CS130 - Graphs Summary

Definition

A set of vertices connected by edges, which can be formally defined as follows:

$G = (V, E)$, where:

- V is the set of vertices
- E is the collection of 2-tuples, denoting the start & end vertices of an edge (it is a collection, not a set, as multiplicity can be useful). $E \subseteq V \times V$, since edges connect vertices

Nomenclature

- Directed graphs are when edges between vertices only connect one way. In this case, the elements of E are <u>ordered</u>, with the first being the start vertex, and the last being the end vertex

Also called digraphs

- Undirected graphs are when edges between vertices connect both ways. In this case, the elements of E are <u>unordered</u>
- Source / initial vertex denotes the start of an edge
- Destination / terminal vertex denotes the end of an edge.
- Loops ~~[illegible]~~ denote an edge with the same initial and terminal node, $(u, u) \in E$ ($\neq$ cycles).

- Parallel edges denote two edges connecting the same terminal & initial vertices (why collections not sets are used in E).

- Adjacent vertices / neighbours denotes two vertices being connected by a single edge.

- Incident denotes an edge connecting a vertex

- Endpoints denotes the vertices an edge connects

- Degree denotes the number of edges a vertex has.
  - ↳ In undirected graphs, loops are counted twice.
  - ↳ In directed graphs, incoming and outgoing edges are differentiated

<u>Edges as relations</u>

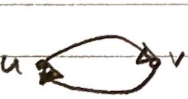
Consider the graph:

$$G = (V, E) \ \& \ E \subseteq V \times V.$$

If $(u,v) \in E$ <u>and</u> $(v,u) \in E$ $\forall u, v \in V$ then the relation is symmetric.

~~this means that it must be undirected~~

$(u,v) \in E$
$(v,u) \in E$



↳ All undirected graphs must be ~~undirected~~ Symmetric, but directed graphs can also be symmetric.

$(u,u) \notin E$

If $(u,u) \notin E \ \forall u$, then the relation is irreflexive
↳ This means that the graph cannot contain any loops.

"Simple" graphs are undirected, and have neither loops nor parallel edges.

Vertex labels & Edge labels

It can be convenient to label the vertices. This can be done trivial by defining a function that maps the set of vertices onto their labels

$$f : V \to L.$$

Edge labels can be labelled in a similar way, using function, however, they can be used to assign 'weights' on graph edges, which is a useful property.

Graph isomorphism

Graph isomorphism is a property of two graphs that are topologically the same, formally:

$\exists$ bijections $f : V_1 \to V_2$ & $g : E_1 \to E_2$
Such that $g((u, v)) = (f(u), g(u))$
$\forall (u, v) \in E$.

This is written as $G_1 \cong G_2$.

Graph classes

- Empty graph; no edges, possible vertices

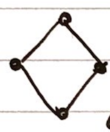
$E = \emptyset$

- complete graphs; every vertex is adjacent to every other vertex.

K_5

↳ Denoted by K_n for n being the number of vertices.

- cyclic graphs: all the vertices form a single complete cycle



C_4

↳ Denoted by C_n for n being the number of vertices.

- Bi-partite graphs: A graph divided into two sections, where vertices don't connect to vertices in their own section.

↳ Denoted by $K_{m,n}$ for <u>complete</u> bi-partite graphs, where every node in one section connects to every node in the other.

More formally, partitioned into two disjoint sets, in which vertices in one set <u>only</u> relate to vertices in the other.

$$V = V_1 \cup V_2 \ \& \ V_1 \cap V_2 = \emptyset$$

$$E \subseteq (V_1 \times V_2) \cup (V_2 \times V_1).$$

- Planar graphs: when a graph can be drawn on a 2D plane without intersections

Walks, paths & tours

- A <u>walk</u> in the graph $G = (V, E)$ is a finite sequence of the form:

$$v_0, (v_0, v_1), v_1, (v_1, v_2), \dots, (v_{n-1}, v_n), v_n.$$

where $v_i \in V \ \forall i$ and $(v_{i-1}, v_i) \in E \ \forall i \geq 1$.

which can also be written as:

$$v_0 \to v_1 \to \dots \to v_n \quad \text{or} \quad v_0 \xrightarrow{*} v_n.$$

Simple path ⊆ path

- A <u>path</u> is a walk where no edges are repeated
- A <u>Simple path</u> is a walk where no vertices are repeated
- A ~~[illegible]~~ <u>tour</u> is a walk where the start and end vertices are the same.
- A <u>cycle</u> is a tour where no edges are repeated
- A <u>Simple cycle</u> is a tour where no vertices are repeated

- A vertex is <u>reachable</u> from another vertex if there exists a path between them.

Graph Colouring

~~The~~ A <u>colouring</u> of a graph G with colours C is a function $f : V \to C$

A <u>proper colouring</u> is when adjacent vertices have different colours:

$$\forall (u, v) \in E : f(u) \neq f(v).$$

A graph is bipartite if it is 2-colourable

Eulerian cycles

An Eulerian ~~cycle~~ path is when a path traverses every edge in the graph exactly once.

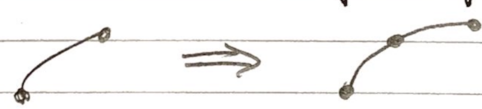
An Eulerian cycle is an Eulerian path with the same start and end vertex.

An undirected graph contains an Eulerian cycle iff every edge has an even degree

Planar graphs

A planar graph is a graph that can be embedded into a 2D plane, and drawn such that no edges intersect

A subdivision of a graph is the process of adding a vertex in the middle of an edge



If a graph is initially non-planar, subdivision will never make it planar.

Subgraphs are another graph composed of vertices and edges in a given graph

~~Subgraphs are graphs undoing the process of subdivision~~

All non-planar graphs ~~[illegible]~~ contain a subgraph that is isomorphic to either K_5 or $K_{3,3}$