2.1 Graph Properties and Graph Classes

Most all problems we'll consider in this course relate to **graph properties** and **graph classes**. A graph property is simply a descriptive characteristic that *some graph* holds. E.g., consider the properties below for clique graph K_3:

- Property: K_3 is *non-empty* (has at least one edge)
- Property: K_3 is *connected*
- Property: $|V(K_3)| = |E(K_3)| = 3$
- Property: K_3 contains a cycle of length three : $C_3 \subseteq K_3$
- Property: $\forall v \in V(K_3) : d(v) = 2$

Conversely, a graph class is a collection of *all possible* graphs that all have some shared property or properties. We already talked about several graph classes. One such example would be the class of **simple graphs**, which contains all possible graphs that have the properties of no multi-edges and no self loops. Other examples we've covered include **connected graphs**, **acyclic graphs**, or **clique graphs**, among many others. As one could infer, the cardinality of a given graph class could be infinite.

Most of the proofs in this course will relate to proving relationships between graph classes and graph properties. E.g., prove that graph G with property P belongs to a more specific graph class C. Similarly, we might want to prove that G with property P also must have properties Q and R. Some of the time, what we might be trying to determine is whether two graph classes fully intersect, partially intersect, have some subset/superset relation, or are fully disjoint. We will discuss this in more detail when we start getting into proof techniques.

2.2 Isomorphism

Two graphs: $G = (V, E)$ and $G' = (V', E')$ are called **isomorphic** if there is a one-to-one mapping f from V onto V' such that any two vertices $v_i, v_j \in V$ are adjacent iff $f(v_i)$ and $f(v_j)$ are adjacent. We would say that G is isomorphic to G', or $G \cong G'$. This mapping is also referred to as an edge-preserving **bijection** from V to V'. Generally, a bijection between two arbitrary sets X,Y can be considered a functional mapping $f : X \rightarrow Y$ that has the following properties:

1. One-to-one: $\forall x \in X : \exists$ exactly one $f(x) = y \in Y$

2. Invertible: $\forall y \in Y : \exists$ exactly one $g(y) = x \in X$, where g is the inverse function of f

Isomorphism can also be extended to non-simple graphs, through adding the language that there also must explicitly also exist a bijection from E to E', where every edge $v_i, v_j \in E(G)$ must map to an edge $(f(v_i), f(v_j) \in E(G'))$.

By permuting the rows of the adjacency matrix of G (A), we should be able to create the adjacency matrix of G' (A'); i.e., there exists a **permutation matrix** P such that $PAP^T = A'$.

If $G(V, E)$ and $G'(V', E')$ are isomorphic, then we can make general statements such as:

1. $|V| = |V'|$ and $|E| = |E'|$
2. the **degree sequences** of G and G' sorted in non-increasing order are identical
3. the lengths of the longest shortest paths (a graph's **diameter**) in G and G' are equal
4. the lengths of the shortest cycles (a graph's **girth**) in G and in G' are equal

Note that properties (1) - (4) are necessary but not sufficient conditions for isomorphism. We'll discuss more about what this means soon.

The **isomorphism relation** on the set of ordered pairs from G to G' is:
reflexive: $G \cong G$
symmetric: if $G \cong H$, then $H \cong G$
transitive: if $G \cong H$ and $H \cong J$, then $G \cong J$

An **isomorphism class** is an equivalence class of graphs that are all under an isomorphic relation.

An **automorphism** is an isomorphism from G to itself. The set of automorphisms of G is known as G's **automorphism group**. This can be loosely thought of the ways in which a graph is *symmetric*. While the notion of isomorphism and automorphism appear quite similar on the surface, an automorphic permutation of G will be **equal** to the original graph (i.e., the *edge list* is preserved).

2.2.1 Subgraph Isomorphism

Sometimes we may talk about the **subgraph isomorphism** problem, which is: Given a graph G and a graph H of equal or smaller size of G, does there exist a subgraph of G that is isomorphic to H? Subgraph isomorphism and related problems (**subgraph counting**: how many different subgraphs of G are isomorphic to H? **subgraph enumeration**: what are those subgraphs of G that are isomorphic to H?) are common techniques of graph mining.

We also want to differentiate between **vertex-induced** (or simply, **induced**) subgraphs and **non-induced** subgraphs. For both, we consider a subgraph S of G that contains a set of vertices $V(S) \in V(G)$. We would say that the subgraph is induced if $\forall (u, v) \in E(G)$ s.t. $u, v \in V(S) \implies (u, v) \in E(S)$. The subgraph is non-induced if it only contains a subset of the edges $(u, v) \in E(G)$ s.t. $u, v \in V(S)$. In the future, we might ask if some vertex subset $V(S) \subset V(G)$ *induces a subgraph* with specific characteristics in G - inducing the subgraph considers all existing edges among the vertices in $V(S)$.

2.2.2 Computational Considerations

In terms of computational complexity, graph isomorphism is thought to be solvable in quasi-polynomial time [$exp((\log n)^{O(1)})$, Babai 2015], though it remains an open problem. Subgraph isomorphism is NP-complete. A naive algorithm for subgraph isomorphism would involve exhaustively checking the local neighborhood of all $v \in V(G)$ for an isomorphic relation to H, and requires $O(n^k)$ time, where $n = |V(G)|$ and $k = |V(H)|$. Although, several specialized algorithms exists; e.g., triangles can be enumerated in $O(m^{\frac{2\omega}{\omega+1}})$ time, cycles can be found in $O(n^\omega \log n)$ time, and trees can be found in polynomial time (ω is the exponent of fast matrix multiplication – most recently, $\omega \approx 2.373$ by Alman and Williams, 2021).