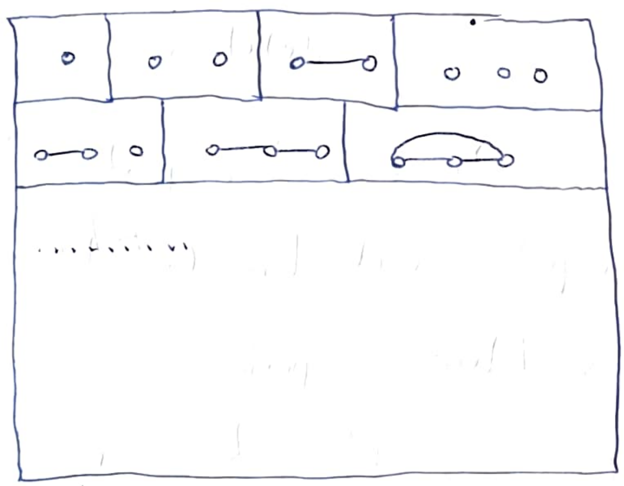
Quick Recap: We have $\boxed{\text{proved}}$ two theorems so far:

Theorem 1: For any graph G, the reachability reln is an equivalence relation on its vertex set $V(G)$.

Theorem 2: For the family $\mathcal{G}$ containing all finite simple graphs, isomorphism is an equivalence relation.

DIY: write complete proof

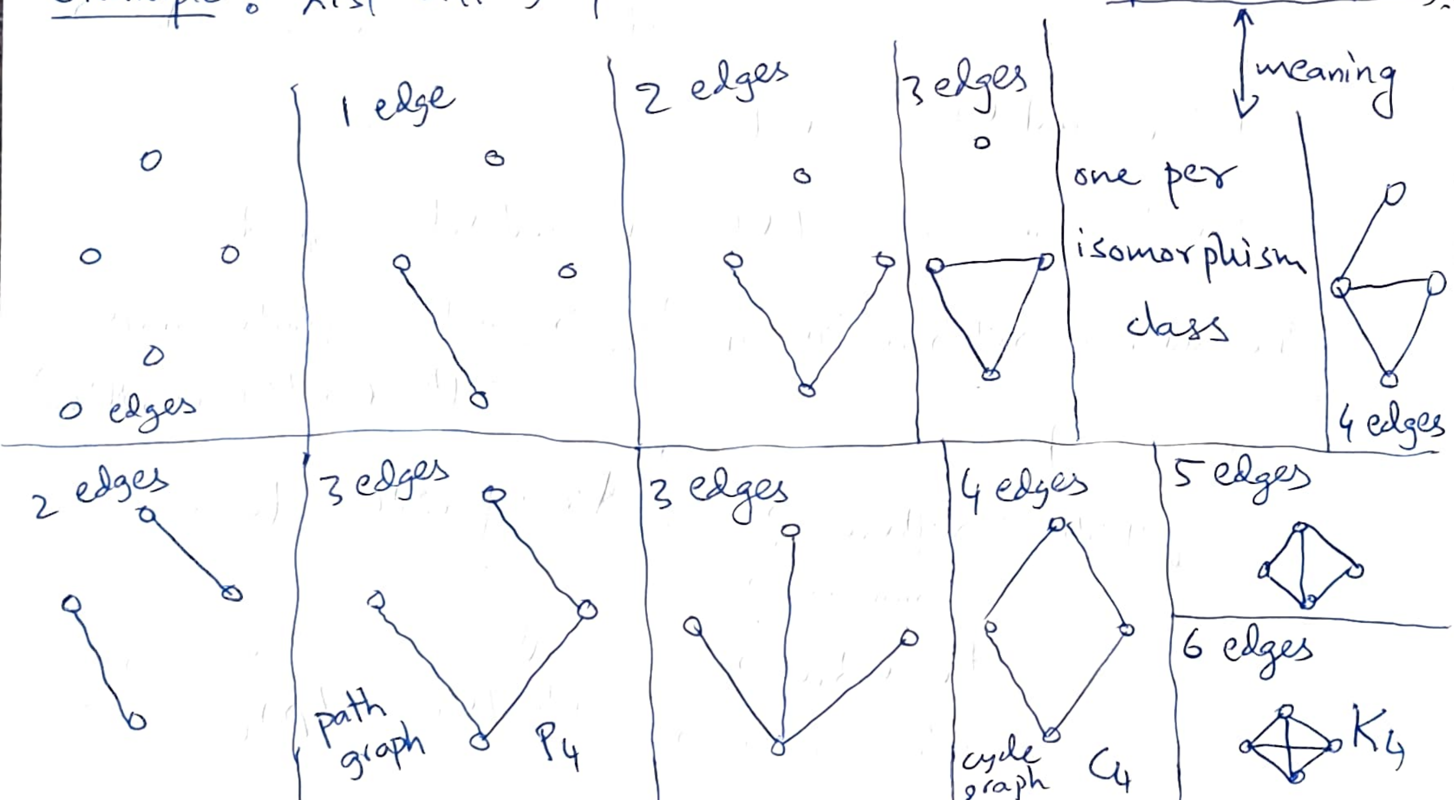
$\mathcal{G}$



The equivalence classes of $\mathcal{G}$ are called isomorphism classes.

Example: List all simple graphs on 4 vertices (up to isomorphism).

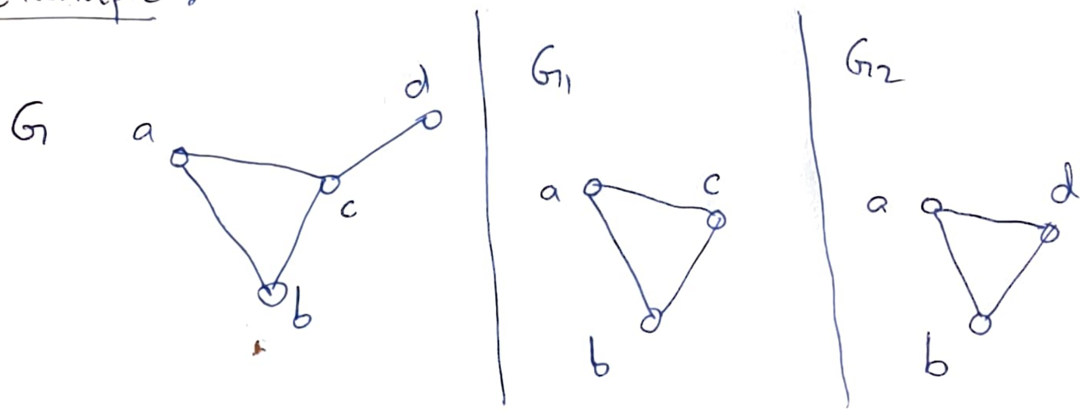
meaning: one per isomorphism class



DIY: List all simple graphs on 5 vertices up to isomorphism.

Subgraph (of a graph)

Example:



Question: Which of G_1 & G_2 is a subgraph of G?

↓

Two answers

If we pay attention to labels (of vertices), G_1 is a subgraph of G but G_2 is NOT a subgraph of G.

If we do NOT pay attention to labels (of vertices), G_1 & G_2 (both are isomorphic) and are both subgraphs of G.

These two answers lead to two different definitions

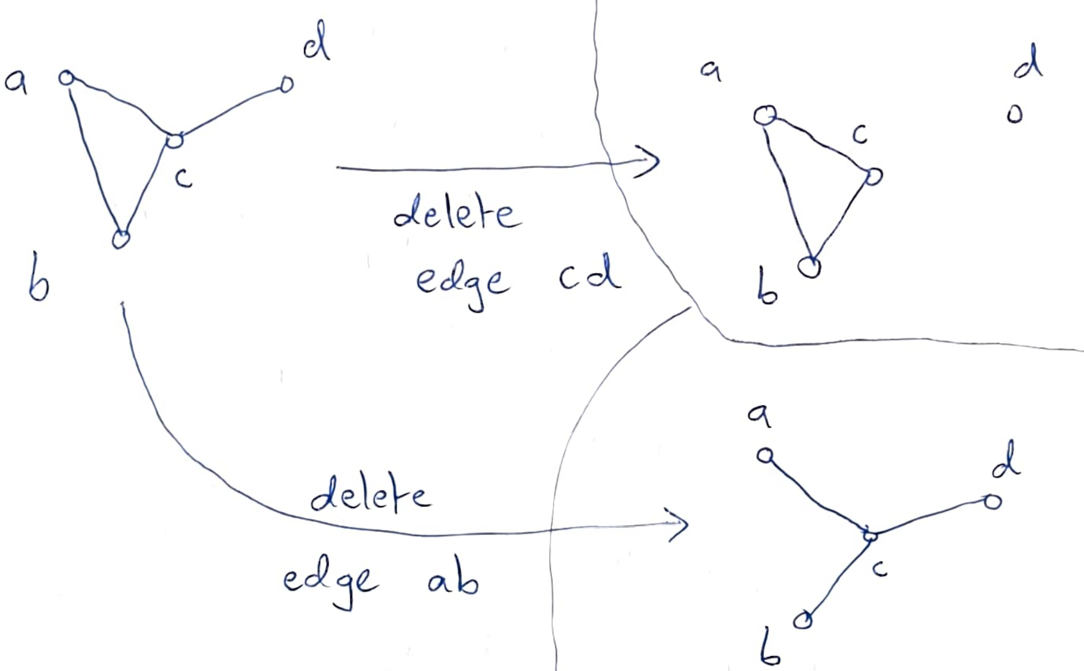
one in the case of labeled graphs

one in the case of unlabeled graphs $\updownarrow$ same as graphs up to isomorphism

Deletion of edge e:

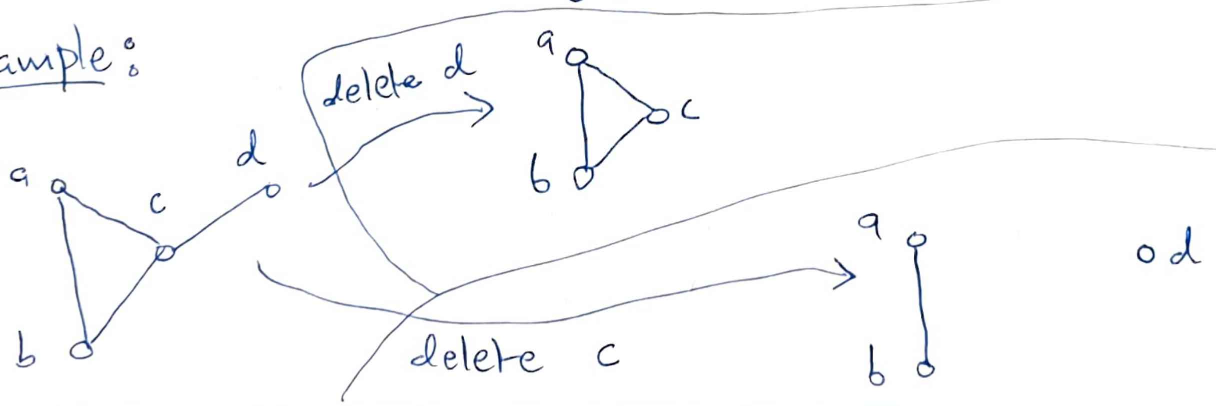
Just remove the edge e from the graph to get a new graph.

Example:



Deletion of vertex v: Just remove the vertex v & all edges incident with v.

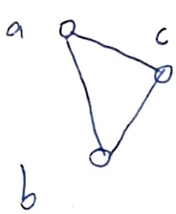
Example:

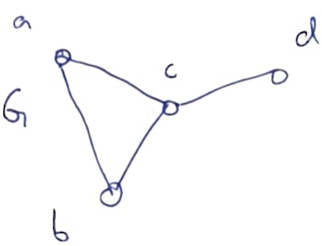


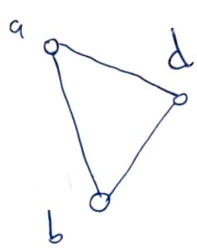
Definition 1 (for labeled graphs): (somewhat boring)

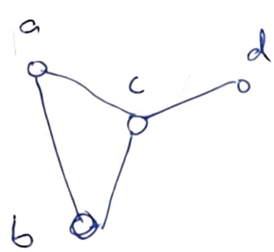
A labeled graph H is a subgraph of a labeled graph G if H can be obtained from G by deleting edges and/or vertices.

As per this definition:

G_1 

is a subgraph of G 

but G_2 

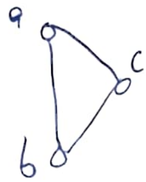
is NOT a subgraph of G 

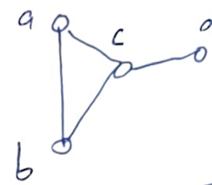
Definition 2: (for unlabeled graphs): (more exciting)

A graph H is a subgraph of a graph G if some graph isomorphic to H can be obtained from G by deleting edges and/or vertices.

As per definition 2, both G_1 & G_2 are subgraphs of G.

does NOT care about vertex/edge labels





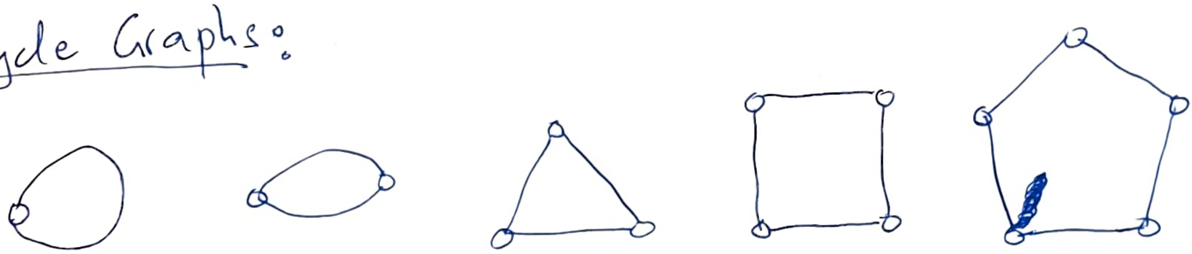
We also say that G contains G_1 & G_2 as subgraphs.

Generally, we will use definition 2.

Next GOAL: To define a class of graphs using this definition.

Before that: we will define a new family of graphs.

Cycle Graphs:



C_1 C_2 C_3 C_4 C_5 ....

DIY: Write down a "boring" definition of cycle graphs — similar to definition of walks/trails/paths.

More exciting definition: Cycle graphs are connected graphs where each vertex has degree 2.

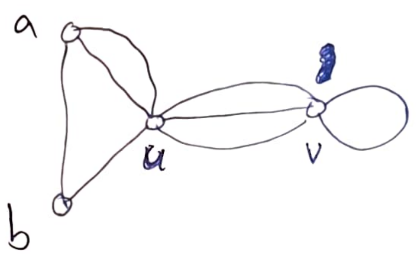
degree of a vertex: # of edges incident; count loops twice

DIY: Prove that both definitions are same (aka equivalent).

Degree of a vertex v: # of edges incident at a vertex v where loops are counted twice

↓ denoted $d(v)$

Example:

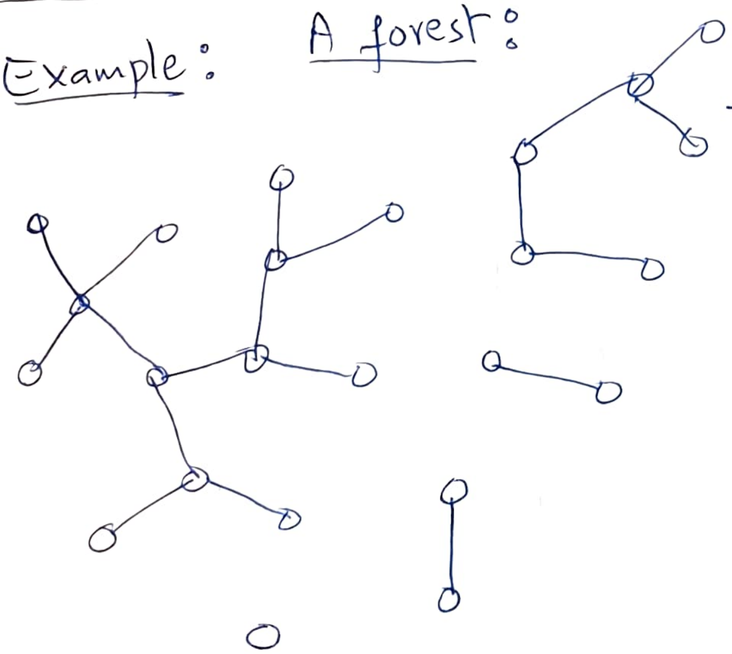


$d(a) = 3$
$d(b) = 2$
$d(u) = 6$
$d(v) = 5$

Now we define a class of graphs using the concepts of subgraphs & cyclegraphs: → that is extremely important/useful in CS/Math

A graph G is a $\boxed{\text{FOREST}}$ if it does NOT contain any cycle (graph) as a subgraph.

Example: A forest:



→ This forest has 5 connected components

↓ these are called trees.

Observe: A forest is always a simple graph.

↓ why?

Definition: A $\boxed{\text{tree}}$ is a connected forest.