

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

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).

### 2.1.1 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  $V(S) \subseteq V(G)$ . 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.  $u, v \in V(S)$ .

### 2.1.2 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 exist; 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 ( $\omega$  is the exponent of fast matrix multiplication – most recently,  $\omega \approx 2.373$  by Alman and Williams, 2021).