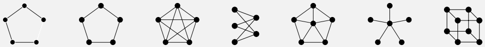
Families of Graphs

- **Path graph** P_n: The path graph P_n has $n+1$ vertices, $V = \{v_0, v_1, \ldots, v_n\}$ and n edges, $E = \{v_0v_1, v_1v_2, \ldots, v_{n-1}v_n\}$.
  - ★ The **length** of a path is the number of *edges* in the path.

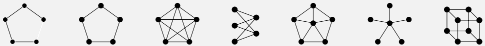
Families of Graphs



- **Path graph** P_n: The path graph P_n has $n+1$ vertices, $V = \{v_0, v_1, \ldots, v_n\}$ and n edges, $E = \{v_0v_1, v_1v_2, \ldots, v_{n-1}v_n\}$.
  - ★ The **length** of a path is the number of *edges* in the path.

- **Cycle graph** C_n: The cycle graph C_n has n vertices, $V = \{v_1, \ldots, v_n\}$ and n edges, $E = \{v_1v_2, v_2v_3, \ldots, v_{n-1}v_n, v_nv_1\}$.

Families of Graphs



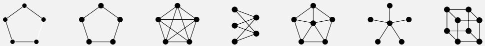
- **Path graph P_n**: The path graph P_n has $n+1$ vertices, $V = \{v_0, v_1, \ldots, v_n\}$ and n edges, $E = \{v_0 v_1, v_1 v_2, \ldots, v_{n-1} v_n\}$.
  - ★ The **length** of a path is the number of *edges* in the path.

- **Cycle graph C_n**: The cycle graph C_n has n vertices, $V = \{v_1, \ldots, v_n\}$ and n edges, $E = \{v_1 v_2, v_2 v_3, \ldots, v_{n-1} v_n, v_n v_1\}$.

We often try to find and/or count paths and cycles in a graph.

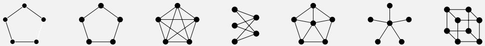
Question. What is the smallest path? Smallest cycle?

Families of Graphs



- **Complete graph K_n**: The complete graph K_n has n vertices, $V = \{v_1, \ldots, v_n\}$ and has an edge connecting every pair of distinct vertices, for a total of ____________ edges.

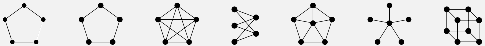
Families of Graphs



- **Complete graph** *K_n*: The complete graph K_n has n vertices, $V = \{v_1, \ldots, v_n\}$ and has an edge connecting every pair of distinct vertices, for a total of ____________ edges.

Definition. A **bipartite** graph is a graph where the vertex set can be broken into two parts such that there are no edges between vertices in the same part.

Families of Graphs

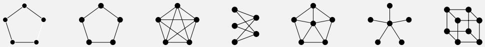


- **Complete graph** K_n: The complete graph K_n has n vertices, $V = \{v_1, \ldots, v_n\}$ and has an edge connecting every pair of distinct vertices, for a total of ____________ edges.

Definition. A **bipartite** graph is a graph where the vertex set can be broken into two parts such that there are no edges between vertices in the same part.

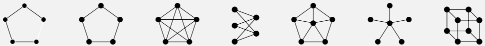
- **Complete bipartite graph** $K_{m,n}$: The complete bipartite graph $K_{m,n}$ has $m + n$ vertices $V = \{v_1, \ldots, v_m, w_1, \ldots, w_n\}$ and an edge connecting each v vertex to each w vertex.

Families of Graphs



- **Wheel graph W_n**: The wheel graph W_n has $n+1$ vertices $V = \{v_0, v_1, \ldots, v_n\}$. Arrange and connect the last n vertices in a cycle (the rim of the wheel). Place v_0 in the center (the hub), and connect it to every other vertex.

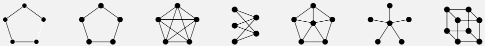
Families of Graphs



- **Wheel graph W_n**: The wheel graph W_n has $n+1$ vertices $V = \{v_0, v_1, \ldots, v_n\}$. Arrange and connect the last n vertices in a cycle (the rim of the wheel). Place v_0 in the center (the hub), and connect it to every other vertex.

- **Star graph St_n**: The star graph St_n has $n+1$ vertices $V = \{v_0, v_1, \ldots, v_n\}$ and n edges $E = \{v_0v_1, v_0v_2, \ldots, v_0v_n\}$.

Families of Graphs



- **Wheel graph W_n**: The wheel graph W_n has $n+1$ vertices $V = \{v_0, v_1, \ldots, v_n\}$. Arrange and connect the last n vertices in a cycle (the rim of the wheel). Place v_0 in the center (the hub), and connect it to every other vertex.

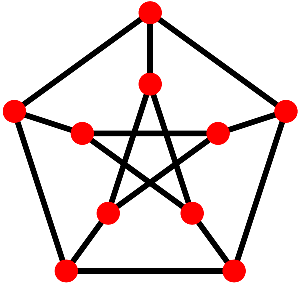
- **Star graph St_n**: The star graph St_n has $n+1$ vertices $V = \{v_0, v_1, \ldots, v_n\}$ and n edges $E = \{v_0 v_1, v_0 v_2, \ldots, v_0 v_n\}$.

- **Cube graph $\square_n$**: The cube graph in n dimensions, $\square_n$, has 2^n vertices. We index the vertices by binary numbers of length n. Two vertices are adjacent when their binary numbers differ by exactly one digit.

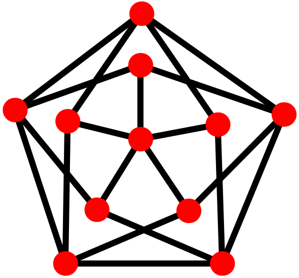
Special Graphs

Two graphs we will see on a consistant basis are:

Petersen graph P



Grötzsch graph Gr



Special Graphs

Definition. The **platonic solids** are the tetrahedron, cube, octahedron, icosahedron, and dodecahedron. They are the only regular convex polyhedra made of regular polygons.

Special Graphs

Definition. The **platonic solids** are the tetrahedron, cube, octahedron, icosahedron, and dodecahedron. They are the only regular convex polyhedra made of regular polygons.

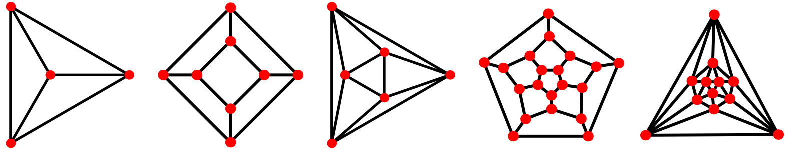
Definition. The **Schlegel diagram** of a polyhedron is a planar 2D graph that represents a 3D object, where vertices of the graph represent vertices of the polyhedron, and edges of the graph represent the edges of the polyhedron.

Special Graphs

Definition. The **platonic solids** are the tetrahedron, cube, octahedron, icosahedron, and dodecahedron. They are the only regular convex polyhedra made of regular polygons.

Definition. The **Schlegel diagram** of a polyhedron is a planar 2D graph that represents a 3D object, where vertices of the graph represent vertices of the polyhedron, and edges of the graph represent the edges of the polyhedron.

- The **Platonic graphs** are the Schlegel diagrams of the five platonic solids.



When are two graphs the same?

When are two graphs the same?

Two graphs G_1 and G_2 are **equal** ($G_1 = G_2$) if they have the exact same vertex sets and edge sets.

When are two graphs the same?

Two graphs G_1 and G_2 are **equal** ($G_1 = G_2$) if they have the exact same vertex sets and edge sets.

The graphs G_1 and G_2 are **isomorphic** ($G_1 \approx G_2$) if there exists a *bijection* on the vertex sets, $\varphi : V(G_1) \to V(G_2)$ such that

$$v_i v_j \text{ is an edge of } G_1 \quad \text{iff} \quad \varphi(v_i)\varphi(v_j) \text{ is an edge of } G_2.$$

When are two graphs the same?

Two graphs G_1 and G_2 are **equal** ($G_1 = G_2$) if they have the exact same vertex sets and edge sets.

The graphs G_1 and G_2 are **isomorphic** ($G_1 \approx G_2$) if there exists a *bijection* on the vertex sets, $\varphi : V(G_1) \to V(G_2)$ such that

$$v_i v_j \text{ is an edge of } G_1 \quad \text{iff} \quad \varphi(v_i)\varphi(v_j) \text{ is an edge of } G_2.$$

In this course, we will spend a large amount of time trying to figure out whether two given graphs are the same.

When are two graphs the same?

Two graphs G_1 and G_2 are **equal** ($G_1 = G_2$) if they have the exact same vertex sets and edge sets.

The graphs G_1 and G_2 are **isomorphic** ($G_1 \approx G_2$) if there exists a *bijection* on the vertex sets, $\varphi : V(G_1) \to V(G_2)$ such that

$$v_i v_j \text{ is an edge of } G_1 \quad \text{iff} \quad \varphi(v_i)\varphi(v_j) \text{ is an edge of } G_2.$$

In this course, we will spend a large amount of time trying to figure out whether two given graphs are the same.

Side note: The set of homomorphisms of a graph (isomorphisms into itself) is a measure of its symmetry. *Example.* 

Simple operations on graphs

The **union** of two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ can mean two different things:

Simple operations on graphs

The **union** of two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ can mean two different things:

- When the vertex sets are different, the **(disjoint) union** H of G_1 and G_2 is formed by placing the graphs side by side. In this case, $H = (V_1 \cup V_2, E_1 \cup E_2)$.

Simple operations on graphs

The **union** of two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ can mean two different things:

- When the vertex sets are different, the **(disjoint) union** H of G_1 and G_2 is formed by placing the graphs side by side. In this case, $H = (V_1 \cup V_2, E_1 \cup E_2)$.
- When the vertex sets are the same, then the **(edge) union** H of G_1 and G_2 contains every edge of both E_1 and E_2. In this case, $H = (V, E_1 \cup E_2)$.

Simple operations on graphs

The **union** of two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ can mean two different things:

- When the vertex sets are different, the **(disjoint) union** H of G_1 and G_2 is formed by placing the graphs side by side. In this case, $H = (V_1 \cup V_2, E_1 \cup E_2)$.
- When the vertex sets are the same, then the **(edge) union** H of G_1 and G_2 contains every edge of both E_1 and E_2. In this case, $H = (V, E_1 \cup E_2)$.

The **complement** G^c or $\overline{G}$ of a graph $G = (V, E)$ is a graph with vertex set V and whose edge set contains all edges **NOT** in G.

Simple operations on graphs

The **union** of two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ can mean two different things:

- When the vertex sets are different, the **(disjoint) union** H of G_1 and G_2 is formed by placing the graphs side by side. In this case, $H = (V_1 \cup V_2, E_1 \cup E_2)$.
- When the vertex sets are the same, then the **(edge) union** H of G_1 and G_2 contains every edge of both E_1 and E_2. In this case, $H = (V, E_1 \cup E_2)$.

The **complement** G^c or $\overline{G}$ of a graph $G = (V, E)$ is a graph with vertex set V and whose edge set contains all edges **NOT** in G.

Consequence: Suppose $G = (V, E_1)$ and $G^c = (V, E_2)$. Then $E_1 \cap E_2 = \emptyset$ and $E_1 \cup E_2 = E(K_{|V|})$. (Recall K_n: complete graph.)

Subgraphs

A **subgraph** H of a graph G is a graph where every vertex of H is a vertex of G, and where every edge of H is an edge of G.

Subgraphs

A **subgraph** H of a graph G is a graph where every vertex of H is a vertex of G, and where every edge of H is an edge of G.

★ If edge e of G is in H, then the endpoints of e must also be in H.

Subgraphs

A **subgraph** H of a graph G is a graph where every vertex of H is a vertex of G, and where every edge of H is an edge of G.
★ If edge e of G is in H, then the endpoints of e must also be in H.

A subgraph H is a **proper subgraph** if $H \neq G$.

Subgraphs

A **subgraph** H of a graph G is a graph where every vertex of H is a vertex of G, and where every edge of H is an edge of G.
★ If edge e of G is in H, then the endpoints of e must also be in H.

A subgraph H is a **proper subgraph** if $H \neq G$.

If G_1 and G_2 are two graphs, we say that G_1 **contains** G_2 if there exists a subgraph H of G_1 such that H is isomorphic to G_2.

Subgraphs

A **subgraph** H of a graph G is a graph where every vertex of H is a vertex of G, and where every edge of H is an edge of G.
★ If edge e of G is in H, then the endpoints of e must also be in H.

A subgraph H is a **proper subgraph** if $H \neq G$.

If G_1 and G_2 are two graphs, we say that G_1 **contains** G_2 if there exists a subgraph H of G_1 such that H is isomorphic to G_2.

Example. Show that the wheel W_6 contains a cycle of length 3, 4, 5, 6, and 7.

Induced Subgraphs

For a graph $G = (V, E)$ and any subset $W \subseteq V(G)$, we can define the subgraph of G **induced by** W.

Induced Subgraphs

For a graph $G = (V, E)$ and any subset $W \subseteq V(G)$,
we can define the subgraph of G **induced by** W.

Define H:

- $V(H) = W$
- $E(H) =$ edges in $E(G)$ that have endpoints *exclusively in* W.

Induced Subgraphs

For a graph $G = (V, E)$ and any subset $W \subseteq V(G)$,
we can define the subgraph of G **induced by** W.

Define H:

- $V(H) = W$
- $E(H) =$ edges in $E(G)$ that have endpoints *exclusively in* W.

Any graph that could be defined in this way is called an **induced subgraph** of G.

Induced Subgraphs

For a graph $G = (V, E)$ and any subset $W \subseteq V(G)$, we can define the subgraph of G **induced by** W.

Define H:

- $V(H) = W$
- $E(H) =$ edges in $E(G)$ that have endpoints *exclusively in* W.

Any graph that could be defined in this way is called an **induced subgraph** of G.

Induced subgraphs of G are always subgraphs of G, but not vice versa.