  
## + attr: name (v/c)  
## + edges from c5771e3 (vertex names):  
## [1] I--J H--Z H--I G--H G--E F--X F--Y F--H F--C F--G B--E B--G A--X A--  
## [16] A--B D--Y D--G D--F D--B D--A
```



U

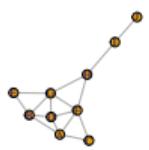
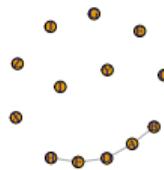


⇒



## Intersection of two graphs

```
G_inter = net1 %s% net2  
plot(G_inter)
```

 $\cap$  $\implies$ 

# **Undirected graphs**

Undirected graph

Degree of a vertex

Isomorphic graphs

Subgraphs, unions of graphs

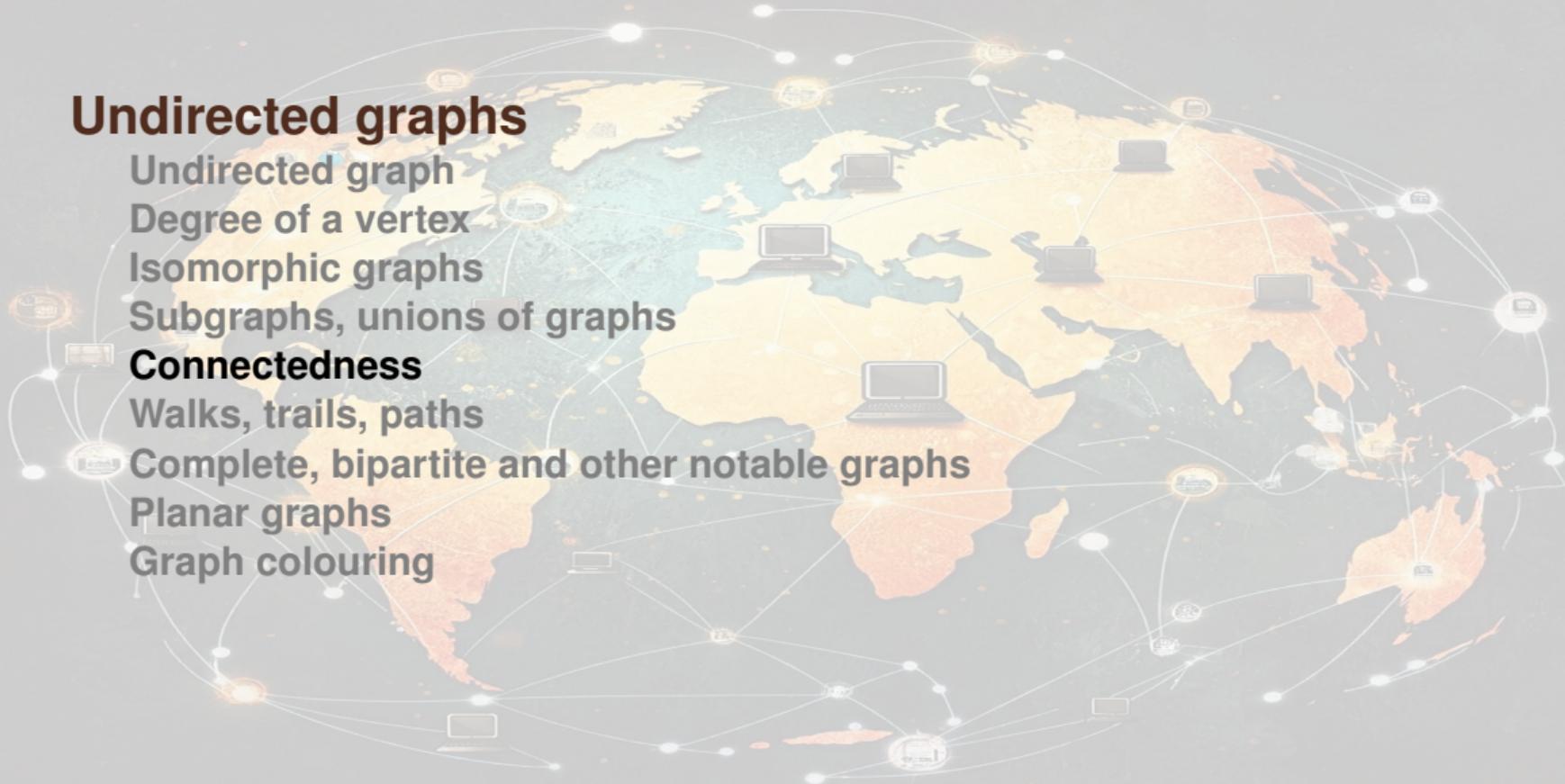
**Connectedness**

Walks, trails, paths

Complete, bipartite and other notable graphs

Planar graphs

Graph colouring



## Connected vertices and graph, components

### Definition 31 (Connected vertices)

Two vertices  $u$  and  $v$  in a graph  $G$  are **connected** if  $u = v$ , or if  $u \neq v$  and there exists a path in  $G$  that links  $u$  and  $v$

(For *path*, see Definition 44 later)

### Definition 32 (Connected graph)

A graph is **connected** if every two vertices of  $G$  are connected; otherwise,  $G$  is **disconnected**

## A necessary condition for connectedness

### Theorem 33

*A connected graph on  $p$  vertices has at least  $p - 1$  edges*

In other words, a connected graph  $G$  of order  $p$  has  $\text{size}(G) \geq p - 1$

## Connectedness is an equivalence relation

Denote  $x \equiv y$  the relation “ $x = y$ , or  $x \neq y$  and there exists a path in  $G$  connecting  $x$  and  $y$ ”.  $\equiv$  is an equivalence relation since

1.  $x \equiv y$  [reflexivity]
2.  $x \equiv y \implies y \equiv x$  [symmetry]
3.  $x \equiv y, y \equiv z \implies x \equiv z$  [transitivity]

### Definition 34 (Connected component of a graph)

The classes of the equivalence relation  $\equiv$  partition  $V$  into connected sub-graphs of  $G$  called **connected components** (or **components** for short) of  $G$

A connected subgraph  $H$  of a graph  $G$  is a component of  $G$  if  $H$  is not contained in any connected subgraph of  $G$  having more vertices or edges than  $H$

## Vertex deletion & cut vertices

### Definition 35 (Vertex deletion)

If  $v \in V(G)$  is a vertex of  $G$ , the graph  $G - v$  is the graph formed from  $G$  by removing  $v$  and all edges incident with  $v$

### Definition 36 (Cut-vertices)

Let  $G$  be a connected graph. Then  $v$  is a **cut-vertex** of  $G$  if  $G - v$  is disconnected

## Edge deletion & bridges

### Definition 37 (Edge deletion)

If  $e$  is an edge of  $G$ , the graph  $G - e$  is the graph formed from  $G$  by removing  $e$  from  $G$

### Definition 38 (Bridge)

An edge  $e$  in a connected graph  $G$  is a **bridge** if  $G - e$  is disconnected

### Theorem 39

*Let  $G$  be a connected graph. An edge  $e$  of  $G$  is a bridge of  $G \iff e$  does not lie on any cycle of  $G$*

(For *cycle*, see Definition 47 later)

# **Undirected graphs**

Undirected graph

Degree of a vertex

Isomorphic graphs

Subgraphs, unions of graphs

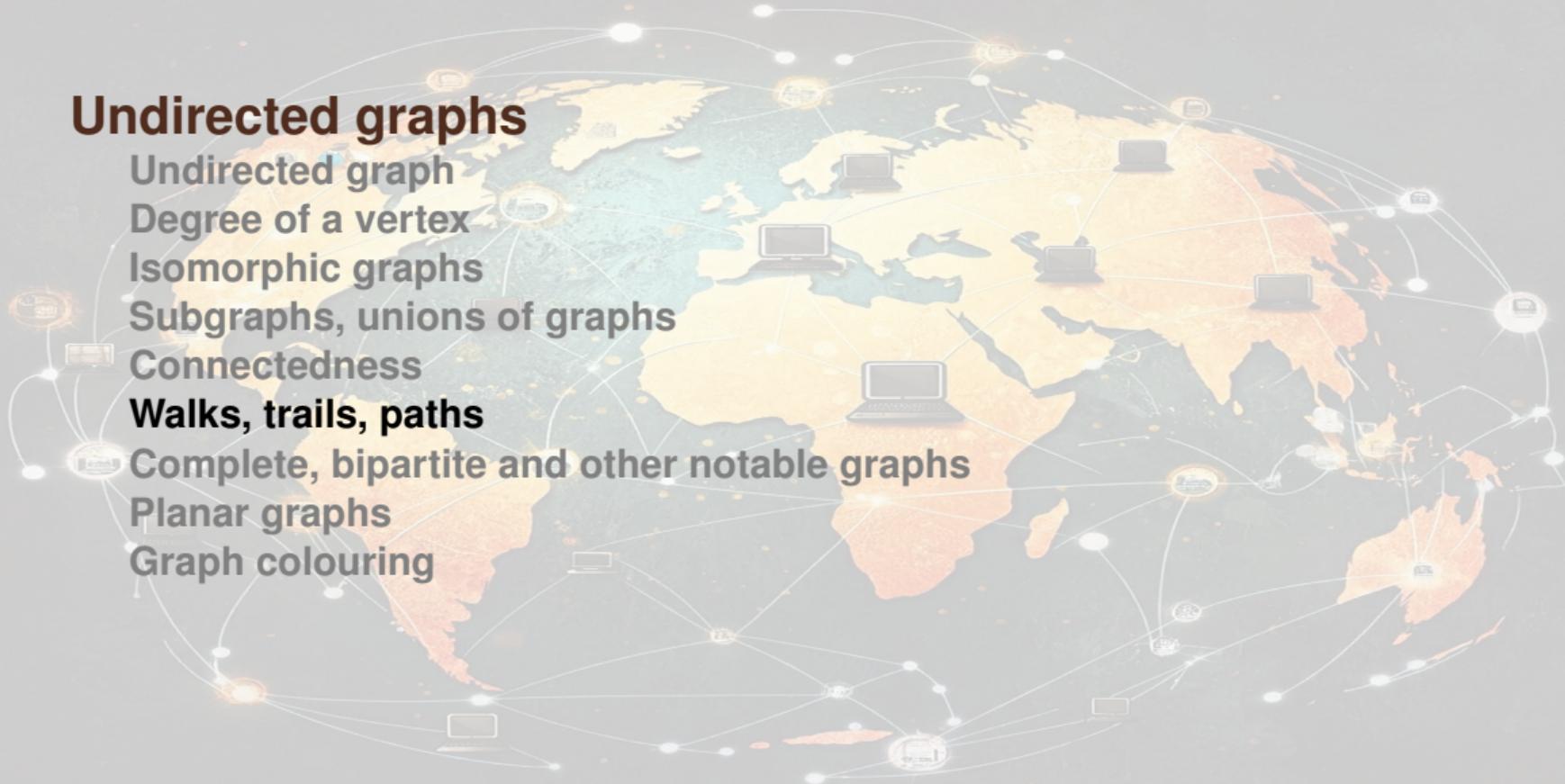
Connectedness

**Walks, trails, paths**

Complete, bipartite and other notable graphs

Planar graphs

Graph colouring



# Walk

## Definition 40 (Walk)

A **walk** in a graph  $G = (V, E)$  is a non-empty alternating sequence  $v_0 e_0 v_1 e_1 v_2 \dots e_{k-1} v_k$  of vertices and edges in  $G$  such that  $e_i = \{v_i, v_{i+1}\}$  for all  $i < k$ . This walk begins with  $v_0$  and ends with  $v_k$

## Definition 41 (Length of a walk)

The **length** of a walk is equal to the number of edges in the walk

## Definition 42 (Closed walk)

If  $v_0 = v_k$ , the walk is **closed**

## Trail and path

### Definition 43 (Trail)

If the edges in the walk are all distinct, it defines a **trail** in  $G = (V, E)$

### Definition 44 (Path)

If the vertices in the walk are all distinct, it defines a **path** in  $G$

The sets of vertices and edges determined by a trail is a subgraph

## Distance between two vertices

Definition 45 (Distance between two vertices)

The **distance**  $d(u, v)$  in  $G = (V, E)$  between two vertices  $u$  and  $v$  is the length of the shortest path linking  $u$  and  $v$  in  $G$

If no such path exists, we assume  $d(u, v) = \infty$

# Circuit and cycle

## Definition 46 (Circuit)

A trail linking  $u$  to  $v$ , containing at least 3 edges and in which  $u = v$ , is a **circuit**

## Definition 47 (Cycle)

A circuit which does not repeat any vertices (except the first and the last) is a **cycle** (or **simple circuit**)

## Definition 48 (Length of a cycle)

The **length of a cycle** is its number of edges

### Definition 49 (Eulerian trail)

A walk in an undirected multigraph  $M$  that uses each edge **exactly once** is a **Eulerian trail** of  $M$

### Definition 50 (Traversable graph)

If a graph  $G$  has a Eulerian trail, then  $G$  is a **traversable graph**

### Definition 51 (Eulerian circuit)

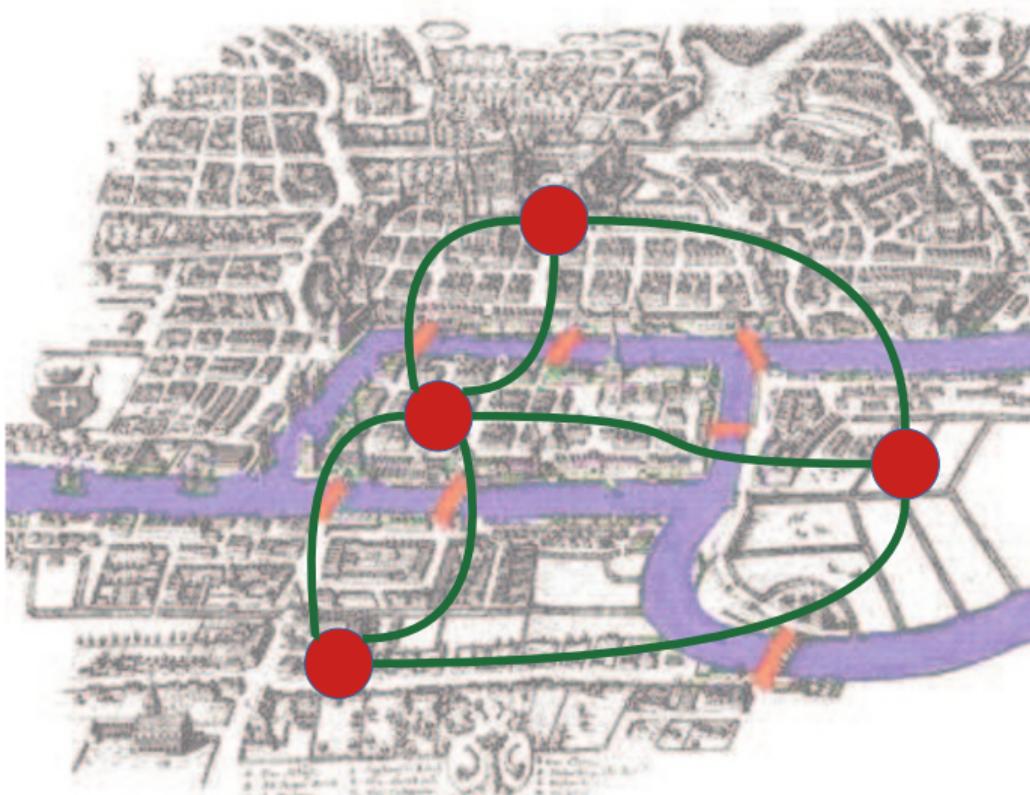
A circuit containing all the vertices and edges of a multigraph  $M$  is a **Eulerian circuit** of  $M$

### Definition 52 (Eulerian graph)

A graph (resp. multigraph) containing an Eulerian circuit is a **Eulerian graph** (resp. **Eulerian multigraph**)

# Remember Euler's bridges of Königsberg?

Cross the 7 bridges in a single walk without recrossing any of them?



### Theorem 53

*A multigraph  $M$  is traversable  $\iff M$  is connected and has exactly two odd vertices*

*Furthermore, any Eulerian trail of  $M$  begins at one of the odd vertices and ends at the other odd vertex*

### Theorem 54

*A multigraph  $M$  is Eulerian  $\iff M$  is connected and every vertex of  $M$  is even*

## Fleury's algorithm to find a Eulerian *trail*

For a connected graph with exactly 2 odd vertices

- ▶ Start at one of the odd vertices
- ▶ Marking your path as you move from vertex to vertex, travel along any edges you wish, but DO NOT travel along an edge that is a bridge for the graph formed by the EDGES THAT HAVE YET TO BE TRAVELED – unless you have to
- ▶ Continue until every edge has been traveled

RESULT: a Eulerian trail

## Fleury's algorithm to find a Eulerian *circuit*

For a connected graph with no odd vertices

- ▶ Pick any vertex as a starting point
- ▶ Marking your path as you move from vertex to vertex, travel along any edges you wish, but DO NOT travel along an edge that is a bridge for the graph formed by the EDGES THAT HAVE YET TO BE TRAVELED – unless you have to
- ▶ Continue until you return to your starting point

RESULT: a Eulerian circuit

### Definition 55 (Hamiltonian path)

A path containing all vertices of a graph  $G$  is a **Hamiltonian path** of  $G$

### Definition 56 (Traceable graph)

If a graph  $G$  has an Hamiltonian path, then  $G$  is a **traceable graph**

### Definition 57 (Hamiltonian cycle)

A cycle containing all vertices of a graph  $G$  is a **Hamiltonian cycle** of  $G$

### Definition 58 (Hamiltonian graph)

A graph containing a Hamiltonian cycle is a **Hamiltonian graph**

### Theorem 59 (Dirac's theorem)

*If  $G$  is a graph of order  $p \geq 3$  such that  $\deg(v) \geq p/2$  for every vertex  $v$  of  $G$ , then  $G$  is Hamiltonian*

### Theorem 60 (Ore's theorem)

*If  $G$  is a graph of order  $p \geq 3$  such that for all distinct nonadjacent vertices  $u$  and  $v$  of  $G$ ,*

$$\deg(u) + \deg(v) \geq p$$

*then  $G$  is Hamiltonian*

# **Undirected graphs**

Undirected graph

Degree of a vertex

Isomorphic graphs

Subgraphs, unions of graphs

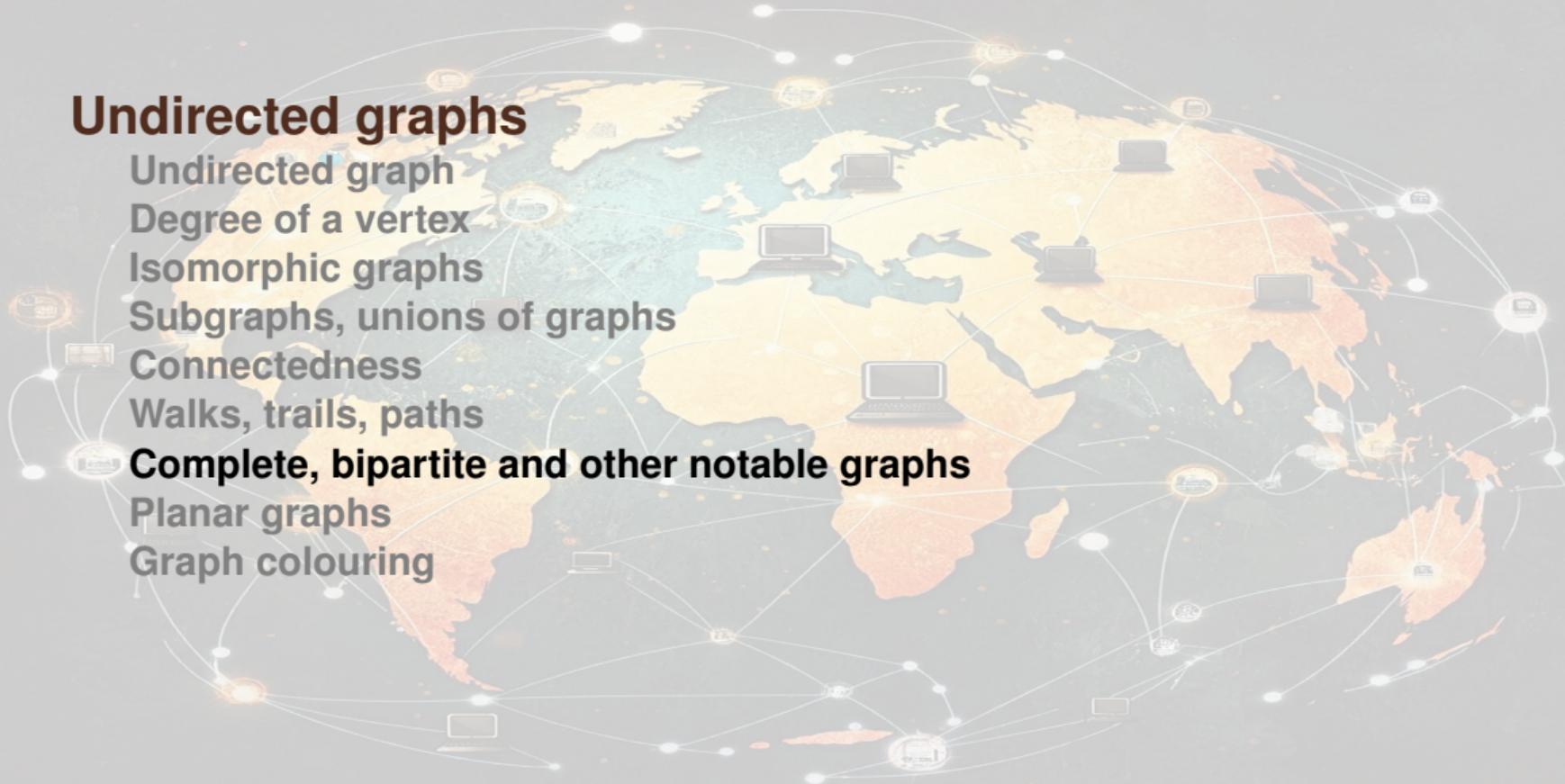
Connectedness

Walks, trails, paths

**Complete, bipartite and other notable graphs**

Planar graphs

Graph colouring



### Definition 61 (Complete graph)

A graph is complete if every two of its vertices are adjacent

### Definition 62 ( $n$ -clique)

A simple, complete graph on  $n$  vertices is called an  $n$ -**clique** and is often denoted  $K_n$

Note that a complete graph of order  $p$  is  $(p - 1)$ -regular

# Bipartite graph

## Definition 63 (Bipartite graph)

A graph is **bipartite** if its vertices can be partitioned into two sets  $V_1$  and  $V_2$ , such that no two vertices in the same set are adjacent. This graph may be written  $G = (V_1, V_2, E)$

## Definition 64 (Complete bipartite graph)

A bipartite graph in which every two vertices from the 2 different partitions are adjacent is called a **complete bipartite graph**

We often denote  $K_{p,q}$  a simple, complete bipartite graph with  $|V_1| = p$  and  $|V_2| = q$

## Some specific graphs

### Definition 65 (Tree)

Any connected graph that has no cycles is a **tree**

### Definition 66 (Cycle $C_n$ )

For  $n \geq 3$ , the **cycle**  $C_n$  is a connected graph of order  $n$  that is a cycle on  $n$  vertices

### Definition 67 (Path $P_n$ )

The **path**  $P_n$  is a connected graph that consists of  $n \geq 2$  vertices and  $n - 1$  edges. Two vertices of  $P_n$  have degree 1 and the rest are of degree 2

### Definition 68 (Star $S_n$ )

The **star** of order  $n$  is the complete bipartite graph  $K_{1,n-1}$  (1 vertex of degree  $n - 1$  and  $n - 1$  vertices of degree 1)

# **Undirected graphs**

Undirected graph

Degree of a vertex

Isomorphic graphs

Subgraphs, unions of graphs

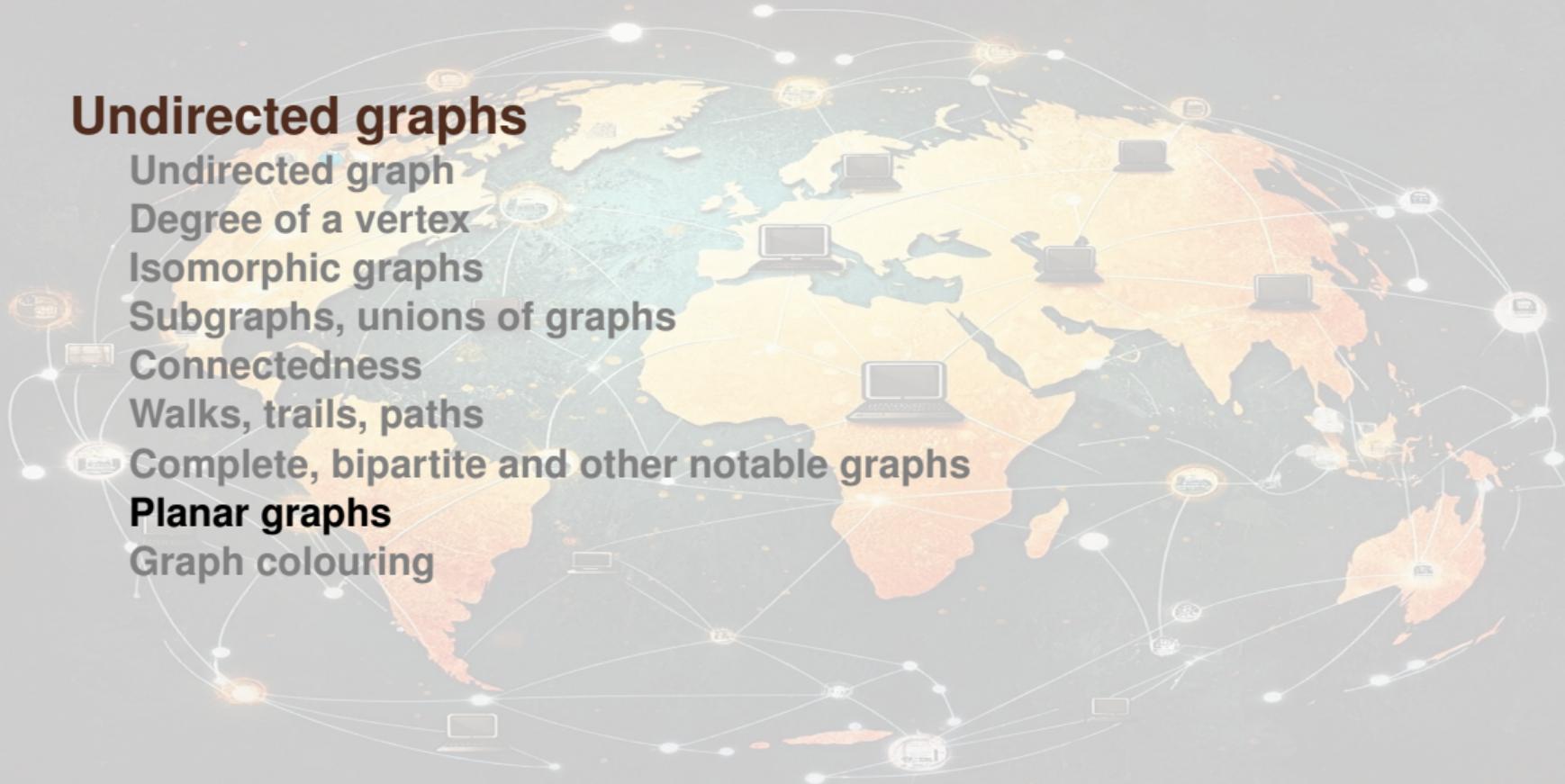
Connectedness

Walks, trails, paths

Complete, bipartite and other notable graphs

**Planar graphs**

Graph colouring



## Planar graph

### Definition 69 (Planar graph)

A graph is **planar** if it *can be drawn* in the plane with no crossing edges (except at the vertices). Otherwise, it is **nonplanar**

### Definition 70 (Plane graph)

A **plane graph** is a graph *that is drawn* in the plane with no crossing edges. (This is only possible if the graph is planar)

(To see the difference, have you ever played this game?)

Let  $G$  be a plane graph

- ▶ the connected parts of the plane are called **regions**
- ▶ vertices and edges that are incident with a region  $R$  make up a **boundary** of  $R$

### Theorem 71 (Euler's formula)

Let  $G$  be a connected plane graph with  $p$  vertices,  $q$  edges, and  $r$  regions, then

$$p - q + r = 2$$

### Corollary 72

Let  $G$  be a plane graph with  $p$  vertices,  $q$  edges,  $r$  regions, and  $k$  connected components, then

$$p - q + r = k + 1$$

### Theorem 73

Let  $G$  be a connected planar graph with  $p$  vertices and  $q$  edges, where  $p \geq 3$ , then

$$q \leq 3p - 6.$$

(a maximal connected planar graph with  $p$  vertices has  $q = 3p - 6$  edges)

### Corollary 74

If  $G$  is a planar graph, then  $\delta(G) \leq 5$ , where  $\delta(G)$  is the minimal degree of  $G$ .  
(every planar graph contains a vertex of degree less than 6)

## Two well-known non-planar graphs

$K_{3,3}$  and  $K_5$  are nonplanar

### Theorem 75 (Kuratowski Theorem)

A graph  $G$  is planar  $\iff$  it contains no subgraph isomorphic to  $K_5$  or  $K_{3,3}$  or any subdivision of  $K_5$  or  $K_{3,3}$

**Note:** If a graph  $G$  is nonplanar and  $G$  is a subgraph of  $G'$ , then  $G'$  is also nonplanar

# **Undirected graphs**

Undirected graph

Degree of a vertex

Isomorphic graphs

Subgraphs, unions of graphs

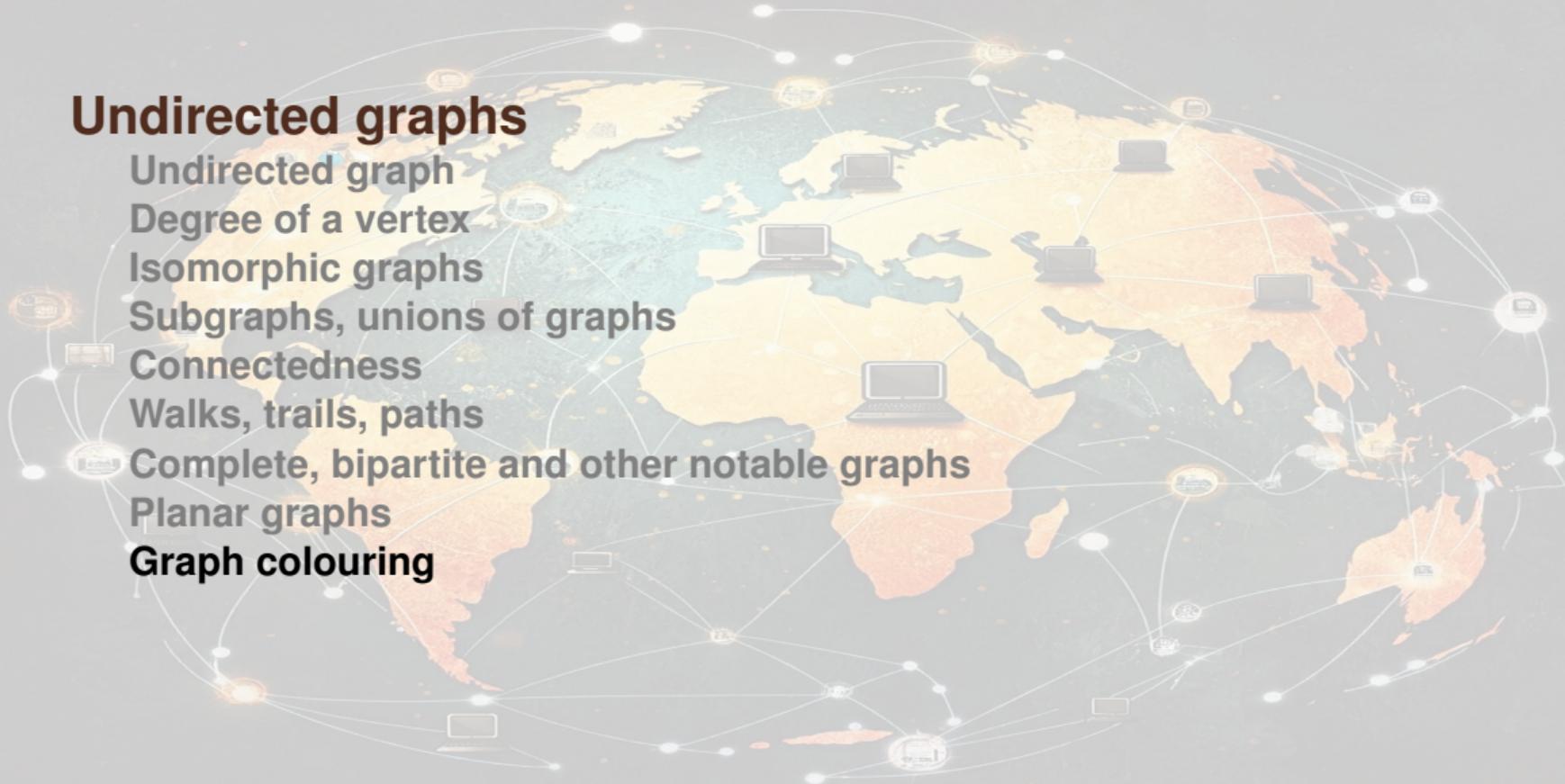
Connectedness

Walks, trails, paths

Complete, bipartite and other notable graphs

Planar graphs

**Graph colouring**



### Definition 76 (Colouring of a graph $G$ )

A **colouring** of a graph  $G$  is an assignment of colours to the vertices of  $G$  such that adjacent vertices have different colours

### Definition 77 ( $n$ -colouring of $G$ )

A  **$n$ -colouring** is a colouring of  $G$  using  $n$  colours

### Definition 78 ( $n$ -colourable)

$G$  is  **$n$ -colourable** if there exists a colouring of  $G$  that uses  $n$  colours

## Definition 79 (Chromatic number)

The **chromatic number**  $\chi(G)$  of a graph  $G$  is the minimal value  $n$  for which an  $n$ -colouring of  $G$  exists

## Property 80

- ▶  $\chi(G) = 1 \iff G$  have no edges
- ▶ If  $G = K_{n,m}$ , then  $\chi(G) = 2$
- ▶ If  $G = K_n$ , then  $\chi(G) = n$
- ▶ For any graph  $G$ ,

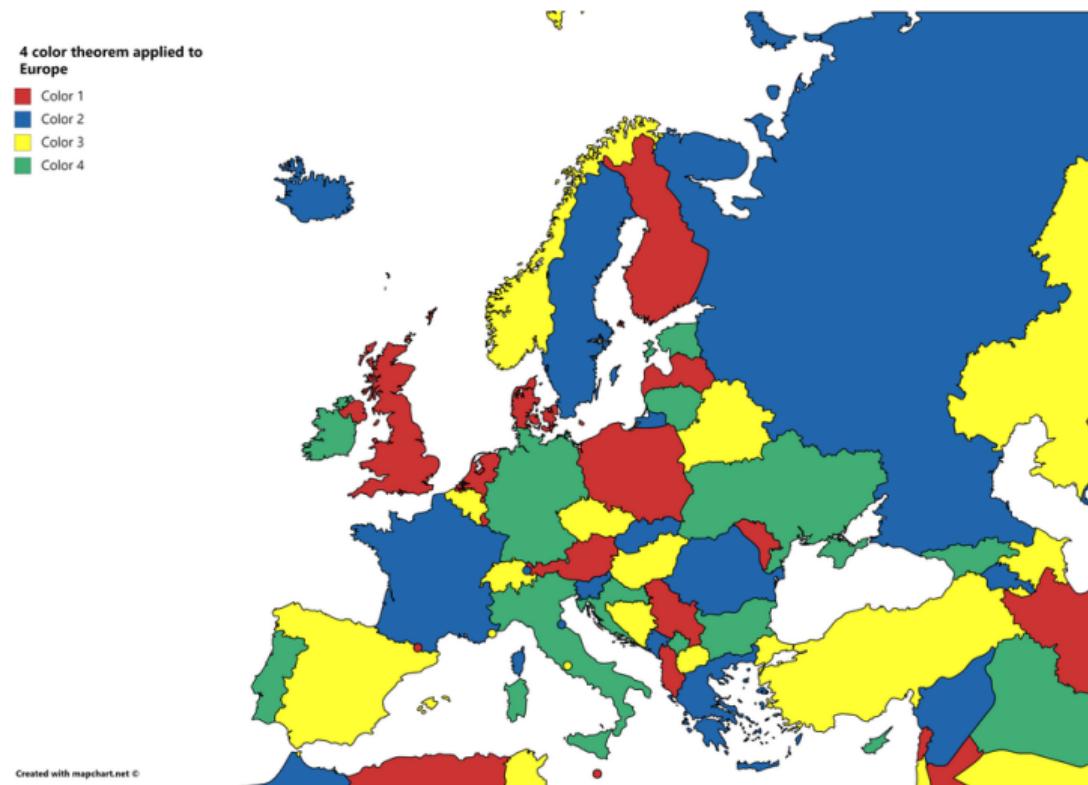
$$\chi(G) \leq 1 + \Delta(G)$$

where  $\Delta(G)$  is the maximum degree of  $G$

- ▶ If  $G$  is a planar graph, then  $\chi(G) \leq 4$

## “Real life” problem

What is the minimal number of colours to colour all countries in the map so that two adjacent countries have different colours?



## “Real life” problem

What is the minimal number of colours to colour all states in the map so that two adjacent states have different colours?

Mathematical representation:

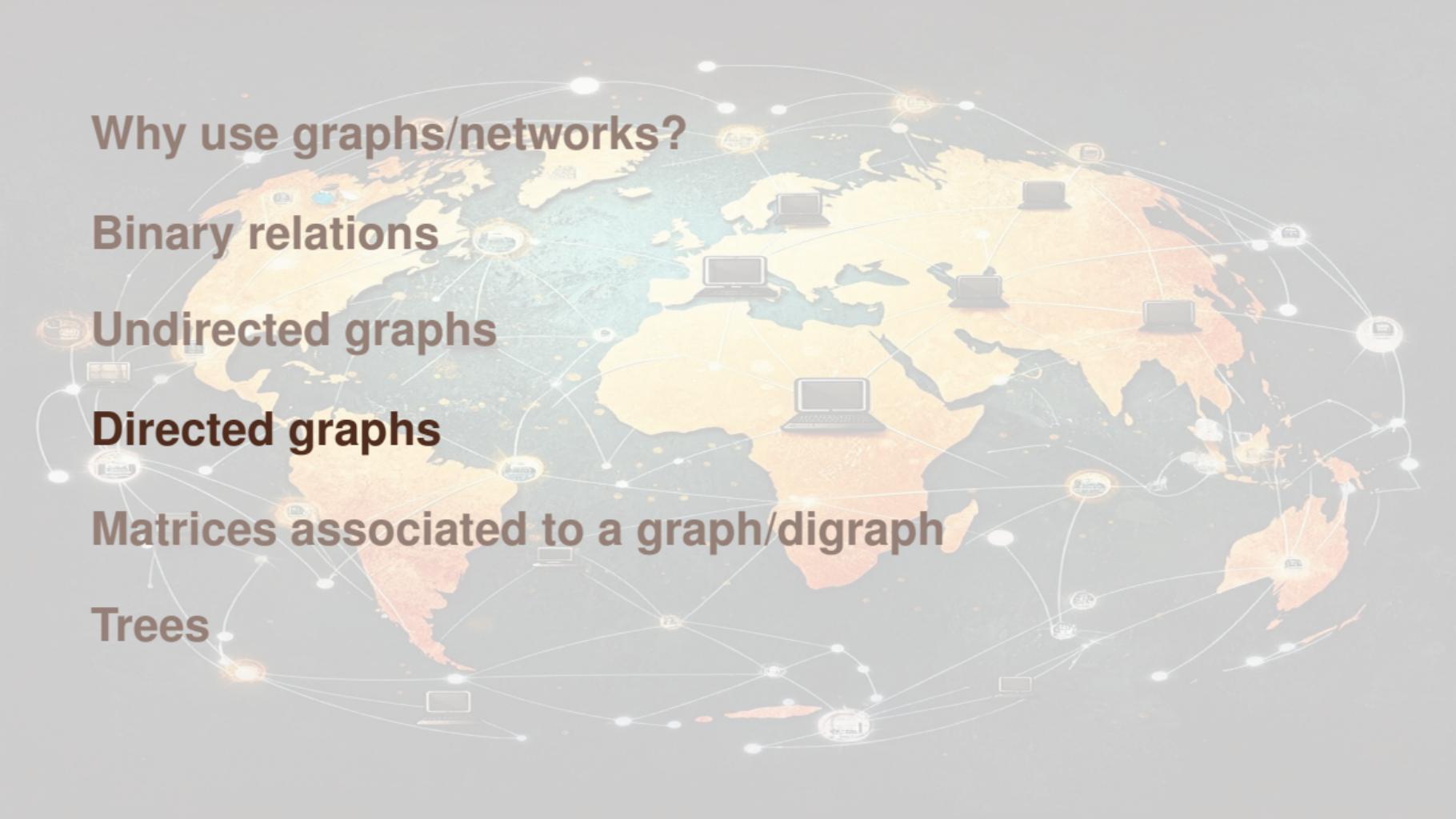
- ▶ vertices correspond to the states
- ▶ vertices are adjacent  $\iff$  the two states are adjacent (sharing an isolated point such as the “Four Corners” does not count)

### Mathematical problem

What is the chromatic number of the graph associated to the map?

## Welch-Powell algorithm for colouring a graph $G$

1. Order the vertices of  $G$  by decreasing degree. (Such an ordering may not be unique since some vertices may have the same degree)
2. Use one colour to paint the first vertex and to paint, in sequential order, each vertex on the list that is not adjacent to a vertex previously painted with this colour
3. Start again at the top of the list and repeat the process, painting previously unpainted vertices using a second colour
4. Repeat with additional colours until all vertices have been painted



Why use graphs/networks?

Binary relations

Undirected graphs

Directed graphs

Matrices associated to a graph/digraph

Trees

## Definitions

### Definition 81 (Digraph)

A directed graph (or **digraph**) is a pair  $G = (V, A)$  of sets such that

- ▶  $V$  is a set of points:  $V = \{v_1, v_2, v_3, \dots, v_p\}$
- ▶  $A$  is a set of ordered pairs of  $V$ :  $A = \{(v_i, v_j), (v_i, v_k), \dots, (v_n, v_p)\}$  or  
 $A = \{v_i v_j, v_i v_k, \dots, v_n v_p\}$

### Definition 82 (Vertex)

The elements of  $V$  are the vertices of the digraph  $G$ .  $V$  or  $V(G)$  is the vertex set of the digraph  $G$

### Definition 83 (Arc)

The elements of  $A$  are the **arcs** (directed edges) of the digraph  $G$ .  $A$  or  $A(G)$  is the arc set of the digraph  $G$

## Digraph and binary relation

A (simple) digraph  $D$  can be defined in term of a vertex set  $V$  and an irreflexive relation  $R$  over  $V$

The defining relation  $R$  of the digraph  $G$  need not be symmetric

## Directed network

### Definition 84 (Directed network)

A directed network is a digraph together with a function  $f$ ,

$$f : A \rightarrow \mathbb{R},$$

which maps the arc set  $A$  into the set of real number. The value of the arc  $uv \in A$  is  $f(uv)$

# Loops & Multiple arcs

## Definition 85 (Loop)

A **loop** is an arc with both the same ends; e.g.  $(u, u)$  is a loop

## Definition 86 (Multiple arcs)

**Multiple arcs** (or multi-arcs) are two or more arcs connecting the same two vertices

# Multidigraph/Digraph

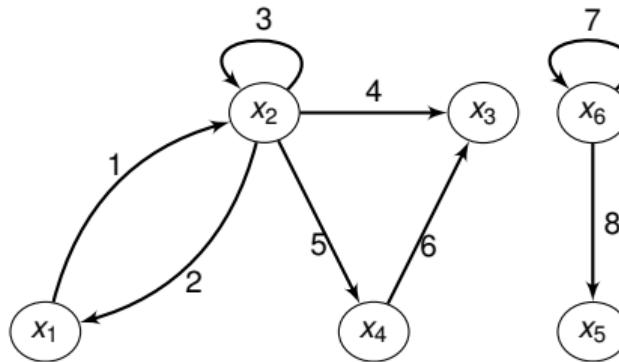
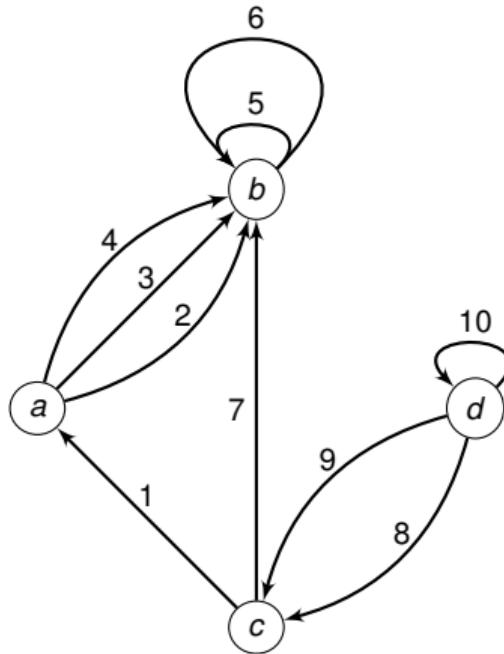
## Definition 87 (Multidigraph)

A **multidigraph** is a digraph which allows repetition of arcs or loops

## Definition 88 (Digraph)

In a digraph, no more than one arc can join any pair of vertices

## Examples



Let  $G = (V, A)$  be a digraph

### Definition 89 (Arc endpoints)

For an arc  $u = (x, y)$ , vertex  $x$  is the **initial endpoint**, and vertex  $y$  is the **terminal endpoint**

### Definition 90 (Predecessor - Successor)

If  $(u, v) \in A(G)$  is an arc of  $G$ , then

- ▶  $u$  is a **predecessor** of  $v$
- ▶  $v$  is a **successor** of  $u$

### Definition 91 (Neighbours of a vertex)

Let  $x \in V$  be a vertex. The **neighbours** of  $x$  is the set  $\Gamma(x) = \Gamma_G^+(x) \cup \Gamma_G^-(x)$ , where  $\Gamma_G^+(x)$  and  $\Gamma_G^-(x)$  are, respectively, the set of successors and predecessors of  $x$

## Sources and sinks

### Definition 92 (Directed away - Directed towards)

If  $a = (u, v) \in A(G)$  is an arc of  $G$ , then

- ▶ the arc  $a$  is said to be **directed away** from  $u$
- ▶ the arc  $a$  is said to be **directed towards**  $v$

### Definition 93 (Source - Sink)

- ▶ Any vertex which has no arcs directed towards it is a **source**
- ▶ Any vertex which has no arcs directed away from it is a **sink**

## Adjacent arcs

### Definition 94 (Adjacent arcs)

Two arcs are **adjacent** if they have at least one endpoint in common

## Arcs incident to a subset of arcs

Definition 95 (Arc incident out of  $X \subset A(G)$ )

If the initial endpoint of an arc  $u$  belongs to  $X \subset A(G)$  and if the terminal endpoint of arc  $u$  does not belong to  $X$ , then  $u$  is said to be **incident out of**  $X$ ; we write  $u \in \omega^+(X)$

Similarly, we define an **arc incident into**  $X$  and the set  $\omega^-(X)$

Finally, the set of arcs **incident to**  $X$  is denoted

$$\omega(X) = \omega^+(X) \cup \omega^-(X)$$

### Definition 96 (Subgraph of $G$ generated by $A \subset V$ )

The **subgraph** of  $G$  generated by  $A$  is the graph with  $A$  as its vertex set and with all the arcs in  $G$  that have both their endpoints in  $A$ . If  $G = (V, \Gamma)$  is a 1-graph, then the subgraph generated by  $A$  is the 1-graph  $G_A = (A, \Gamma_A)$  where

$$\Gamma_A(x) = \Gamma(x) \cap A \quad (x \in A)$$

### Definition 97 (Partial graph of $G$ generated by $V \subset U$ )

The graph  $(X, V)$  whose vertex set is  $X$  and whose arc set is  $V$ . In other words, it is graph  $G$  without the arcs  $U - V$

### Definition 98 (Partial subgraph of $G$ )

A partial subgraph of  $G$  is the subgraph of a partial graph of  $G$

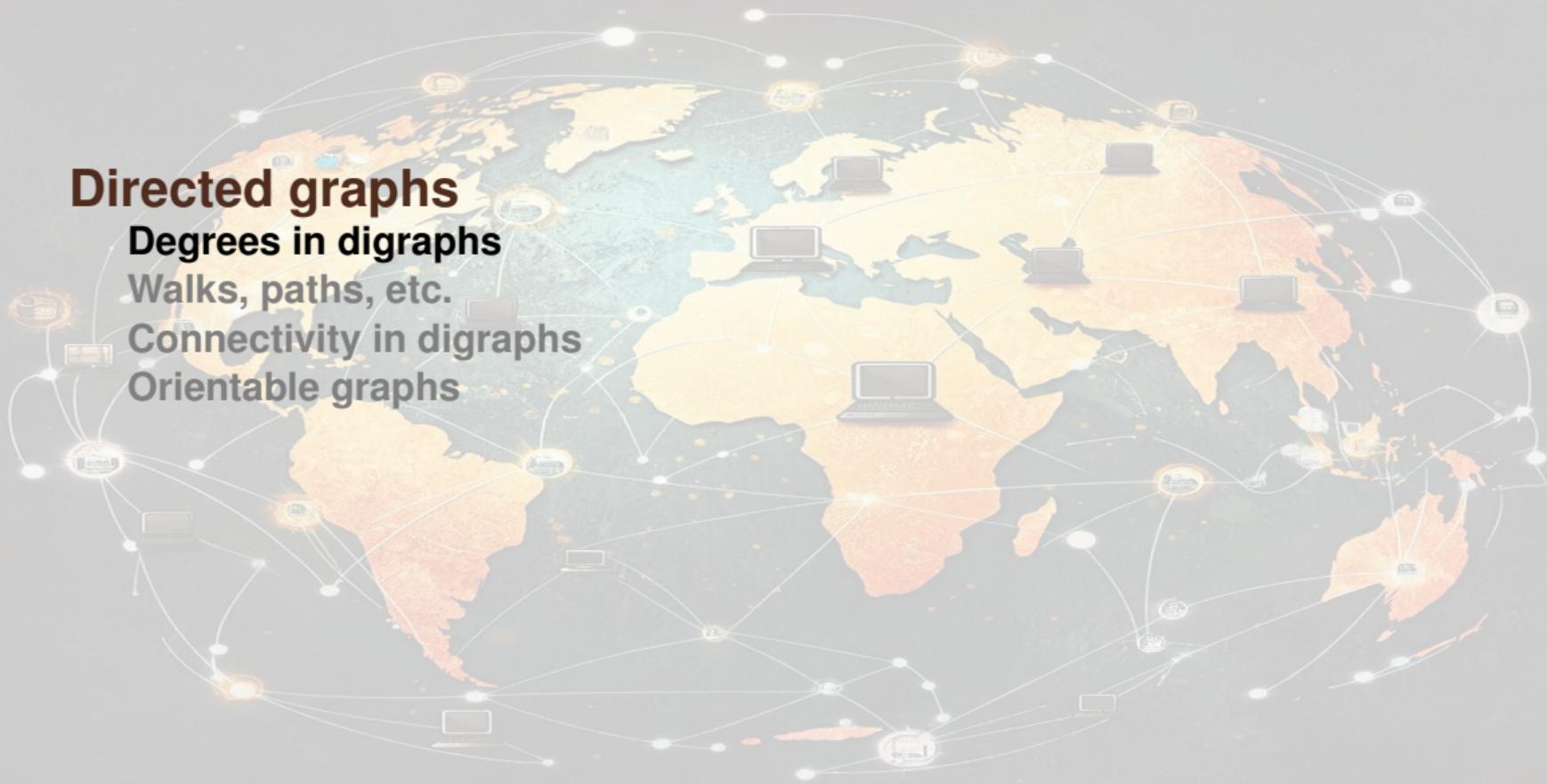
# Directed graphs

Degrees in digraphs

Walks, paths, etc.

Connectivity in digraphs

Orientable graphs



## Degree

Let  $v$  be a vertex of a digraph  $G = (V, A)$

### Definition 99 (Outdegree of a vertex)

The number of arcs directed away from a vertex  $v$ , in a digraph is called the **outdegree** of  $v$  and is written  $d_G^+(v)$

### Definition 100 (Indegree of a vertex)

The number of arcs directed towards a vertex  $v$ , in a digraph is called the **indegree** of  $v$  and is written  $d_G^-(v)$

### Definition 101 (Degree)

For any vertex  $v$  in a digraph, the **degree** of  $v$  is defined as

$$d_G(v) = d_G^+(v) + d_G^-(v)$$

## Theorem 102

*For any (di)graph, the sum of the degrees of the vertices equals twice the number of edges (arcs)*

## Corollary 103

*In any (di)graph, the sum of the degrees of the vertices is a nonnegative even integer*

## Theorem 104

*If  $G$  is a digraph with vertex set  $V(G) = \{v_1, \dots, v_p\}$  and  $q$  arcs, then*

$$\sum_{i=1}^p d_G^+(v_i) = \sum_{i=1}^p d_G^-(v_i) = q$$

### Definition 105 (Regular digraph)

A digraph  $G$  is  $r$ -regular if  $d_G^+(v) = d_G^-(v) = r$  for all  $v \in V(G)$

## Symmetric/antisymmetric digraphs

### Definition 106 (Symmetric digraph)

Let  $G = (V, A)$  be a digraph with associated binary relation  $R$ . If  $R$  is *symmetric*, the digraph is symmetric

### Definition 107 (Anti-symmetric digraph)

Let  $G = (V, A)$  be a digraph with associated binary relation  $R$ . The digraph  $G$  is **anti-symmetric** if

$$xRy \implies yRx$$

### Definition 108 (Symmetric multidigraph)

Let  $G = (V, A)$  be a multidigraph.  $G$  is symmetric if  $\forall x, y \in V(G)$ , the number of arcs from  $x$  to  $y$  equals the number of arcs from  $y$  to  $x$

# Directed graphs

Degrees in digraphs

Walks, paths, etc.

Connectivity in digraphs

Orientable graphs



# Walks

Let  $G = (V, A)$  be a digraph.

## Definition 109 (Directed walk)

A **directed walk** in a digraph  $G$  is a non-empty alternating sequence  $v_0 a_0 v_1 a_1 v_2 \dots a_{k-1} v_k$  of vertices and arcs in  $G$  such that  $a_i = (v_i, v_{i+1})$  for all  $i < k$ . This walk begins with  $v_0$  and ends with  $v_k$

## Definition 110 (Length of a directed walk)

The length of a directed walk is equal to the number of arcs in the directed walk

## Definition 111 (Closed walk)

If  $v_0 = v_k$ , the walk is closed

# Trails

Let  $G = (V, A)$  be a digraph.

## Definition 112 (Directed trail)

A directed walk in  $G$  in which all arcs are distinct is a **directed trail** in  $G$

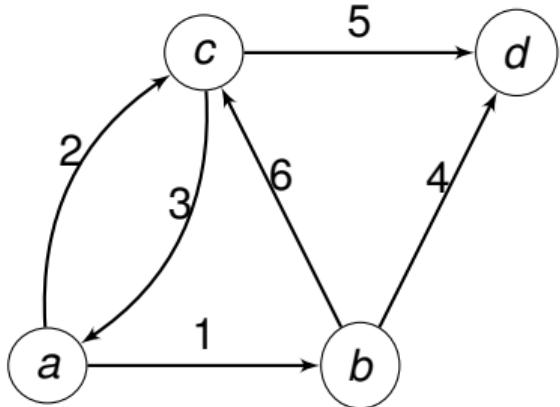
## Definition 113 (Directed path)

A directed walk in  $G$  in which all vertices are distinct is a **directed path** in  $G$

## Definition 114 (Directed cycle)

A closed walk is a **directed cycle** if it contains at least three vertices and all its vertices are distinct except for  $v_0 = v_k$

## Examples of directed cycles



Cycles:

- ▶  $\mu^1 = (1, 6, 2) = [abca]$
- ▶  $\mu^2 = (1, 6, 3) = [abca]$
- ▶  $\mu^3 = (2, 3) = [aca]$
- ▶  $\mu^4 = (1, 4, 5, 2) = [abdca]$
- ▶  $\mu^5 = (6, 5, 4) = [acdb]$
- ▶  $\mu^6 = (1, 4, 5, 3) = [abdca]$

# Directed graphs

Degrees in digraphs

Walks, paths, etc.

## Connectivity in digraphs

Orientable graphs



## Definitions

### Definition 115 (Underlying graph)

Given a digraph, the undirected graph with each arc replaced by an edge is called the **underlying graph**

### Definition 116 (Weakly connected digraph)

If the underlying graph is a connected graph, then the digraph is **weakly connected**

### Definition 117 (Strongly connected digraph)

A digraph  $G$  is **strongly connected** if for every two distinct vertices  $u$  and  $v$  of  $G$ , there exists a directed path from  $u$  to  $v$

### Definition 118 (Disconnected digraph)

A digraph is said to be **disconnected** if it is not weakly connected

## Strong connectedness is an equivalence relation

Denote  $x \equiv y$  the relation “ $x = y$ , or  $x \neq y$  and there exists a directed path in  $G$  from  $x$  to  $y$ ”.  $\equiv$  is an equivalence relation since

1.  $x \equiv y$  [reflexivity]
2.  $x \equiv y \implies y \equiv x$  [symmetry]
3.  $x \equiv y, y \equiv z \implies x \equiv z$  [transitivity]

### Definition 119 (Connected component of a graph)

Sets of the form

$$A(x_0) = \{x : x \in V, x \equiv x_0\}$$

are equivalence classes. They partition  $V$  into strongly connected sub-digraphs of  $G$  called **strongly connected components** (or **strong components**) of  $G$

A strong component in  $G$  is a maximal strongly connected subdigraph of  $G$

## Theorem 120 (Properties)

Let  $G = (V, A)$  be a digraph

- ▶ If  $G$  is strongly connected, it has only one strongly connected component
- ▶ The strongly connected components partition the vertices  $V(G)$ , with every vertex in exactly one strongly connected component

## Algorithm for determining strongly connected components in $G = (V, A)$

- ▶ Determine the strongly connected component  $C(v)$  containing the vertex  $v$ ; if  $V - C(v)$  is non-empty, re-do the same operation on the sub-digraph  $G' = (V - C(v), A')$
- ▶ To determine  $C(v)$ , the strongly connected component containing  $v$ : let  $v$  be a vertex of a digraph , which is not already in any strongly connected component
  - 1. Mark the vertex  $v$  with  $\pm$
  - 2. Mark with  $+$  all successors (not already marked with  $+$ ) of a vertex marked with  $+$
  - 3. Mark with  $-$  all predecessors (not already marked with  $-$ ) of a vertex marked with  $-$
  - 4. Repeat until no more possible marking with  $+$  or  $-$

All vertices marked with  $\pm$  belong to the same strongly connected component  $C(v)$  containing the vertex  $v$

## Condensation of a digraph

### Definition 121 (Condensation of a digraph)

The condensation  $G^*$  of a digraph  $G$  is a digraph having as vertices the strongly connected components (SCC) of  $G$  and such that there exists an arc in  $G^*$  from a SCC  $C_i$  to another SCC  $C_j$  if there is an arc in  $G$  from some vertex of  $S_i$  to a vertex of  $S_j$ .

### Definition 122 (Articulation set)

For a connected graph, a set  $X$  of vertices is called an **articulation set** (or a **cutset**) if the subgraph of  $G$  generated by  $V - X$  is not connected

### Definition 123 (Stable set)

A set  $S$  of vertices is called a **stable set** if no arc joins two distinct vertices in  $S$

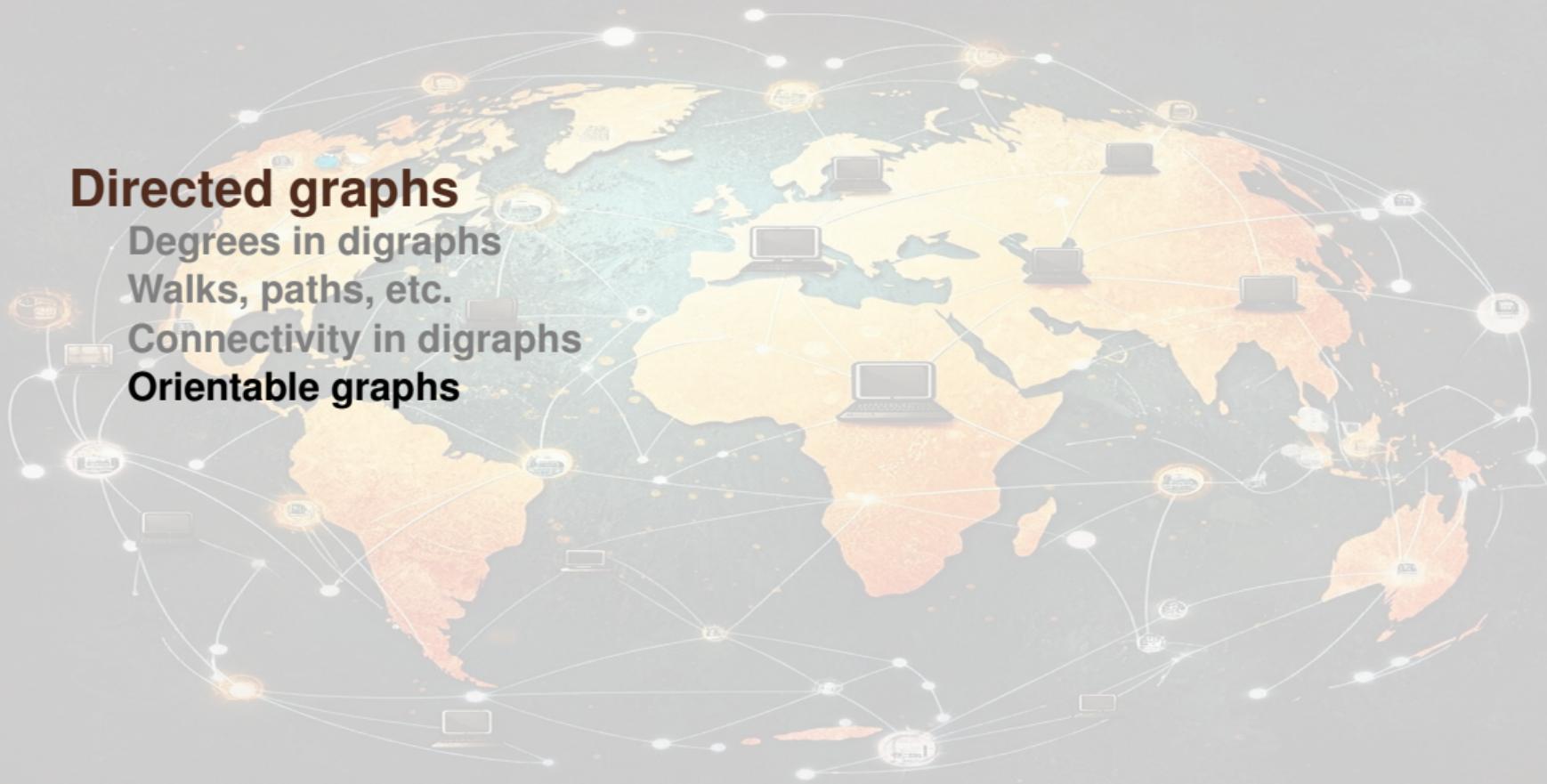
# Directed graphs

Degrees in digraphs

Walks, paths, etc.

Connectivity in digraphs

Orientable graphs



# Orientation

## Definition 124 (Orienting a graph)

Given a connected *graph*, we describe the act of assigning a direction to each edge (edge → arc) as **orienting the graph**

## Definition 125 (Strong orientation)

If the digraph resulting from orienting a graph is strongly connected, the orientation is a **strong orientation**

# Orientable graph

Definition 126 (Orientable graph)

A connected graph  $G$  is **orientable** if it admits a strong orientation

Theorem 127

*A connected graph  $G = (V, E)$  is orientable  $\iff G$  contains no bridges*

(in other words, iff every edge is contained in a cycle)

## Matrices associated to a graph/digraph

There are multiple matrices associated to a graph/digraph

The branch of graph theory that studies the properties of matrices derived from graphs and uses of these matrices in determining graph properties is *spectral graph theory*

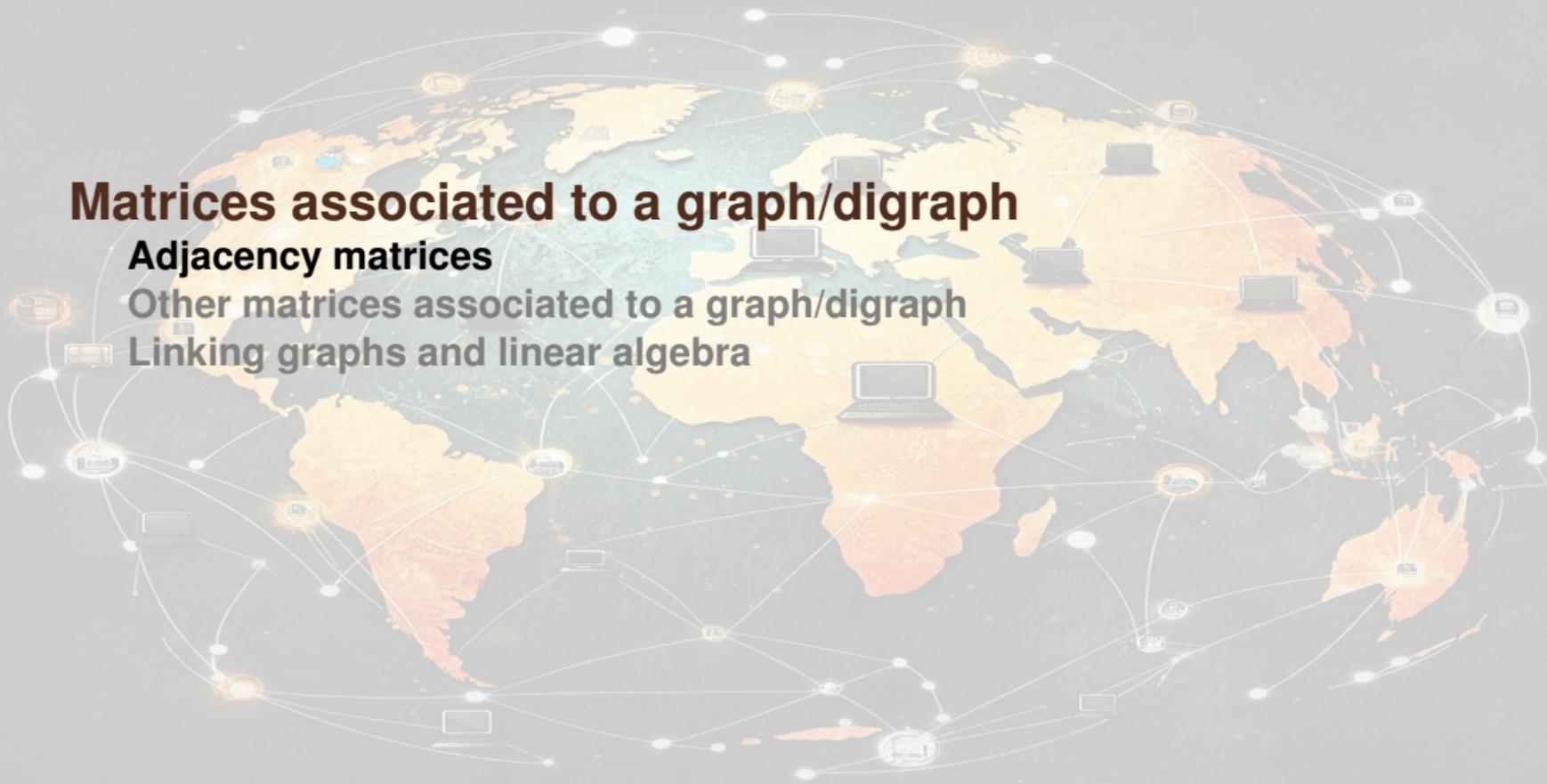
Graphs greatly simplify some problems in linear algebra and vice versa

# Matrices associated to a graph/digraph

Adjacency matrices

Other matrices associated to a graph/digraph

Linking graphs and linear algebra



## Adjacency matrix (undirected case)

Let  $G = (V, E)$  be a graph of order  $p$  and size  $q$ , with vertices  $v_1, \dots, v_p$  and edges  $e_1, \dots, e_q$

**Definition 128 (Adjacency matrix)**

The **adjacency matrix** is

$$M_A = M_A(G) = [m_{ij}]$$

is a  $p \times p$  matrix in which

$$m_{ij} = \begin{cases} 1 & \text{if } v_i \text{ and } v_j \text{ are adjacent} \\ 0 & \text{otherwise} \end{cases}$$

## Theorem 129 (Adjacency matrix and degree)

*The sum of the entries in row  $i$  of the adjacency matrix is the degree of  $v_i$  in the graph*

We often write  $A(G)$  and, reciprocally, if  $A$  is an adjacency matrix,  $G(A)$  the corresponding graph

$G$  undirected  $\implies A(G)$  symmetric

$A(G)$  has nonzero diagonal entries if  $G$  is not simple

## Adjacency matrix (directed case)

Let  $G = (V, A)$  be a digraph of order  $p$  with vertices  $v_1, \dots, v_p$

**Definition 130 (Adjacency matrix)**

The **adjacency matrix**  $M = M(G) = [m_{ij}]$  is a  $p \times p$  matrix in which

$$m_{ij} = \begin{cases} 1 & \text{if arc } v_i v_j \in A \\ 0 & \text{otherwise} \end{cases}$$

## Theorem 131 (Properties)

- ▶  $M$  is not necessarily symmetric
- ▶ The sum of any column of  $M$  is equal to the number of arcs directed towards  $v_j$
- ▶ The sum of the entries in row  $i$  is equal to the number of arcs directed away from vertex  $v_i$
- ▶ The  $(i, j)$ -entry of  $M^n$  is equal to the number of walks of length  $n$  from vertex  $v_i$  to  $v_j$

## Definition 132 (Multiplicity of a pair)

The **multiplicity** of a pair  $x, y$  is the number  $m_G^+(x, y)$  of arcs with initial endpoint  $x$  and terminal endpoint  $y$ . Let

$$m_G^-(x, y) = m_G^+(y, x)$$

$$m_G(x, y) = m_G^+(x, y) + m_G^-(x, y)$$

If  $x \neq y$ , then  $m_G(x, y)$  is number of arcs with both  $x$  and  $y$  as endpoints. If  $x = y$ , then  $m_G(x, y)$  equals twice the number of loops attached to vertex  $x$ . If  $A, B \subset V$ ,  $A \neq B$ , let

$$m_G^+(A, B) = \{u : u \in U, u = (x, y), x \in A, y \in B\}$$

$$m_G(A, B) = m_G^+(A, B) + m_G^-(A, B)$$

## Adjacency matrix of a multigraph

Definition 133 (Matrix associated with  $G$ )

If  $G$  has vertices  $x_1, x_2, \dots, x_n$ , then the **matrix associated** with  $G$  is

$$a_{ij} = m_G^+(x_i, x_j)$$

Definition 134 (Adjacency matrix)

The matrix  $a_{ij} + a_{ji}$  is the **adjacency matrix** associated with  $G$

## Adjacency matrix (multigraph case)

Definition 135 (Adjacency matrix of a multigraph)

$G$  an  $\ell$ -graph, then the adjacency matrix  $M_A = [m_{ij}]$  is defined as follows

$$m_{ij} = \begin{cases} k & \text{if arc there are } k \text{ arcs } (i,j) \in U \\ 0 & \text{otherwise} \end{cases}$$

with  $k \leq \ell$

$G$  undirected  $\implies M_A(G)$  symmetric

$M_A(G)$  has nonzero diagonal entries if  $G$  is not simple.

## Weighted adjacency matrices

Sometimes, adjacency matrices (typically for 1-graphs) have real entries, usually positive

This means that the arcs/edges have been given a weight

## Theorem 136 (Number of walks of length $n$ )

Let  $A$  be the adjacency matrix of a graph  $G = (V(G), E(G))$ , where  $V(G) = \{v_1, v_2, \dots, v_p\}$ . Then the  $(i, j)$ -entry of  $A^n$ ,  $n \geq 1$ , is the number of different walks linking  $v_i$  to  $v_j$  of length  $n$  in  $G$ .

(two walks of the same length are equal if their edges occur in exactly the same order)

Example: let  $A$  be the adjacency matrix of a graph  $G = (V(G), E(G))$ .

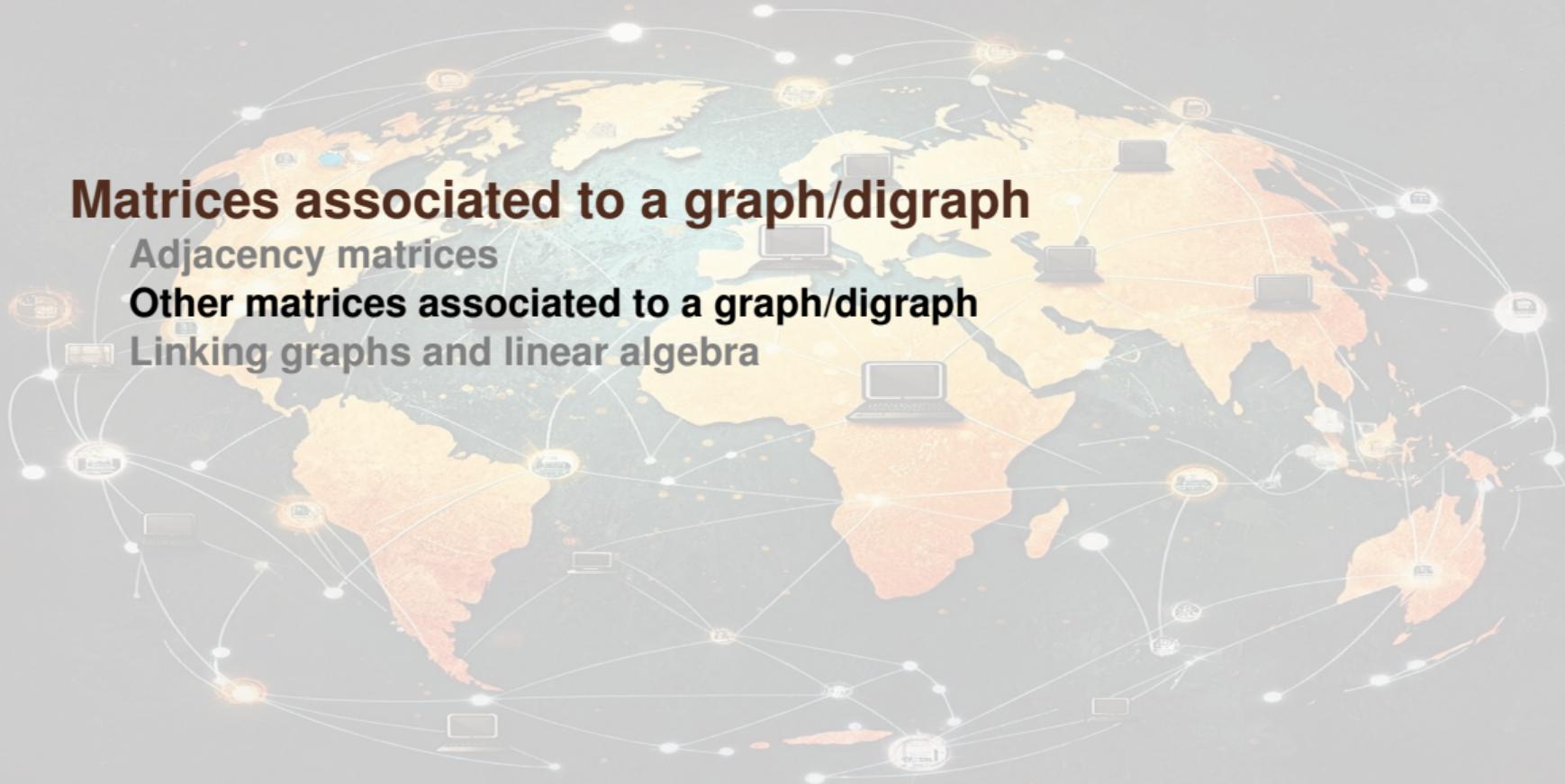
- ▶ the  $(i, i)$ -entry of  $A^2$  is equal to the degree of  $v_i$ .
- ▶ the  $(i, i)$ -entry of  $A^3$  is equal to twice the number of  $C_3$  containing  $v_i$ .

# Matrices associated to a graph/digraph

Adjacency matrices

Other matrices associated to a graph/digraph

Linking graphs and linear algebra



## Incidence matrix (undirected case)

Let  $G = (V, E)$  be a graph of order  $p$ , and size  $q$ , with vertices  $v_1, \dots, v_p$ , and edges  $e_1, \dots, e_q$

### Definition 137 (Incidence matrix)

The incidence matrix is

$$B = B(G) = [b_{ij}]$$

is that  $p \times q$  matrix in which

$$b_{ij} = \begin{cases} 1 & \text{if } v_i \text{ is incident with } e_j \\ 0 & \text{otherwise} \end{cases}$$

### Theorem 138 (Incidence matrix and degrees)

*The sum of the entries in row  $i$  of the incidence matrix is the degree of  $v_i$  in the graph*

## Incidence matrix (directed case)

Let  $G = (V, A)$  be a digraph of order  $p$  and size  $q$ , with vertices  $v_1, \dots, v_p$  and arcs  $a_1, \dots, a_q$

### Definition 139 (Incidence matrix)

The **incidence matrix**  $B = B(G) = [b_{ij}]$  is a  $p \times q$  matrix in which

$$b_{ij} = \begin{cases} 1 & \text{if arc } a_j \text{ is directed away from a vertex } v_i \\ -1 & \text{if arc } a_j \text{ is directed towards a vertex } v_i \\ 0 & \text{otherwise} \end{cases}$$

## Spectrum of a graph

We will come back to this later, but for now..

Definition 140 (Spectrum of a graph)

The **spectrum** of a graph  $G$  is the spectrum (set of eigenvalues) of its associated adjacency matrix  $M(G)$

This is regardless of the type of adjacency matrix or graph

## Degree matrix

### Definition 141 (Degree matrix)

The **degree** matrix  $D = [d_{ij}]$  for  $G$  is a  $n \times n$  diagonal matrix defined as

$$d_{ij} = \begin{cases} d_G(v_i) & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

In an undirected graph, this means that each loop increases the degree of a vertex by two

In a directed graph, the term “degree” may refer either to indegree (the number of incoming edges at each vertex) or outdegree (the number of outgoing edges at each vertex)

## Laplacian matrix

Definition 142 (Laplacian matrix)

$G = (V, A)$  a simple graph with  $n$  vertices. The **Laplacian** matrix is

$$L = D(G) - M(G)$$

where  $D(G)$  is the degree matrix and  $M(G)$  is the adjacency matrix

## Laplacian matrix (continued)

$G$  simple graph  $\implies M(G)$  only contains 1 or 0 and its diagonal elements are all 0

For directed graphs, either the indegree or outdegree is used, depending on the application

Elements of  $L$  are given by

$$\ell_{ij} = \begin{cases} d_G(v_i) & \text{if } i = j \\ -1 & \text{if } i \neq j \text{ and } v_i \text{ is adjacent to } v_j \\ 0 & \text{otherwise} \end{cases}$$

## Distance matrix

Let  $G$  be a graph of order  $p$  with vertices  $v_1, \dots, v_p$

**Definition 143 (Distance matrix)**

The distance matrix  $\Delta(G) = [d_{ij}]$  is a  $p \times p$  matrix in which

$$\delta_{ij} = d_G(v_i, v_j)$$

Note  $\delta_{ii} = 0$  for  $i = 1, \dots, p$

## Property 144

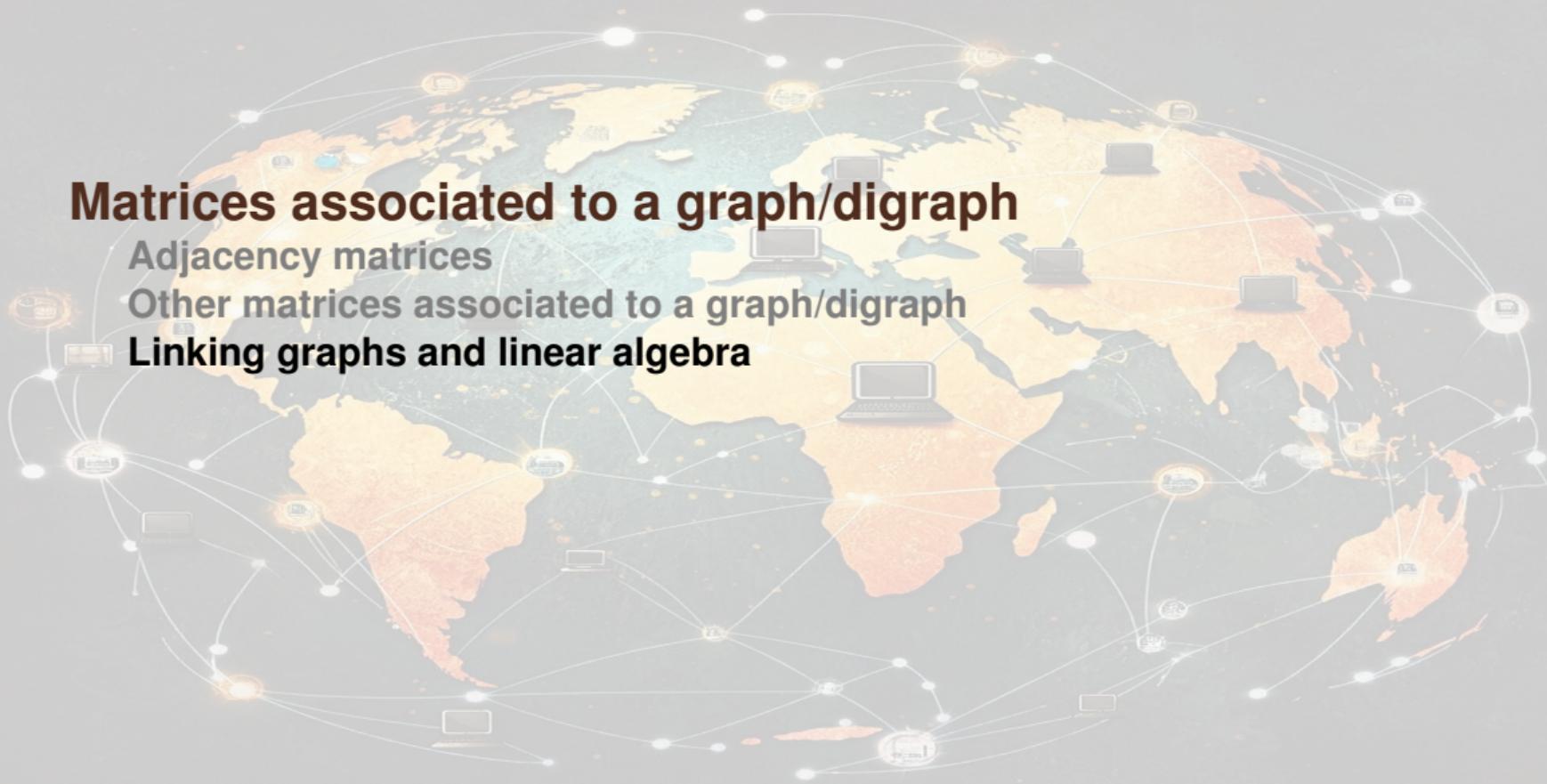
- ▶  $M$  is not necessarily symmetric
- ▶ The sum of any column of  $M$  is equal to the number of arcs directed towards  $v_j$
- ▶ The sum of the entries in row  $i$  is equal to the number of arcs directed away from vertex  $v_i$
- ▶ The  $(i, j)$ -entry of  $M^n$  is equal to the number of walks of length  $n$  from vertex  $v_i$  to  $v_j$

# Matrices associated to a graph/digraph

Adjacency matrices

Other matrices associated to a graph/digraph

Linking graphs and linear algebra



## Counting paths

### Theorem 145

$G$  a digraph and  $M_A(G)$  its adjacency matrix. Denote  $P = [p_{ij}]$  the matrix  $P = M_A^k$ . Then  $p_{ij}$  is the number of distinct paths of length  $k$  from  $i$  to  $j$  in  $G$

### Definition 146 (Irreducible matrix)

A matrix  $A \in \mathcal{M}_n$  is **reducible** if  $\exists P \in \mathcal{M}_n$ , permutation matrix, s.t.  $P^T A P$  can be written in block triangular form. If no such  $P$  exists,  $A$  is **irreducible**

### Theorem 147

$A$  irreducible  $\iff G(A)$  strongly connected

### Theorem 148

Let  $A$  be the adjacency matrix of a graph  $G$  on  $p$  vertices. A graph  $G$  on  $p$  vertices is connected  $\iff$

$$I + A + A^2 + \cdots + A^{p-1} = C$$

has no zero entries

### Theorem 149

Let  $M$  be the adjacency matrix of a digraph  $D$  on  $p$  vertices. A digraph  $D$  on  $p$  vertices is strongly connected  $\iff$

$$I + M + M^2 + \cdots + M^{p-1} = C$$

has no zero entries

## Nonnegative matrix

$A = [a_{ij}] \in \mathcal{M}_n(\mathbb{R})$  **nonnegative** if  $a_{ij} \geq 0 \forall i, j = 1, \dots, n$ ;  $\mathbf{v} \in \mathbb{R}^n$  nonnegative if  $v_i \geq 0 \forall i = 1, \dots, n$ . **Spectral radius** of  $A$

$$\rho(A) = \max_{\lambda \in \text{Sp}(A)} \{|\lambda|\}$$

$\text{Sp}(A)$  the **spectrum** of  $A$

## Perron-Frobenius (PF) theorem

### Theorem 150 (PF – Nonnegative case)

$0 \leq A \in \mathcal{M}_n(\mathbb{R})$ . Then  $\exists \mathbf{v} \geq \mathbf{0}$  s.t.

$$A\mathbf{v} = \rho(A)\mathbf{v}$$

### Theorem 151 (PF – Irreducible case)

Let  $0 \leq A \in \mathcal{M}_n(\mathbb{R})$  irreducible. Then  $\exists \mathbf{v} > \mathbf{0}$  s.t.

$$A\mathbf{v} = \rho(A)\mathbf{v}$$

$\rho(A) > 0$  and with algebraic multiplicity 1. No nonnegative eigenvector is associated to any other eigenvalue of  $A$

## Primitive matrices

Definition 152

$0 \leq A \in \mathcal{M}_n(\mathbb{R})$  **primitive** (with **primitivity index**  $k \in \mathbb{N}_+^*$ ) if  $\exists k \in \mathbb{N}_+^*$  s.t.

$$A^k > 0,$$

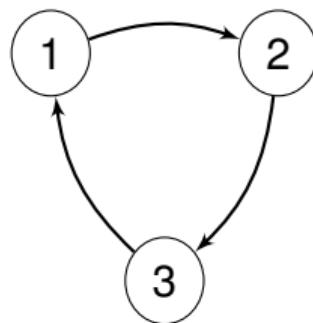
with  $k$  the smallest integer for which this is true.  $A$  **imprimitive** if it is not primitive

$A$  primitive  $\implies A$  irreducible; the converse is false

### Theorem 153

$A \in M_n(\mathbb{R})$  irreducible and  $\exists i = 1, \dots, n$  s.t.  $a_{ii} > 0 \implies A$  primitive

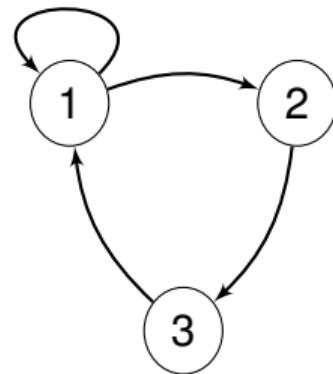
Here  $d$  is the index of imprimitivity (i.e., the number of eigenvalues that have the same modulus as  $\lambda_p = \rho(A)$ ). If  $d = 1$ , then  $A$  is primitive. We have that  $d = \text{gcd}$  of all the lengths of closed walks in  $G(A)$



Adjacency matrix

$$A = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$$

Closed walks in  $G(A)$  (lengths):  $1 \rightarrow 1$  (3),  $2 \rightarrow 2$  (3),  $2 \rightarrow 2$  (3)  $\implies \gcd = 3 \implies d = 3$  (here, all eigenvalues have modulus 1)



$$A = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$$

Closed walk  $1 \rightarrow 1$  has length 1  $\implies \gcd$  of lengths of closed walks is 1  $\implies A$  primitive

Let  $\mathbf{0} \leq A \in \mathcal{M}_n$

### Theorem 154

$A$  primitive  $\implies \exists 0 < k \leq (n - 1)n^n$  such that  $A^k > \mathbf{0}$

### Theorem 155

If  $A$  is primitive and the shortest simple directed cycle in  $G(A)$  has length  $s$ , then the primitivity index is  $\leq n + s(n - 1)$

### Theorem 156

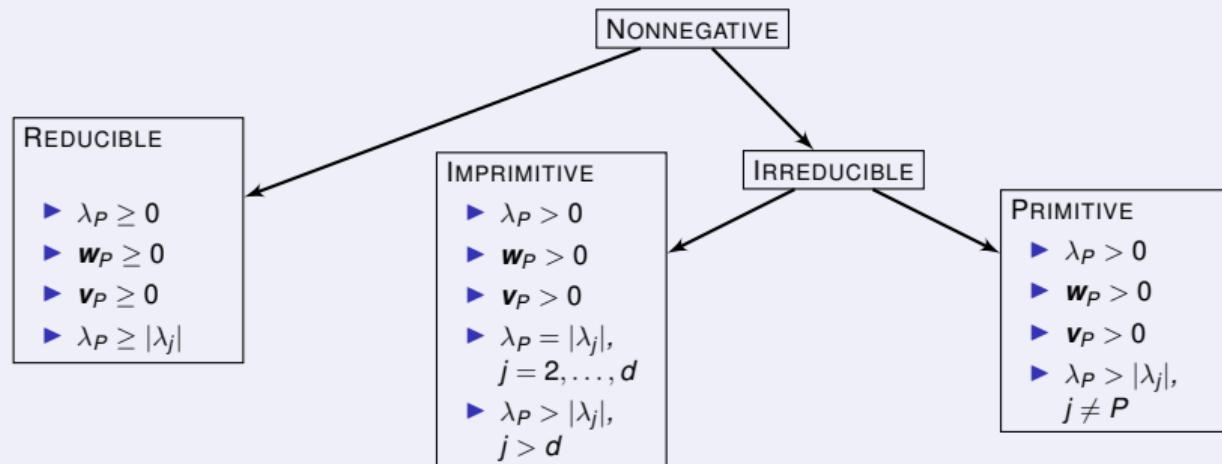
$A$  primitive  $\iff A^{n^2-2n+2} > \mathbf{0}$

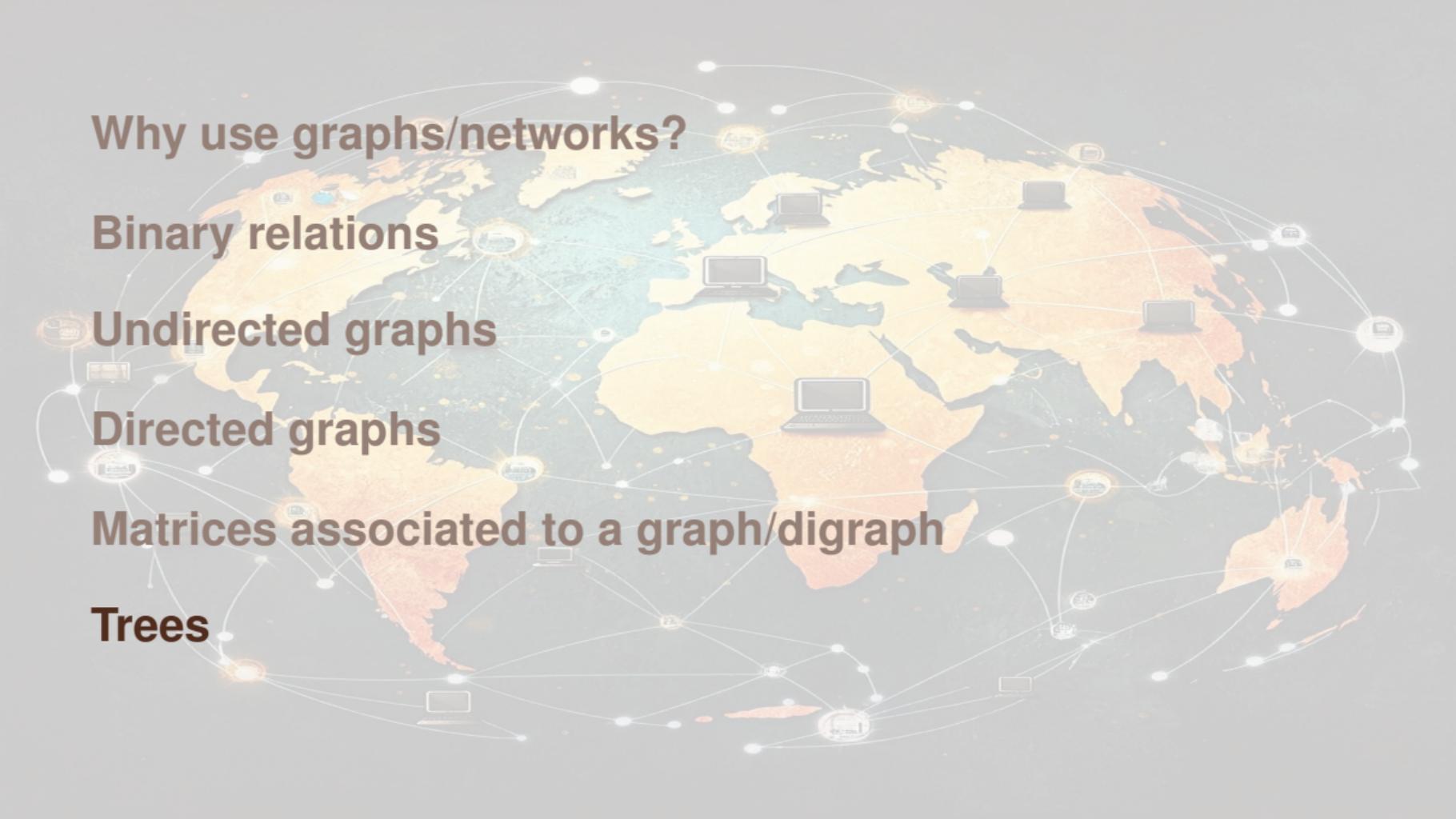
### Theorem 157

If  $A$  is irreducible and has  $d$  positive entries on the diagonal, then the primitivity index  $\leq 2n - d - 1$

## Theorem 158

$\mathbf{0} \leq A \in \mathcal{M}_n$ ,  $\lambda_P = \rho(A)$  the Perron root of  $A$ ,  $\mathbf{v}_P$  and  $\mathbf{w}_P$  the corresponding right and left Perron vectors of  $A$ , respectively,  $d$  the index of imprimitivity of  $A$  (with  $d = 1$  when  $A$  is primitive) and  $\lambda_j \in \sigma(A)$  the spectrum of  $A$ , with  $j = 2, \dots, n$  unless otherwise specified (assuming  $\lambda_1 = \lambda_P$ )





Why use graphs/networks?

Binary relations

Undirected graphs

Directed graphs

Matrices associated to a graph/digraph

Trees

## Definition 159 (Forest, trees and branches)

- ▶ A connected graph with no cycle is a **tree**
- ▶ A tree is a connected acyclic graph, its edges are called **branches**
- ▶ A graph (connected or not) without any cycle is a **forest**. Each component is a tree

(A forest is a graph whose connected components are trees)

## Is the “Karate graph” a tree?

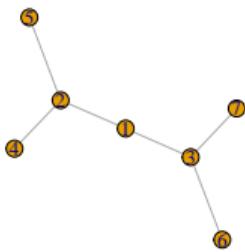
```
is_acyclic(G)
## [1] FALSE

is_tree(G)
## [1] FALSE
```

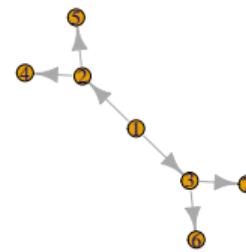
So we need friend to play with!

```
G_tu <- make_tree(7, 2, mode = "undirected")
G_td <- make_tree(7, 2)
```

An undirected tree



A (out) directed tree



## Property 160

- ▶ *Every edge of a tree is a bridge*
- ▶ *Given two vertices  $u$  and  $v$  of a tree, there is an unique path linking  $u$  to  $v$*
- ▶ *A tree with  $p$  vertices and  $q$  edges satisfies  $q = p - 1$ . Thus, a tree is minimally connected*

(First property: the deletion of any edge of a tree disconnects it)

## Every edge of a tree is a bridge

```
E(G_tu)
```

```
## + 6/6 edges from aab4c3f:  
## [1] 1--2 1--3 2--4 2--5 3--6 3--7
```

```
bridges(G_tu)
```

```
## + 6/6 edges from aab4c3f:  
## [1] 2--4 2--5 1--2 3--6 3--7 1--3
```

```
all(sort(E(G_tu)) == sort(bridges(G_tu)))
```

```
## [1] TRUE
```

## Spanning tree

Definition 161 (Spanning tree)

A **spanning tree** of a connected graph  $G$  is a subgraph of  $G$  that contains all the vertices of  $G$  and is a tree.

A graph may have many spanning trees

## Minimal spanning tree

Definition 162 (Value of a spanning tree)

The **value of a spanning tree**  $T$  of order  $p$  is

$$\sum_{i=1}^{p-1} f(e_i)$$

where  $f$  is the function that maps the edge set into  $\mathbb{R}$

Definition 163 (Minimal spanning tree)

Let  $G$  be an undirected network, and let  $T$  be a **minimal spanning tree** of  $G$ . Then  $T$  is a spanning tree whose the value is minimum

## Algorithm to find a minimal spanning tree

Let  $G = (V(G), E(G))$  be an undirected network and  $T$  be a minimal spanning tree

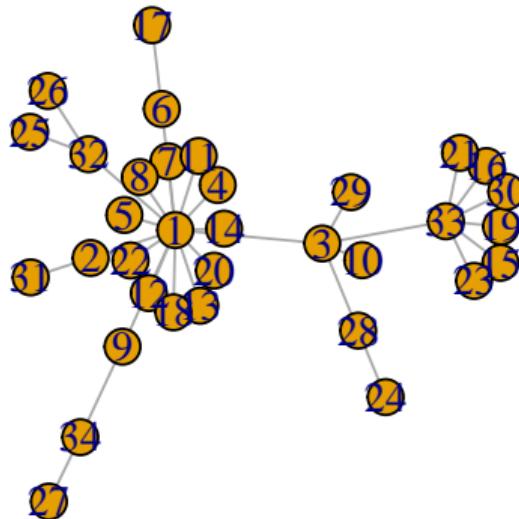
1. Sort the edges of  $G$  in increasing order by value
2.  $T = (V(G), \emptyset)$
3. For each edge  $e$  in sorted order if the endpoints of  $e$  are disconnected in  $T$   
add  $e$  to  $T$

## Finding a minimal spanning tree of the Karate graph

The function `mst` finds minimal spanning trees, using distances if no edge weights are provided

```
G_mst = mst(G)
```

## A minimal spanning tree of the Karate graph



## Minimal connector problem

- ▶ Model: a graph  $G$  such that edges represent all possible connections, and each edge has a positive value which represents its cost; an undirected network  $G$
- ▶ Solution: a minimal spanning tree  $T$  of  $G$ 
  - ▶ a spanning tree of  $G$  is a subgraph of  $G$  that contains all the vertices of  $G$  and is a tree.
  - ▶ the cost of the spanning tree is the sum of values of the edges of  $T$
  - ▶ a spanning tree such that no other spanning tree has a smaller cost is a minimal spanning tree.

## Theorem 164 (Characterisation of trees)

$H = (V, U)$  a graph of order  $|V| = n > 2$ . The following are equivalent and all characterise a tree :

1.  $H$  connected and has no cycles
2.  $H$  has  $n - 1$  arcs and no cycles
3.  $H$  connected and has exactly  $n - 1$  arcs
4.  $H$  has no cycles, and if an arc is added to  $H$ , exactly one cycle is created
5.  $H$  connected, and if any arc is removed, the remaining graph is not connected
6. Every pair of vertices of  $H$  is connected by one and only one chain

### Definition 165 (Pendant vertex)

A vertex is **pendant** if it is adjacent to exactly one other vertex

### Theorem 166

*A tree of order  $n \geq 2$  has at least two pendant vertices*

## Theorem 167

*A graph  $G = (V, U)$  has a partial graph that is a tree  $\iff G$  connected*

Recall that a partial graph is a graph generated by a subset of the arcs  
(Definition 97 slide 82)

## Spanning tree

The procedure in the proof of Theorem 167 gives a **spanning tree**

Can also build a spanning tree as follows:

- ▶ Consider any arc  $u_0$
- ▶ Find arc  $u_1$  that does not form a cycle with  $u_0$
- ▶ Find arc  $u_2$  that does not form a cycle with  $\{u_0, u_1\}$
- ▶ Continue
- ▶ When you cannot continue anymore, you have a spanning tree

## Theorem 168

$G$  connected graph with  $\geq 1$  arc. TFAE

1.  $G$  strongly connected
2. Every arc lies on a circuit
3.  $G$  contains no cocircuits

## Theorem 169

*G graph with  $\geq 1$  arc. TFAE*

1. *G is a graph without circuits*
2. *Each arc is contained in a cocircuit*

## Theorem 170

*If G is a strongly connected graph of order n, then G has a cycle basis of  $\nu(G)$  circuits*

### Definition 171 (Node, anti-node, branch)

$G = (V, U)$  strongly connected without loops and  $> 1$  vertex. For each  $x \in V$ , there is a path from it and a path to it so  $x$  has at least 2 incident arcs. Specifically,

- ▶  $x \in V$  with  $> 2$  incident arcs is a **node**
- ▶  $x \in V$  with 2 incident arcs is an **anti-node**

A path whose only nodes are its endpoints is a **branch**

### Definition 172 (Minimally connected graph)

$G$  is **minimally connected** if it is strongly connected and removal of any arc destroys strong-connectedness

A minimally connected graph is 1-graph without loops

### Definition 173 (Contraction)

$G = (V, U)$ . The **contraction** of the set  $A \subset V$  of vertices consists in replacing  $A$  by a single vertex  $a$  and replacing each arc into (resp. out of)  $A$  by an arc with same index into (resp. out of)  $a$

### Theorem 174

*G minimally connected,  $A \subset V$  generating a strongly connected subgraph of G.  
Then the contraction of A gives a minimally connected graph*

### Theorem 175

$G$  a minimally connected graph,  $G'$  be the minimally connected graph obtained by the contraction of an elementary circuit of  $G$ . Then

$$\nu(G) = \nu(G') + 1$$

### Theorem 176

$G$  minimally connected of order  $n \geq 2 \implies G$  has  $\geq 2$  anti-nodes

### Theorem 177

$G = (V, U)$ . Then the graph  $C'$  obtained by contracting each strongly connected component of  $G$  contains no circuits

## Arborescences

### Definition 178 (Root)

Vertex  $a \in V$  in  $G = (V, U)$  is a **root** if all vertices of  $G$  can be reached by paths *starting* from  $a$

Not all graphs have roots

### Definition 179 (Quasi-strong connectedness)

$G$  is **quasi-strongly connected** if  $\forall x, y \in V$ , exists  $z \in V$  (denoted  $z(x, y)$ ) to emphasize dependence on  $x, y$  from which there is a path to  $x$  and a path to  $y$

Strongly connected  $\implies$  quasi-strongly connected (take  $z(x, y) = x$ ); converse not true

Quasi-strongly connected  $\implies$  connected

# Arborescence

## Definition 180 (Arborescence)

An **arborescence** is a tree that has a root

## Lemma 181

$G = (V, U)$  has a root  $\iff G$  quasi-strongly connected

## Theorem 182

*H graph of order  $n > 1$ . TFAE (and all characterise an arborescence)*

1. *H quasi-strongly connected without cycles*
2. *H quasi-strongly connected with  $n - 1$  arcs*
3. *H tree having a root a*
4.  *$\exists a \in V$  s.t. all other vertices are connected with a by 1 and only 1 path from a*
5. *H quasi-strongly connected and loses quasi-strong connectedness if any arc is removed*
6. *H quasi-strongly connected and  $\exists a \in V$  s.t.*

$$\begin{aligned}d_H^-(a) &= 0 \\d_H^-(x) &= 1 \quad \forall x \neq a\end{aligned}$$

7. *H has no cycles and  $\exists a \in V$  s.t.*

$$\begin{aligned}d_H^-(a) &= 0 \\d_H^-(x) &\equiv 1 \quad \forall x \neq a\end{aligned}$$

### Theorem 183

$G$  has a partial graph that is an arborescence  $\iff G$  quasi-strongly connected

### Theorem 184

$G = (V, E)$  simple connected graph and  $x_1 \in V$ . It is possible to direct all edges of  $E$  so that the resulting graph  $G_0 = (V, U)$  has a spanning tree  $H$  s.t.

1.  $H$  is an arborescence with root  $x_1$
2. The cycles associated with  $H$  are circuits
3. The only elementary circuits of  $G_0$  are the cycles associated with  $H$

## Counting trees

### Proposition 185

$X$  a set with  $n$  distinct objects,  $n_1, \dots, n_p$  nonnegative integers s.t.

$n_1 + \dots + n_p = n$ . The number of ways to place the  $n$  objects into  $p$  boxes  $X_1, \dots, X_p$  containing  $n_1, \dots, n_p$  objects respectively is

$$\binom{n}{n_1, \dots, n_p} = \frac{n!}{n_1! \cdots n_p!}$$

### Proposition 186 (Multinomial formula)

Let  $a_1, \dots, a_p \in \mathbb{R}$  be  $p$  real numbers, then

$$(a_1 + \dots + a_p)^n = \sum_{n_1, \dots, n_p \geq 0} \binom{n}{n_1, \dots, n_p} (a_1)^{n_1} \cdots (a_p)^{n_p}$$

## Theorem 187

Denote  $T(n; d_1, \dots, d_n)$  the number of distinct trees  $H$  with vertices  $x_1, \dots, x_n$  and with degrees  $d_H(x_1) = d_1, \dots, d_H(x_n) = d_n$ . Then

$$T(n; d_1, \dots, d_n) = \binom{n-2}{d_1-1, \dots, d_n-1}$$

## Theorem 188

The number of different trees with vertices  $x_1, \dots, x_n$  is  $n^{n-2}$

There is a whole industry of similar results (as well as for arborescences), but we will stop here. The main point is that we are talking about a large number of possibilities..