CSC 225 - Summer 2019

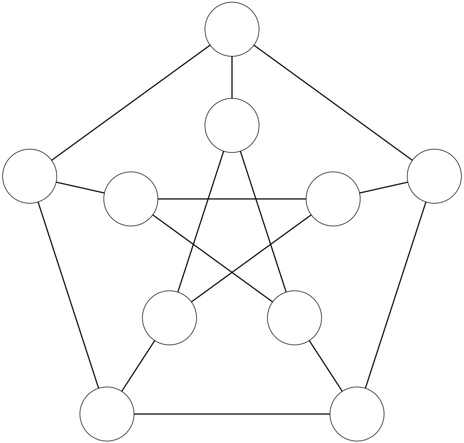
Graphs I

Bill Bird

Department of Computer Science
University of Victoria

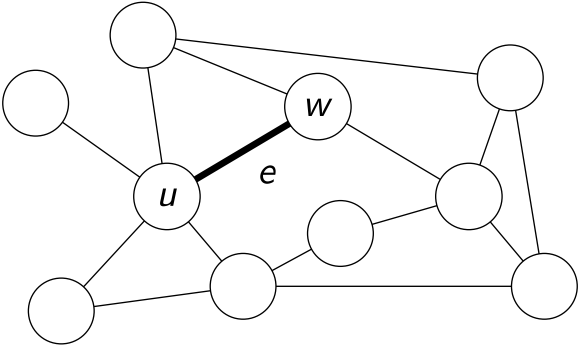
July 8, 2019

Graphs (1)



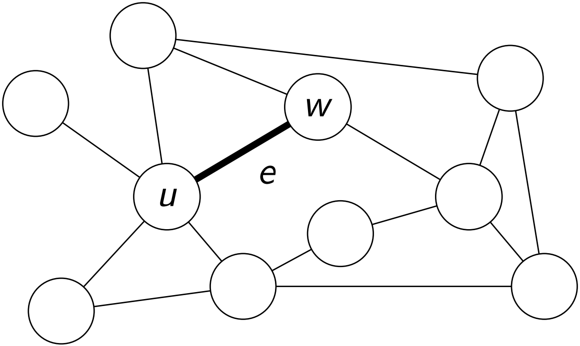
- A **graph** is a collection of **vertices** and **edges**.
- Formally, a graph G is an ordered pair (V, E), where V is a set of vertices and E is a set of edges.

Graphs (2)



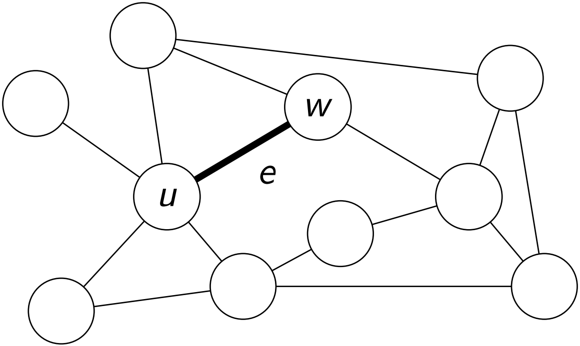
- In the graph G above, u and w are vertices (that is, $u, w \in V(G)$).
- Similarly, $e \in E(G)$ is an edge of G.

Graphs (3)



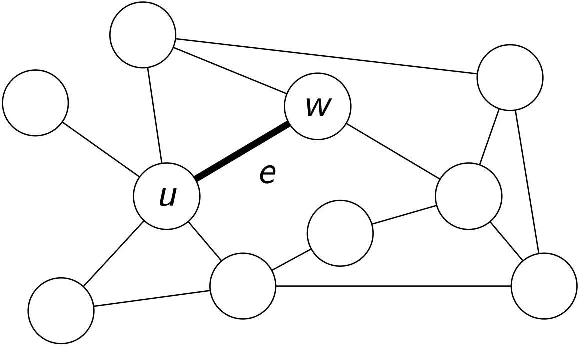
- The graph above is **undirected**: Edges join pairs of vertices, but have no directionality.
- In undirected graphs, edges correspond to unordered pairs of vertices.
- The edge e corresponds to the pair $\{u, w\}$.

Graphs (4)



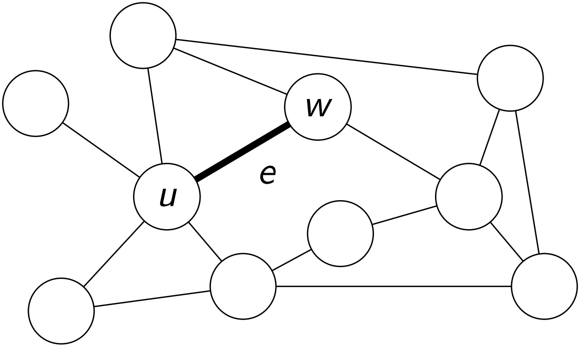
- Usually, however, the set notation is omitted, so we write $e = uw$.
- We say that u and w are the **endpoints** of e, and that e is **incident to** u and w.

Graphs (5)



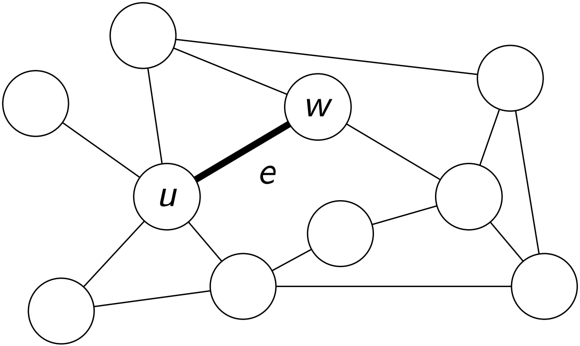
- Normally, the number of vertices in a graph is denoted by n and the number of edges is denoted by m. Sometimes, these names are used without being explicitly defined.

Graphs (6)



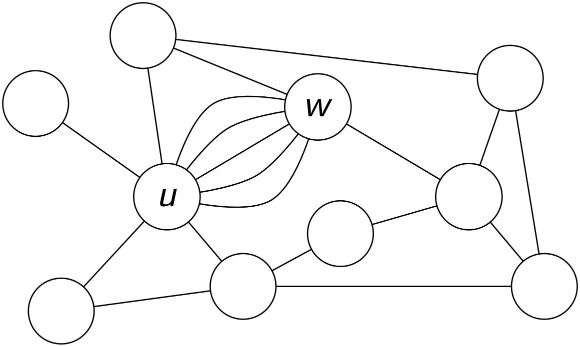
- Two vertices which are joined by an edge are said to be **neighbours** or **adjacent**.

Graphs (7)



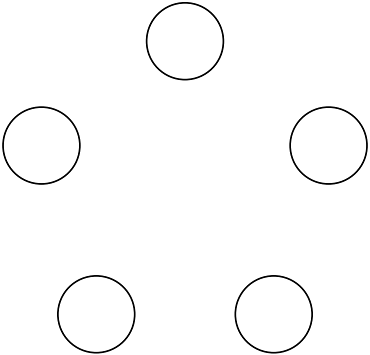
- The **degree** of a vertex, denoted by $\deg(v)$, is the number of edges incident to v.
- In the graph above, $\deg(u) = 5$ and $\deg(w) = 3$.

Graphs (8)



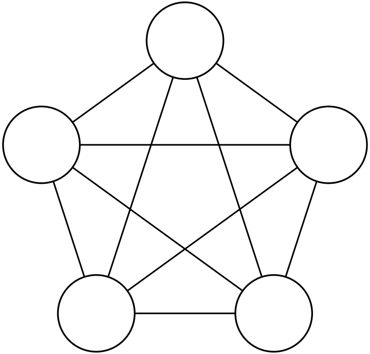
- In some contexts, multiple edges between the same pair of vertices are permitted.
- Such graphs are called **multigraphs**.
- In this course, unless otherwise stated, all graphs will be **simple graphs**, where multiple edges are not permitted.

Complete Graphs (1)



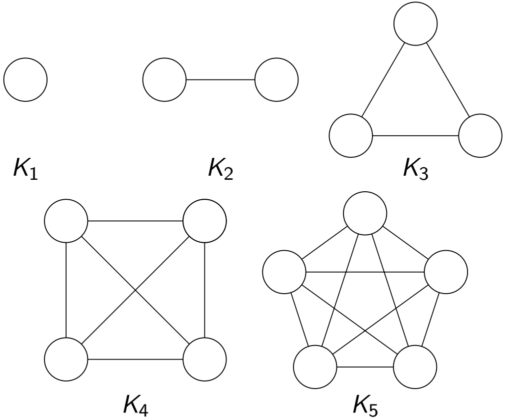
- **Question**: In a graph with n vertices, what is the maximum number of edges m?
- Clearly, the minimum number of edges is 0.

Complete Graphs (2)



- There can be at most 1 edge between any pair of vertices.
- A graph on n vertices with an edge between every pair of vertices is called a **complete graph** and denoted K_n.
- The graph above is K_5.

Complete Graphs (3)



- **Exercise**: Prove that the number of edges in K_n is

$$\binom{n}{2} = \frac{n(n-1)}{2}$$

The Handshake Lemma (1)

Theorem: (The Handshake Lemma)
In a graph G with n vertices and m edges,

$$\sum_{v \in V(G)} \deg(v) = 2m$$

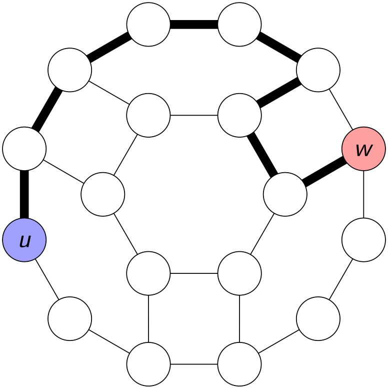
The Handshake Lemma (2)

Theorem: (The Handshake Lemma)
In a graph G with n vertices and m edges,

$$\sum_{v \in V(G)} \deg(v) = 2m$$

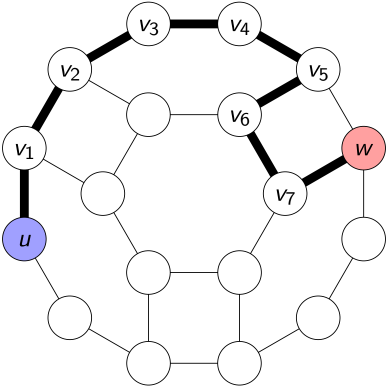
Proof Outline: The theorem can be proven by induction on n or with a counting argument. The counting approach is fairly simple: Adding up the degree of every vertex will count each edge twice (once for each endpoint).

Graph Properties (1)



- For two vertices $u, w \in V(G)$, a **path** between u and w (or a uw-path) is a sequence of vertices (without repetition) which connects u to w.
- If a uw-path exists, we say that w is **reachable** from u (and vice versa).

Graph Properties (2)

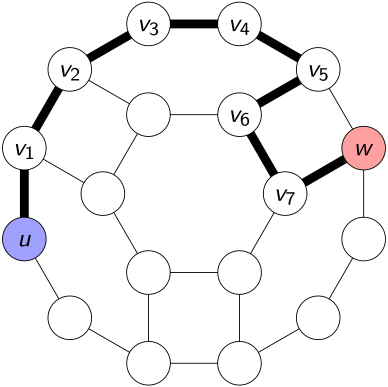


- Formally, a *uw*-path of length k is a sequence

$$u = v_0, v_1, v_2, \ldots, v_{k-1}, v_k = w$$

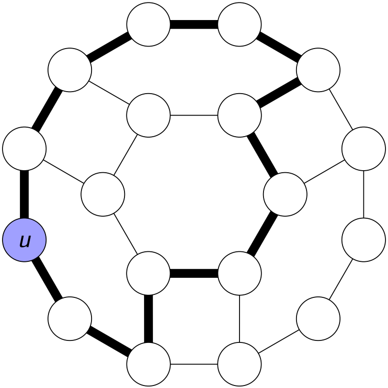
such that no vertex appears twice and for $0 \leq i < k$, there is an edge between v_i and v_{i+1}.

Graph Properties (3)



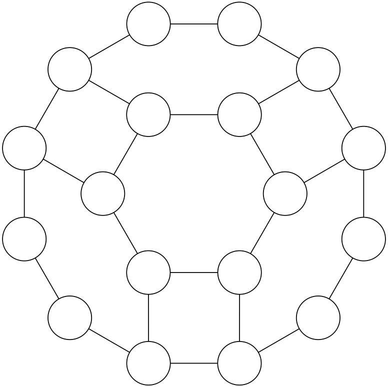
- Note that the length of a path is the number of edges in the path (so a single edge is a path of length 1).
- The path above has length 8.

Graph Properties (4)



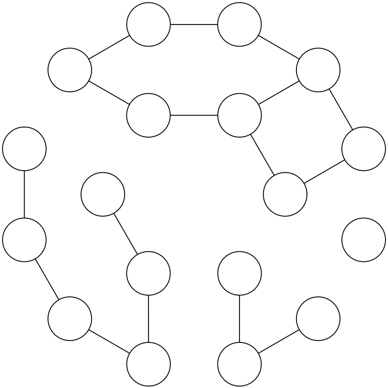
- A **cycle** is a path of length at least 1 which begins and ends at the same vertex.

Graph Properties (5)



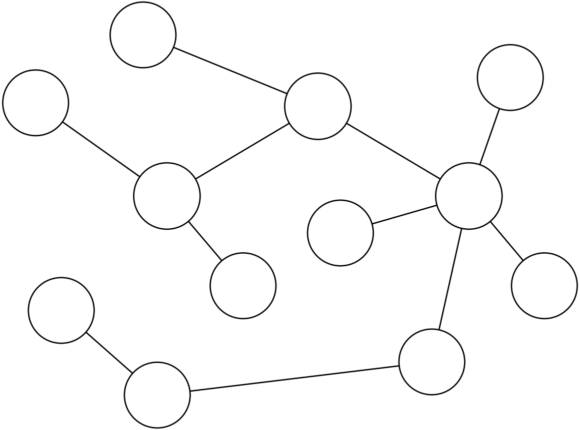
- A graph G is **connected** if, for every pair of vertices u and w, there exists at least one uw-path.
- The graph above is connected.

Graph Properties (6)



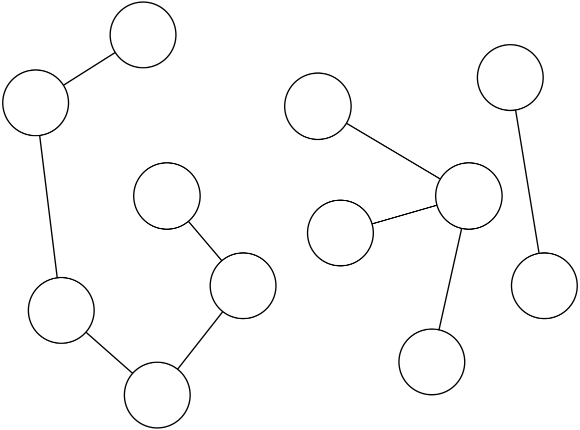
- The graph above is not connected.
- A disconnected graph consists of at least two **connected components** (the above graph has four).

Trees (1)



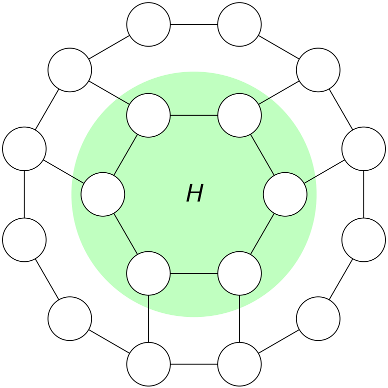
- A **tree** is a connected, acyclic graph.
- **Exercise**: Prove that a tree on n vertices has exactly $n-1$ edges.

Trees (2)



- A graph consisting of one or more disjoint trees is called a **forest**.

Subgraphs (1)

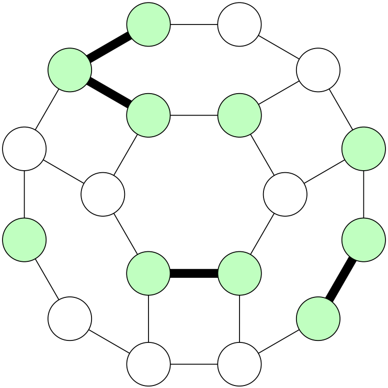


- A *subgraph* of a graph G is a graph H such that

$$V(H) \subseteq V(G) \quad \text{and} \quad E(H) \subseteq E(G)$$

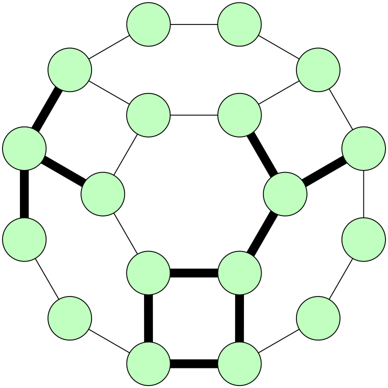
- The shaded region of the graph above encloses a subgraph H.

Subgraphs (2)



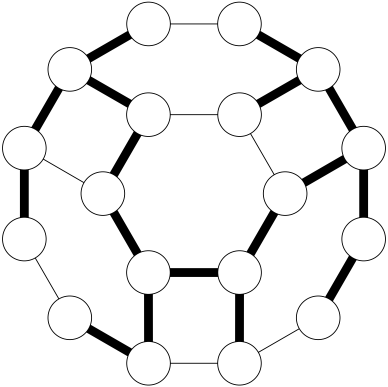
- The highlighted vertices and edges above also form a subgraph.
- Subgraphs may be disconnected, and may not contain all edges between the corresponding vertices in the original graph.

Subgraphs (3)



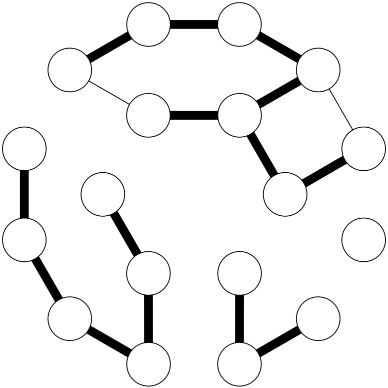
- A **spanning subgraph** is a subgraph which contains all vertices of the original graph.

Subgraphs (4)



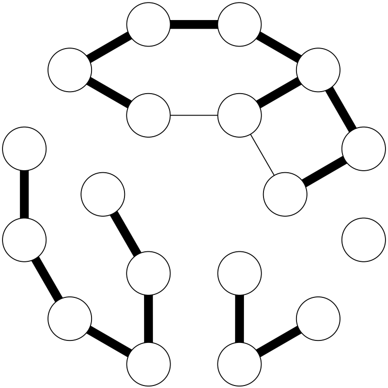
- A **spanning tree** is a spanning subgraph which is also a tree.
- Spanning trees can only exist in connected graphs.

Subgraphs (5)



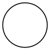
- A **spanning forest** is a subgraph consisting of a spanning tree for each connected component.

Subgraphs (6)



- In general, spanning trees and spanning forests are not unique.
- If the graph is a tree, it has exactly one spanning tree.
- **Exercise**: Prove that a connected graph G which is not a tree has at least two spanning trees.

Hypercubes (1)



Q_0

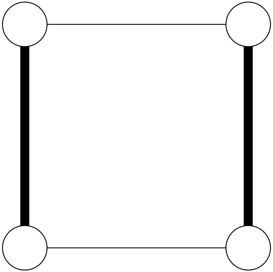
- The **hypercubes** are a family of graphs which can be defined inductively.
- Define the **hypercube of order 0**, denoted Q_0, to be a single vertex.

Hypercubes (2)

Q_1

- To construct Q_n for any $n \geq 1$, create two copies of Q_{n-1} and add edges between each vertex and its copy.

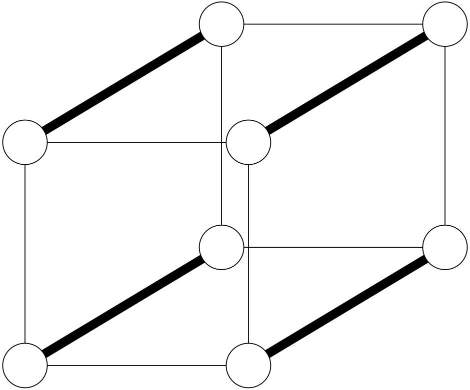
Hypercubes (3)



Q_2

- Each graph Q_n contains all previous hypercubes ($Q_0, Q_1, \ldots, Q_{n-1}$) as subgraphs.

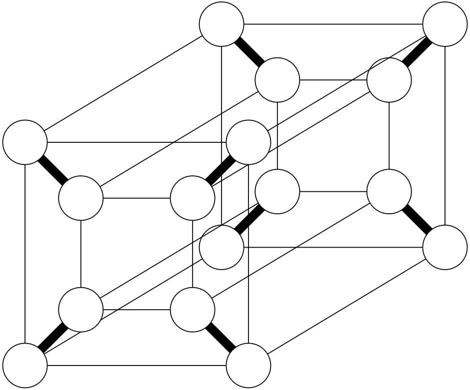
Hypercubes (4)



Q_3

- The number of vertices in Q_n doubles at each step.
- **Easy exercise**: Prove that $|V(Q_n)| = 2^n$ for $n \geq 0$.

Hypercubes (5)



Q_4

- The number of edges is not as easy to characterize.
- **Exercise**: Find a closed form for the number of edges in a hypercube of order n.

Hypercubes (6)

The hypercube of order 0, Q_0, has 0 edges.

Hypercubes of order $n \geq 1$ contain two copies of Q_{n-1} (and all edges therein) plus one edge for each vertex in Q_{n-1} (which connects the two copies of that vertex). The number of vertices in Q_{n-1} is 2^{n-1}.

Therefore, the number of edges can be counted with the recurrence

$$
\begin{aligned}
h(n) &= 0 && \text{if } n = 0 \\
&= 2h(n-1) + 2^{n-1} && \text{if } n \geq 1
\end{aligned}
$$

Hypercubes (7)

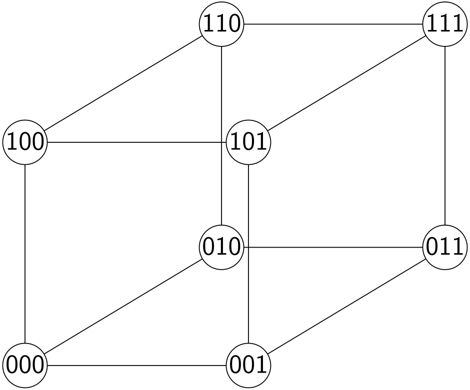
$$
\begin{aligned}
h(n) &= 0 && \text{if } n = 0 \\
&= 2h(n-1) + 2^{n-1} && \text{if } n \geq 1
\end{aligned}
$$

Solving the recurrence produces the closed form

$$h(n) = n2^{n-1}$$

which can be proven correct by induction.

Hypercubes (8)



Q_3

- The hypercube Q_n can also be constructed by creating a vertex for each binary sequence of length n and adding edges between two vertices whose sequences differ in exactly one position.