Lecture 1: The Basics of Graph Theory

Dr. Amitabh Trehan

amitabh.trehan@durham.ac.uk

**Based on the slides of ADS-21/22 by Dr. George Mertzios*

Contents for today's lecture

- Introduction
- Graphs and types of graphs;
- Graph models;
- Basic terminology;
- Classes of graphs;
- Examples and exercises.

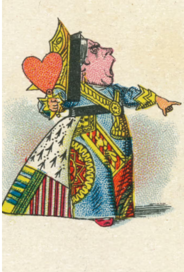
Meet the Lecturer

- Associate Professor; Head of NESTiD (Network Engineering, Science, and Theory in Durham) research group.
- PhD in Computer Science, University of New Mexico, USA. [in the research area of distributed algorithms: *Algorithms for Self-Healing Networks*]
- Algorithms and Data Structures
- Distributed algorithms and systems [Imagine networks - Internet, mobile, social, sensor, $\cdots$]
- Reliability: Self-healing networks
- Game theory

Figure: Amitabh!

The Seven Bridges of Königsberg ca. 1736

- The Queen asked: *"Can I walk my kingdom crossing every bridge exactly once?"* (**Königsberg Bridge Walk**)
- If you don't know - *"Off with your head"*



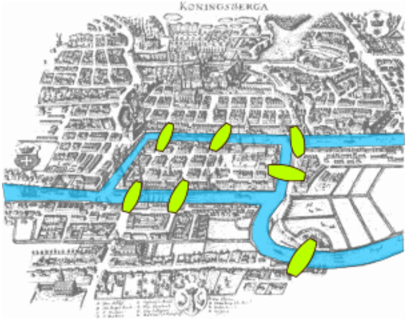


Figure: Königsberg ca. 1736: The river *Pregel*, the islands *Kneiphof* and *Lomse* and 7 bridges

The Seven Bridges of Königsberg ca. 1736

- *Leonhard Euler* came to the Queen's rescue and saved his head!

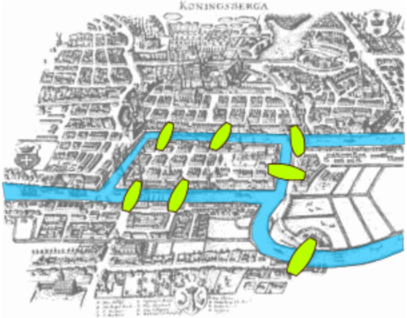
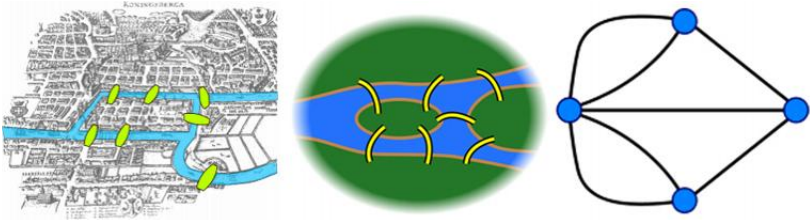


Figure: Königsberg in *Euler*'s time! The river *Pregel*, the islands *Kneiphof* and *Lomse* and 7 bridges

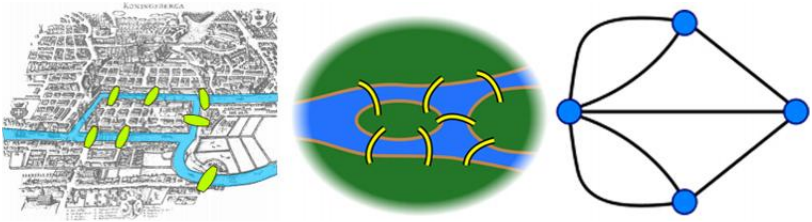
The Seven Bridges of Königsberg ca. 1736



- *Euler* realised islands were nodes and bridges edges!!
- $\cdots$ And **Graph Theory** was born!

The first `graph'!

The Seven Bridges of Königsberg ca. 1736



- *Euler* realised islands were nodes and bridges edges!!
- $\cdots$ And **Graph Theory** was born!
- So, **Do you think this Königsberg Bridge Walk is possible?**
- Answer at *PollEV.com/amitabhtrehan005*

What is a graph?

- A mathematical model (central for Computer Science)
- A representation of objects and relations between them
- The objects can be ‘anything’
- The relations are between pairs of objects

What is a graph?

- A mathematical model (central for Computer Science)
- A representation of objects and relations between them
- The objects can be 'anything'
- The relations are between pairs of objects

Example

Objects: people on Facebook
Relation: friends

What is a graph?

- A mathematical model (central for Computer Science)
- A representation of objects and relations between them
- The objects can be 'anything'
- The relations are between pairs of objects

Example

Objects: people on Facebook
Relation: friends

Example

Objects: family members (or animal species)
Relation: parent-child (or evolutionary links)

What is a graph?

- A mathematical model (central for Computer Science)
- A representation of objects and relations between them
- The objects can be ‘anything’
- The relations are between pairs of objects

Example

Objects: people on Facebook
Relation: friends

Example

Objects: family members (or animal species)
Relation: parent-child (or evolutionary links)

Example

Objects: lecturers and modules (or machines and jobs)
Relation: capability/availability

Formal definitions

Definition

A graph G is a pair $(V(G), E(G))$, where $V(G)$ is a nonempty set of vertices (or nodes) and $E(G)$ is a set of unordered pairs $\{u, v\}$ with $u, v \in V(G)$ and $u \neq v$, called the edges of G.

Formal definitions

Definition

A graph G is a pair $(V(G), E(G))$, where $V(G)$ is a nonempty set of vertices (or nodes) and $E(G)$ is a set of unordered pairs $\{u, v\}$ with $u, v \in V(G)$ and $u \neq v$, called the edges of G.

- $V(G)$ can be infinite, but all our graphs here will be finite.

Formal definitions

Definition

A graph G is a pair $(V(G), E(G))$, where $V(G)$ is a nonempty set of vertices (or nodes) and $E(G)$ is a set of unordered pairs $\{u, v\}$ with $u, v \in V(G)$ and $u \neq v$, called the edges of G.

- $V(G)$ can be infinite, but all our graphs here will be finite.
- If no confusion can arise, we write uv instead of $\{u, v\}$.

Formal definitions

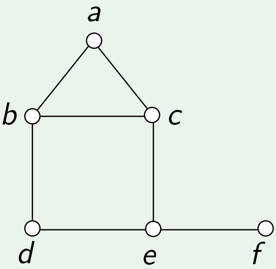
Definition

A graph G is a pair $(V(G), E(G))$, where $V(G)$ is a nonempty set of vertices (or nodes) and $E(G)$ is a set of unordered pairs $\{u, v\}$ with $u, v \in V(G)$ and $u \neq v$, called the edges of G.

- $V(G)$ can be infinite, but all our graphs here will be finite.
- If no confusion can arise, we write uv instead of $\{u, v\}$.
- If the graph G is clear from the context, we write V and E instead of $V(G)$ and $E(G)$.
- It often helps to draw graphs:
    - represent each vertex by a point, and
    - each edge by a line or curve connecting the corresponding points;
    - only endpoints of lines/curves matter, not the exact shape.

A drawing of a graph

Example



This is a drawing of the graph $G = (V, E)$ with $V = \{a, b, c, d, e, f\}$ and $E = \{ab, ac, bc, bd, ce, de, ef\}$.

Of course the drawing is not unique.

Visualizing Graphs: GraphViz

Try it out: 1) Open https://magjac.com/graphviz-visual-editor/

2) In the window on the left, copy-paste the graph definition below:

```
strict graph {
rankdir = "LR";
A -- B
A -- B
A -- D
D -- C
C -- A
}
```

3) Draw the graph from the previous slide: i.e. $G = (V, E)$ with $V = \{a, b, c, d, e, f\}$ and $E = \{ab, ac, bc, bd, ce, de, ef\}$

Types of graphs

Possible variations in definition:

Types of graphs

Possible variations in definition:

- directed graphs or digraphs — edges can have directions.

Types of graphs

Possible variations in definition:

- directed graphs or digraphs — edges can have directions.
- Formally:

Definition

A graph G is a pair $(V(G), E(G))$, where $V(G)$ is a nonempty set of vertices (or nodes) and $E(G)$ is a set of ordered pairs (u, v) with $u, v \in V(G)$ and $u \neq v$, called the edges of G.

Types of graphs

Possible variations in definition:

- directed graphs or digraphs — edges can have directions.
- Formally:

Definition

A graph G is a pair $(V(G), E(G))$, where $V(G)$ is a nonempty set of vertices (or nodes) and $E(G)$ is a set of ordered pairs (u, v) with $u, v \in V(G)$ and $u \neq v$, called the edges of G.

  - the Web graph: vertices are webpages and edges are hyperlinks.

Types of graphs

Possible variations in definition:

- directed graphs or digraphs — edges can have directions.
- Formally:

Definition

A graph G is a pair $(V(G), E(G))$, where $V(G)$ is a nonempty set of vertices (or nodes) and $E(G)$ is a set of ordered pairs (u, v) with $u, v \in V(G)$ and $u \neq v$, called the edges of G.

- the Web graph: vertices are webpages and edges are hyperlinks.
- the precedence graph: vertices are program statements, edges reflect execution order.

Types of graphs

Possible variations in definition:

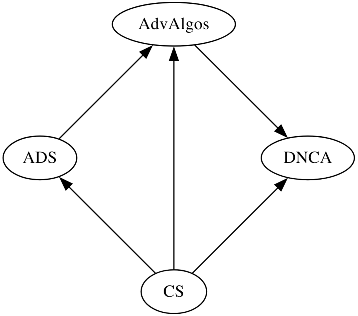
- directed graphs or digraphs — edges can have directions.
- Formally:

Definition

A graph G is a pair $(V(G), E(G))$, where $V(G)$ is a nonempty set of vertices (or nodes) and $E(G)$ is a set of ordered pairs (u, v) with $u, v \in V(G)$ and $u \neq v$, called the edges of G.

- the Web graph: vertices are webpages and edges are hyperlinks.
- the precedence graph: vertices are program statements, edges reflect execution order.
- the influence graph: vertices are people in the group, edges mean "influences"

Directed Graph



Draw the above directed graph at *https://magjac.com/graphviz-visual-editor/*

More Types of graphs

Possible variations in definition:

- multi-graphs — multiple edges are allowed between two vertices.

More Types of graphs

Possible variations in definition:

- multi-graphs — multiple edges are allowed between two vertices.
  - the air link graph: several different airlines can fly between two towns.

More Types of graphs

Possible variations in definition:

- multi-graphs — multiple edges are allowed between two vertices.
  - the air link graph: several different airlines can fly between two towns.
- pseudo-graphs — edges of the form uu, called loops, are allowed.

More Types of graphs

Possible variations in definition:

- multi-graphs — multiple edges are allowed between two vertices.
  - the air link graph: several different airlines can fly between two towns.
- pseudo-graphs — edges of the form uu, called loops, are allowed.
  - region pseudo-graph in computer graphics: Vertices are connected regions, edges mean "can get from one to the other by crossing a fence".

More Types of graphs

Possible variations in definition:

- multi-graphs — multiple edges are allowed between two vertices.
  - the air link graph: several different airlines can fly between two towns.
- pseudo-graphs — edges of the form uu, called loops, are allowed.
  - region pseudo-graph in computer graphics: Vertices are connected regions, edges mean "can get from one to the other by crossing a fence".
- vertex- or edge-weighted graphs — vertices and/or edges can have weights

More Types of graphs

Possible variations in definition:

- multi-graphs — multiple edges are allowed between two vertices.
  - the air link graph: several different airlines can fly between two towns.
- pseudo-graphs — edges of the form uu, called loops, are allowed.
  - region pseudo-graph in computer graphics: Vertices are connected regions, edges mean "can get from one to the other by crossing a fence".
- vertex- or edge-weighted graphs — vertices and/or edges can have weights
  - the road map graph: weights on edges are distances.

More Types of graphs

Possible variations in definition:

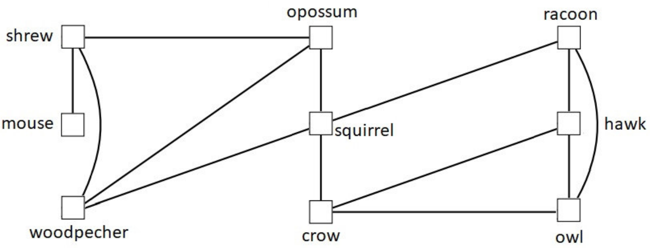
- multi-graphs — multiple edges are allowed between two vertices.
  - the air link graph: several different airlines can fly between two towns.
- pseudo-graphs — edges of the form uu, called loops, are allowed.
  - region pseudo-graph in computer graphics: Vertices are connected regions, edges mean "can get from one to the other by crossing a fence".
- vertex- or edge-weighted graphs — vertices and/or edges can have weights
  - the road map graph: weights on edges are distances.

By default, all our graphs are simple undirected or simple directed graphs (sometimes edge-weighted too), i.e. no multiple edges, no loops.

Types of graphs

Case study example: system of species

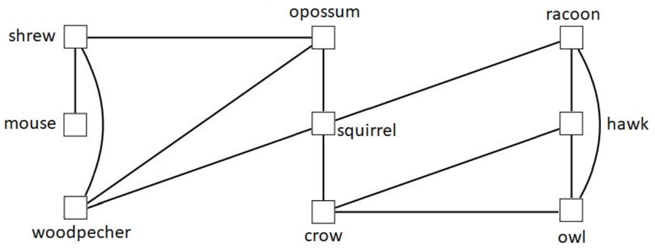
- undirected edge between two vertices: two species compete for the same food



Types of graphs

Case study example: system of species

- undirected edge between two vertices: two species compete for the same food



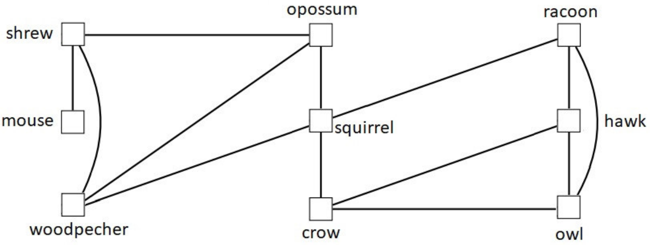
Possible questions:

- "independent set": A set of non-competing species (to live together in a zoo)

Types of graphs

Case study example: system of species

- undirected edge between two vertices: two species compete for the same food



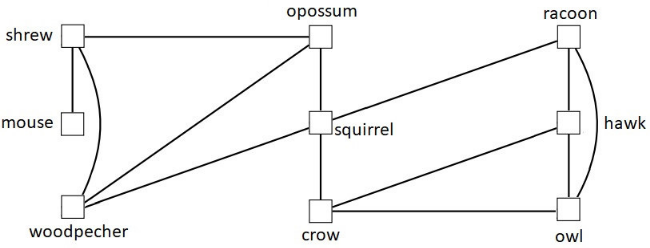
Possible questions:

- “independent set”: A set of non-competing species
  (to live together in a zoo)
- **Which of the following are independent sets?**

Types of graphs

Case study example: System of Species

- undirected edge between two vertices: two species compete for the same food



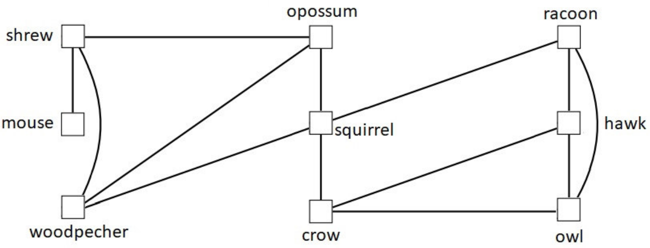
questions:

- "Maximum independent set": largest set of non-competing species (to live together in a zoo)

Types of graphs

Case study example: System of Species

- undirected edge between two vertices: two species compete for the same food



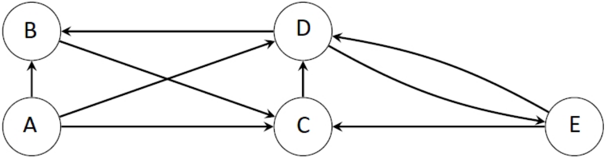
questions:

- "Maximum independent set": largest set of non-competing species (to live together in a zoo)
- "Minimum coloring": partition into the smallest number of independent sets (smallest number of rooms in the zoo)

Types of graphs

Case study example: A Social Network

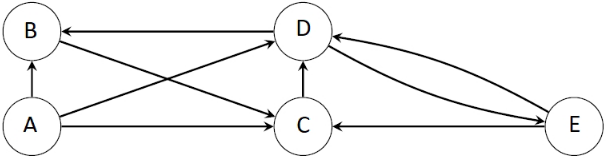
- vertices are persons
- directed edge from x to y: person x influences person y



Types of graphs

Case study example: A Social Network

- vertices are persons
- directed edge from x to y: person x influences person y



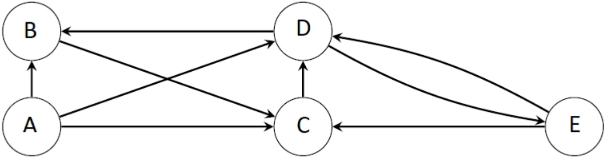
Possible question:

- "(Min) Dominating set": smallest number of persons, which collectively influence all others (best influencer set). e.g.???

Types of graphs

Case study example: A Social Network

- vertices are persons
- directed edge from x to y: person x influences person y



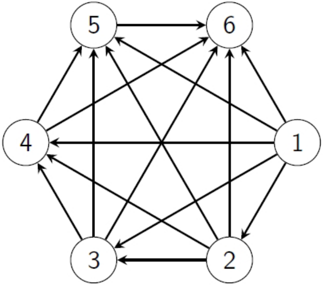
Possible question:

- "(Min) Dominating set": smallest number of persons, which collectively influence all others (best influencer set). $=> \{A, E\}, \{A, D\}$

Types of graphs

Case study example: sports tournament (**Tournament Graph**)

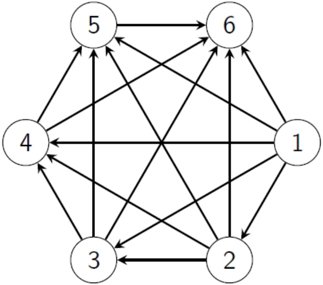
- vertices are teams
- directed edge from x to y: team x wins over team y or team x *dominates* team y



Types of graphs

Case study example: sports tournament (**Tournament Graph**)

- vertices are teams
- directed edge from x to y: team x wins over team y or team x *dominates* team y

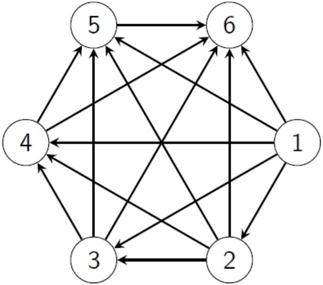


- team 1: absolute winner ("unconquerable!")
- team 6: absolute loser

Types of graphs

Case study example: sports tournament (**Tournament Graph**)

- vertices are teams
- directed edge from x to y: team x wins over team y or team x *dominates* team y



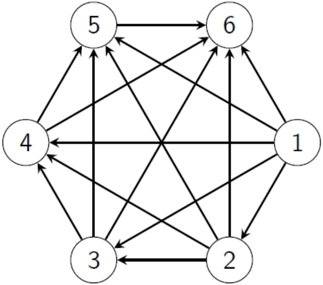
- team 1: absolute winner ("unconquerable!")
- team 6: absolute loser

Q: Does an absolute winner / loser always exist?

Types of graphs

Case study example: sports tournament (**Tournament Graph**)

- vertices are teams
- directed edge from x to y: team x wins over team y or team x *dominates* team y



- team 1: absolute winner ("unconquerable!")
- team 6: absolute loser

Q: Does an absolute winner / loser always exist?
No!

More examples of graph models

Graphs can be useful to express conflicting situations between objects.

More examples of graph models

Graphs can be useful to express conflicting situations between objects.

- Vertices: base stations for mobile phones, Edges: overlapping service areas

More examples of graph models

Graphs can be useful to express conflicting situations between objects.

- Vertices: base stations for mobile phones, Edges: overlapping service areas
- Vertices: traffic flows at a junction, Edges: conflicting flows

More examples of graph models

Graphs can be useful to express conflicting situations between objects.

- Vertices: base stations for mobile phones, Edges: overlapping service areas
- Vertices: traffic flows at a junction, Edges: conflicting flows

Graphs can be useful for analysing strategies and solutions.

More examples of graph models

Graphs can be useful to express conflicting situations between objects.

- Vertices: base stations for mobile phones, Edges: overlapping service areas
- Vertices: traffic flows at a junction, Edges: conflicting flows

Graphs can be useful for analysing strategies and solutions.

- Vertices: states in a game, Edges: transitions between states.

More examples of graph models

Graphs can be useful to express conflicting situations between objects.

- Vertices: base stations for mobile phones, Edges: overlapping service areas
- Vertices: traffic flows at a junction, Edges: conflicting flows

Graphs can be useful for analysing strategies and solutions.

- Vertices: states in a game, Edges: transitions between states.
- Vertices: steps in a solution, Edges: transitions between steps.

More examples of graph models

Example of graph problems for finding "good strategies" in a game (e.g. chess):

- (Directed) transition edges of player 1 are red, while edges of player 2 are blue
- Some vertices (states) are winning states for player 1, some for player 2

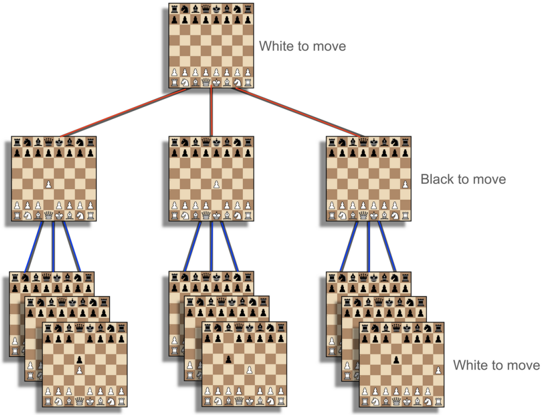


Figure: *https://chrisbutner.github.io/ChessCoach/high-level-explanation.html*

More examples of graph models

Example of graph problems for finding "good strategies" in a game (e.g. chess):

- (Directed) transition edges of player 1 are red, while edges of player 2 are blue
- Some vertices (states) are winning states for player 1, some for player 2

Q. 1: Does there exist an alternating red-blue path from the initial state to a winning state of player 1? [if not, no chance at all for player 1!]

More examples of graph models

Example of graph problems for finding "good strategies" in a game (e.g. chess):

- (Directed) transition edges of player 1 are red, while edges of player 2 are blue
- Some vertices (states) are winning states for player 1, some for player 2

Q. 1: Does there exist an alternating red-blue path from the initial state to a winning state of player 1? [if not, no chance at all for player 1!]

Q. 2: (winning strategy for pl. 1). Starting from the initial state, is there a red transition (of pl. 1) such that, for any follow-up blue transition (of pl. 2) there exists a red transition such that , . . . , such that there exists a red transition leading to a winning state for player 1?

More examples of graph models

Example of graph problems for finding "good strategies" in a game (e.g. chess):

- (Directed) transition edges of player 1 are red, while edges of player 2 are blue
- Some vertices (states) are winning states for player 1, some for player 2

Q. 1: Does there exist an alternating red-blue path from the initial state to a winning state of player 1? [if not, no chance at all for player 1!]

Q. 2: (winning strategy for pl. 1). Starting from the initial state, is there a red transition (of pl. 1) such that, for any follow-up blue transition (of pl. 2) there exists a red transition such that , . . . , such that there exists a red transition leading to a winning state for player 1?

- Question 1 is relatively simple to answer (a type of "reachability problem"), if the graph of game states is small. However, usually this graph is huge!

More examples of graph models

Example of graph problems for finding "good strategies" in a game (e.g. chess):

- (Directed) transition edges of player 1 are red, while edges of player 2 are blue
- Some vertices (states) are winning states for player 1, some for player 2

Q. 1: Does there exist an alternating red-blue path from the initial state to a winning state of player 1? [if not, no chance at all for player 1!]

Q. 2: (winning strategy for pl. 1). Starting from the initial state, is there a red transition (of pl. 1) such that, for any follow-up blue transition (of pl. 2) there exists a red transition such that , . . . , such that there exists a red transition leading to a winning state for player 1?

- Question 1 is relatively simple to answer (a type of "reachability problem"), if the graph of game states is small. However, usually this graph is huge!
- Question 2 is among the hardest questions that one can ask, even when the graph is small. Imagine when the graph is huge (as in a graph of game states)...

Terminology

Terminology

Definitions

Let G be a graph and uv an edge in it. Then

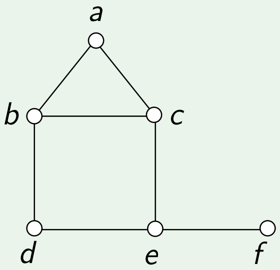
- u and v are called endpoints of the edge uv
- u and v are called neighbours or adjacent vertices
- uv is said to be incident to u (and to v)
- if vw is also an edge (where $w \neq u$) then uv and vw are called adjacent.

Terminology

Definitions

Let G be a graph and uv an edge in it. Then

- u and v are called endpoints of the edge uv
- u and v are called neighbours or adjacent vertices
- uv is said to be incident to u (and to v)
- if vw is also an edge (where $w \neq u$) then uv and vw are called adjacent.



More terminology

Definitions

Let $G = (V, E)$ be a graph. The neighbourhood of a vertex $v \in V$, notation $N(v)$, is the set of neighbours of v, i.e., $N(v) = \{ u \in V \mid uv \in E \}$.

The degree of a vertex $v \in V$, notation $deg(v)$, is the number of neighbours of v, i.e. $deg(v) = |N(v)|$.

With $\delta(G)$ or δ we denote the smallest degree in G, and with $\Delta(G)$ or Δ the largest degree.

A vertex with degree 0 will be called an isolated vertex.

A vertex with degree 1 an end vertex or a pendant vertex.

More terminology

Definitions

Let $G = (V, E)$ be a graph. The neighbourhood of a vertex $v \in V$, notation $N(v)$, is the set of neighbours of v, i.e., $N(v) = \{ u \in V \mid uv \in E \}$.

The degree of a vertex $v \in V$, notation $deg(v)$, is the number of neighbours of v, i.e. $deg(v) = |N(v)|$.

With $\delta(G)$ or δ we denote the smallest degree in G, and with $\Delta(G)$ or Δ the largest degree.

A vertex with degree 0 will be called an isolated vertex.

A vertex with degree 1 an end vertex or a pendant vertex.

Definition

A subgraph $G' = (V', E')$ of $G = (V, E)$ is a graph with $V' \subseteq V$ and $E' \subseteq E$.

More terminology

Definitions

Let $G = (V, E)$ be a graph. The neighbourhood of a vertex $v \in V$, notation $N(v)$, is the set of neighbours of v, i.e., $N(v) = \{ u \in V \mid uv \in E \}$.

The degree of a vertex $v \in V$, notation $deg(v)$, is the number of neighbours of v, i.e. $deg(v) = |N(v)|$.

With $\delta(G)$ or δ we denote the smallest degree in G, and with $\Delta(G)$ or Δ the largest degree.

A vertex with degree 0 will be called an isolated vertex.

A vertex with degree 1 an end vertex or a pendant vertex.

Definition

A subgraph $G' = (V', E')$ of $G = (V, E)$ is a graph with $V' \subseteq V$ and $E' \subseteq E$. This subgraph is called proper if $G' \neq G$ and spanning if $V' = V$.

More terminology

Definitions

Let $G = (V, E)$ be a graph. The neighbourhood of a vertex $v \in V$, notation $N(v)$, is the set of neighbours of v, i.e., $N(v) = \{ u \in V \mid uv \in E \}$.

The degree of a vertex $v \in V$, notation $deg(v)$, is the number of neighbours of v, i.e. $deg(v) = |N(v)|$.

With $\delta(G)$ or δ we denote the smallest degree in G, and with $\Delta(G)$ or Δ the largest degree.

A vertex with degree 0 will be called an isolated vertex.

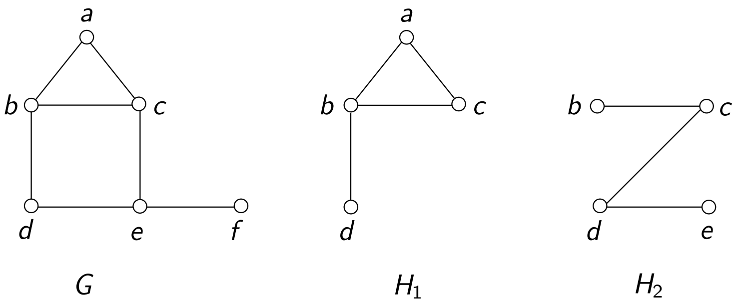
A vertex with degree 1 an end vertex or a pendant vertex.

Definition

A subgraph $G' = (V', E')$ of $G = (V, E)$ is a graph with $V' \subseteq V$ and $E' \subseteq E$. This subgraph is called proper if $G' \neq G$ and spanning if $V' = V$.

It is called induced subgraph if E' contains all edges of E between vertices of V', i.e. it is obtained by just removing from G all vertices of $V \setminus V'$ (and their edges).

Examples of the above concepts



Examples

The graph H_1 is a subgraph of G, but not a spanning subgraph, so it is also a proper subgraph of G.
H_2 is not a subgraph of $G : cd \notin E(G)$.
The pair $(\{a, b, c\}, \{ab, bd\})$ is no subgraph of G either, since it is not a graph.

First theorem in Graph Theory

Can you guess the relationship between the sum of the degrees of the vertices of a graph G and the number of edges of G?

First theorem in Graph Theory

Can you guess the relationship between the sum of the degrees of the vertices of a graph G and the number of edges of G?

Theorem (Handshaking Lemma)

Let $G = (V, E)$ be a graph. Then $\sum_{v \in V} deg(v) = 2|E|$.

How to prove this?

First theorem in Graph Theory

Can you guess the relationship between the sum of the degrees of the vertices of a graph G and the number of edges of G?

Theorem (Handshaking Lemma)

Let $G = (V, E)$ be a graph. Then $\sum_{v \in V} deg(v) = 2|E|$.

Proof.

Every edge has two endpoints and contributes one to each of their degrees, so contributes two to the sum of the degrees of all the vertices of V. $\square$

First theorem in Graph Theory

Can you guess the relationship between the sum of the degrees of the vertices of a graph G and the number of edges of G?

Theorem (Handshaking Lemma)

Let $G = (V, E)$ be a graph. Then $\sum_{v \in V} deg(v) = 2|E|$.

Proof.

Every edge has two endpoints and contributes one to each of their degrees, so contributes two to the sum of the degrees of all the vertices of V. $\square$

This simple relationship can be useful for proving non-existence of graphs with certain properties.

First theorem in Graph Theory

Corollary

In every undirected graph G, the number of vertices with an odd degree (i.e. number of neighbours) is even.

First theorem in Graph Theory

Corollary

In every undirected graph G, the number of vertices with an odd degree (i.e. number of neighbours) is even.

Proof.

Let $G = (V, E)$. Partition V to two subsets:

- $V_{odd} = \{v : \deg(v) \text{ is odd}\}$
- $V_{even} = \{v : \deg(v) \text{ is even}\}$

Clearly, $\sum_{v \in V_{even}} \deg(v)$ is even. By the Handshaking Lemma it follows that:

$$\sum_{v \in V_{odd}} \deg(v) = 2 \cdot |E| - \sum_{v \in V_{even}} \deg(v)$$

is even too.

$\square$

First theorem in Graph Theory

Corollary

In every undirected graph G, the number of vertices with an odd degree (i.e. number of neighbours) is even.

Proof.

Let $G = (V, E)$. Partition V to two subsets:

- $V_{odd} = \{v : \deg(v) \text{ is odd}\}$
- $V_{even} = \{v : \deg(v) \text{ is even}\}$

Clearly, $\sum_{v \in V_{even}} \deg(v)$ is even. By the Handshaking Lemma it follows that:

$$\sum_{v \in V_{odd}} \deg(v) = 2 \cdot |E| - \sum_{v \in V_{even}} \deg(v)$$

is even too.
Now, if we have an odd number of vertices with odd degree, then $\sum_{v \in V_{odd}} \deg(v)$ is odd, a contradiction.

$\square$

First theorem in Graph Theory

Corollary

In every undirected graph G, the number of vertices with an odd degree (i.e. number of neighbours) is even.

Proof.

Let $G = (V, E)$. Partition V to two subsets:

- $V_{odd} = \{v : \deg(v) \text{ is odd}\}$
- $V_{even} = \{v : \deg(v) \text{ is even}\}$

Clearly, $\sum_{v \in V_{even}} \deg(v)$ is even. By the Handshaking Lemma it follows that:

$$\sum_{v \in V_{odd}} \deg(v) = 2 \cdot |E| - \sum_{v \in V_{even}} \deg(v)$$

is even too.

Now, if we have an odd number of vertices with odd degree, then $\sum_{v \in V_{odd}} \deg(v)$ is odd, a contradiction.

Thus there is an even number of vertices with odd degree. $\square$

The most basic graph classes

Some graphs appear so often that they got special names or even special dedicated symbols.

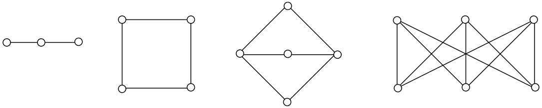


Figure: Special graph classes

The most basic graph classes

Some graphs appear so often that they got special names or even special dedicated symbols.

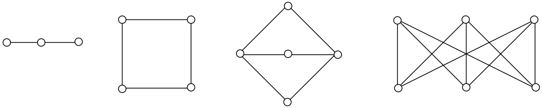


Figure: Special graph classes

The first graph is often denoted by P_3, and in general we define P_n as the path on n vertices, i.e. a graph with vertex set $V = \{v_1, v_2, \ldots, v_n\}$ and edge set $E = \{v_1v_2, v_2v_3, \ldots, v_{n-1}v_n\}$.

So, P_n has ??? edges.

Definition

A path in a graph G is a subgraph of G which is *(isomorphic to)* the graph P_k, for some integer $k \geq 1$. Sometimes a path is also called a simple path.

The most basic graph classes

Some graphs appear so often that they got special names or even special dedicated symbols.

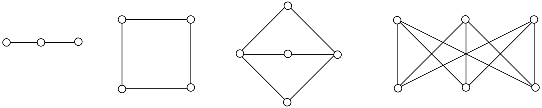


Figure: Special graph classes

The first graph is often denoted by P_3, and in general we define P_n as the path on n vertices, i.e. a graph with vertex set $V = \{v_1, v_2, \ldots, v_n\}$ and edge set $E = \{v_1v_2, v_2v_3, \ldots, v_{n-1}v_n\}$.

So, P_n has $n - 1$ edges.

Definition

A path in a graph G is a subgraph of G which is *(isomorphic to)* the graph P_k, for some integer $k \geq 1$. Sometimes a path is also called a simple path.

The most basic graph classes

Some graphs appear so often that they got special names or even special dedicated symbols.

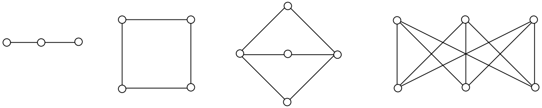


Figure: Special graph classes

The second graph is often denoted by C_4, the cycle on 4 vertices. In general a C_n on n vertices is defined similarly to the P_n, but now with an additional edge between v_n and v_1. So, C_n has ??? edges.

Definition

A cycle in a graph G is a subgraph of G which is *(isomorphic to)* the graph C_k, for some integer $k \geq 3$. Sometimes a cycle is also called a simple circuit.

The most basic graph classes

Some graphs appear so often that they got special names or even special dedicated symbols.

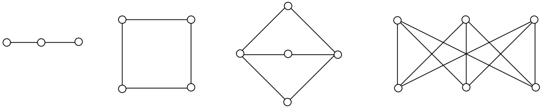


Figure: Special graph classes

The second graph is often denoted by C_4, the cycle on 4 vertices. In general a C_n on n vertices is defined similarly to the P_n, but now with an additional edge between v_n and v_1. So, C_n has n edges.

Definition

A cycle in a graph G is a subgraph of G which is *(isomorphic to)* the graph C_k, for some integer $k \geq 3$. Sometimes a cycle is also called a simple circuit.

The most basic graph classes

Some graphs appear so often that they got special names or even special dedicated symbols.

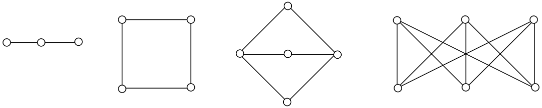


Figure: Special graph classes

The second graph is often denoted by C_4, the cycle on 4 vertices. In general a C_n on n vertices is defined similarly to the P_n, but now with an additional edge between v_n and v_1. So, C_n has n edges.

How many cycles does the third graph have?

Definition

A cycle in a graph G is a subgraph of G which is *(isomorphic to)* the graph C_k, for some integer $k \geq 3$. Sometimes a cycle is also called a simple circuit.

The most basic graph classes

Some graphs appear so often that they got special names or even special dedicated symbols.

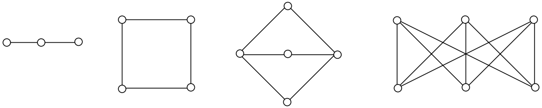


Figure: Special graph classes

All four of these graphs can be described as a $K_{p,q}$: a graph consisting of two disjoint vertex sets on p and on q vertices, and all possible edges between these two vertex sets (and no other edges). So, $K_{p,q}$ has ??? edges.

Definition

$K_{p,q}$ is called a complete bipartite graph. Any subgraph of $K_{p,q}$ is called a bipartite graph.

The most basic graph classes

Some graphs appear so often that they got special names or even special dedicated symbols.

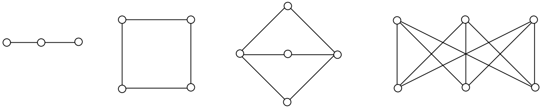


Figure: Special graph classes

All four of these graphs can be described as a $K_{p,q}$: a graph consisting of two disjoint vertex sets on p and on q vertices, and all possible edges between these two vertex sets (and no other edges). So, $K_{p,q}$ has $p \cdot q$ edges.

Definition

$K_{p,q}$ is called a complete bipartite graph. Any subgraph of $K_{p,q}$ is called a bipartite graph.

The most basic graph classes

Some graphs appear so often that they got special names or even special dedicated symbols.

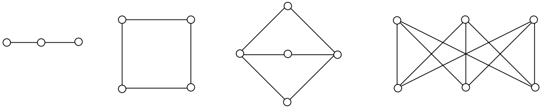


Figure: Special graph classes

All four of these graphs can be described as a $K_{p,q}$: a graph consisting of two disjoint vertex sets on p and on q vertices, and all possible edges between these two vertex sets (and no other edges). So, $K_{p,q}$ has $p \cdot q$ edges.

Definition

$K_{p,q}$ is called a complete bipartite graph. Any subgraph of $K_{p,q}$ is called a bipartite graph.
So a graph is bipartite if and only if we can partition its vertex set to two vertex sets such that every edge has one endpoint in each set.

Bipartite graphs play an eminent role in scheduling and assignment problems.

Some graph classes

Definition

A complete graph on n vertices, denote by K_n, contains all the possible edges between pairs of vertices.

Some graph classes

Definition

A complete graph on n vertices, denote by K_n, contains all the possible edges between pairs of vertices.

How many edges has a K_n?

Some graph classes

Definition

A complete graph on n vertices, denote by K_n, contains all the possible edges between pairs of vertices.

How many edges has a K_n? Answer: $\binom{n}{2} = \frac{1}{2}n(n-1)$.

Some graph classes

Definition

A complete graph on n vertices, denote by K_n, contains all the possible edges between pairs of vertices.

How many edges has a K_n? Answer: $\binom{n}{2} = \frac{1}{2}n(n-1)$.

Definition

The (n-dimensional) hypercube or n-cube Q_n $(n \geq 1)$ is the graph with

$$V = \{\,(e_1, \dots, e_n) \mid e_i \in \{0, 1\}\ (i = 1, \dots, n)\,\},$$

in which two vertices are neighbours if and only if the corresponding rows differ in exactly one entry.

Definition

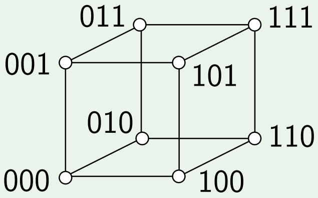
The (n-dimensional) hypercube or n-cube Q_n ($n \geq 1$) is the graph with

$$V = \{ (e_1, \ldots, e_n) \mid e_i \in \{0,1\} \ (i = 1, \ldots, n) \},$$

in which two vertices are neighbours if and only if the corresponding rows differ in exactly one entry.

Examples

$Q_1 = P_2 = K_2$; $Q_2 = C_4$. For $n = 3$ the set V consists of $2^3 = 8$ elements, namely all rows (in short hand notation) 000, 001, 010, 011, 100, 101, 110, 111.



Q_3

More on n-cubes

Theorem

All n-cubes are bipartite.

How to prove this?

More on n-cubes

Theorem

All n-cubes are bipartite.

How to prove this?

Proof.

- We give a bipartition of the vertex set of the n-cube.

More on n-cubes

Theorem

All n-cubes are bipartite.

How to prove this?

Proof.

- We give a bipartition of the vertex set of the n-cube.
- Let V_1 contain all the vertices with an odd number of 1s
- Let V_2 contain all vertices with an even (possibly 0) number of 1s.

More on n-cubes

Theorem

All n-cubes are bipartite.

How to prove this?

Proof.

- We give a bipartition of the vertex set of the n-cube.
- Let V_1 contain all the vertices with an odd number of 1s
- Let V_2 contain all vertices with an even (possibly 0) number of 1s.
- This is clearly a partition of V into two disjoint sets.

More on n-cubes

Theorem

All n-cubes are bipartite.

How to prove this?

Proof.

- We give a bipartition of the vertex set of the n-cube.
- Let V_1 contain all the vertices with an odd number of 1s
- Let V_2 contain all vertices with an even (possibly 0) number of 1s.
- This is clearly a partition of V into two disjoint sets.
- It is easy to see that each edge has one endpoint in each of the sets.

More on n-cubes

Theorem

All n-cubes are bipartite.

How to prove this?

Proof.

- We give a bipartition of the vertex set of the n-cube.
- Let V_1 contain all the vertices with an odd number of 1s
- Let V_2 contain all vertices with an even (possibly 0) number of 1s.
- This is clearly a partition of V into two disjoint sets.
- It is easy to see that each edge has one endpoint in each of the sets.
- So it proves that all n-cubes are bipartite.

□

Exercises

A graph is called k-regular if all of its vertices have degree k.

- **Exercise 1: Which of the graphs P_n, C_n, $K_{p,q}$, K_n, Q_n are k-regular (for some k)?**
- Answer at *PollEV.com/amitabhtrehan005*

Exercises

Q_n is the (n-dimensional) hypercube

- **Exercise 1: Which of the graphs P_n, C_n, $K_{p,q}$, K_n, Q_n are k-regular (for some k)?**
- **Exercise 2:** Find the number of edges in Q_n.
- Answer at *PollEV.com/amitabhtrehan005*

Exercises

A (bi-partite graph is a two-colorable graoh!

- **Exercise 1: Which of the graphs P_n, C_n, $K_{p,q}$, K_n, Q_n are k-regular (for some k)?**
- **Exercise 2:** Find the number of edges in Q_n.
- **Exercise 3:** Which of the graphs P_n, C_n, K_n are bipartite?
- Answer at *PollEV.com/amitabhtrehan005*