CS 105: Department Introductory Course on Discrete Structures

Instructor : S. Akshay

Graph theory

Graph Isomorphism, Cliques and Independent sets, Connected Components

Lecture 20
Oct 15 2025

Logistics

Evaluation

- Quiz 1 (15%), Midsem (25%), 2 Popquizzes over, more to come? (5%)

Logistics

Evaluation

- Quiz 1 (15%), Midsem (25%), 2 Popquizzes over, more to come? (5%)
- Upcoming Quiz 2 (15%) next Thursday, Endsem (40%)

Logistics

Evaluation

- Quiz 1 (15%), Midsem (25%), 2 Popquizzes over, more to come? (5%)
- Upcoming Quiz 2 (15%) next Thursday, Endsem (40%)
- So about 45% done, 55% left!

Logistics

Evaluation

- Quiz 1 (15%), Midsem (25%), 2 Popquizzes over, more to come? (5%)
- Upcoming Quiz 2 (15%) next Thursday, Endsem (40%)
- So about 45% done, 55% left!

Quiz 2 logistics: 3:45pm on Thursday 23rd Oct

- Syllabus: Lattices, Combinatorics and Graph theory till this Tuesday
- Syllabus: Lectures 12 - 19 (both included!)
- Venue
  - LH 301 (Roll numbers: 25B0901– 25B1001)
  - LH 302 (Roll numbers: 25B1002 onwards)
  - LT 305 (PwD students)

Logistics

Evaluation

- Quiz 1 (15%), Midsem (25%), 2 Popquizzes over, more to come? (5%)
- Upcoming Quiz 2 (15%) next Thursday, Endsem (40%)
- So about 45% done, 55% left!

Quiz 2 logistics: 3:45pm on Thursday 23rd Oct

- Syllabus: Lattices, Combinatorics and Graph theory till this Tuesday
- Syllabus: Lectures 12 - 19 (both included!)
- Venue
  - LH 301 (Roll numbers: 25B0901– 25B1001)
  - LH 302 (Roll numbers: 25B1002 onwards)
  - LT 305 (PwD students)

Do NOT cheat!

Pop Quiz:

Logistics

Evaluation

- Quiz 1 (15%), Midsem (25%), 2 Popquizzes over, more to come? (5%)
- Upcoming Quiz 2 (15%) next Thursday, Endsem (40%)
- So about 45% done, 55% left!

Quiz 2 logistics: 3:45pm on Thursday 23rd Oct

- Syllabus: Lattices, Combinatorics and Graph theory till this Tuesday
- Syllabus: Lectures 12 - 19 (both included!)
- Venue
  - LH 301 (Roll numbers: 25B0901– 25B1001)
  - LH 302 (Roll numbers: 25B1002 onwards)
  - LT 305 (PwD students)

Do NOT cheat!

Pop Quiz:What is the punishment for cheating in the exam?

Logistics

Evaluation

- Quiz 1 (15%), Midsem (25%), 2 Popquizzes over, more to come? (5%)
- Upcoming Quiz 2 (15%) next Thursday, Endsem (40%)
- So about 45% done, 55% left!

Quiz 2 logistics: 3:45pm on Thursday 23rd Oct

- Syllabus: Lattices, Combinatorics and Graph theory till this Tuesday
- Syllabus: Lectures 12 - 19 (both included!)
- Venue
  - LH 301 (Roll numbers: 25B0901– 25B1001)
  - LH 302 (Roll numbers: 25B1002 onwards)
  - LT 305 (PwD students)

Do NOT cheat!

Pop Quiz:What is the punishment for cheating in the exam? FAIL/FR Grade

Part 4: Graph theory

Recap of last few lectures:

1. Basics: graphs, paths, cycles, walks, trails; connected graphs.
2. Eulerian graphs and a characterization in terms of degrees of vertices.
3. Bipartite graphs and a characterization in terms of odd length cycles.

Reference: Sections 1.1-1.3 of Chapter 1 from Douglas West.

Part 4: Graph theory

Recap of last few lectures:

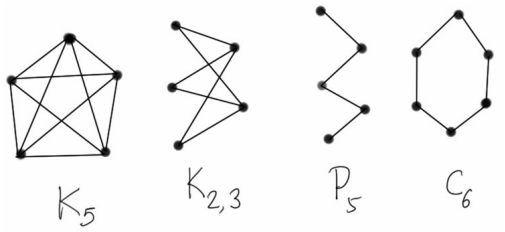
1. Basics: graphs, paths, cycles, walks, trails; connected graphs.
2. Eulerian graphs and a characterization in terms of degrees of vertices.
3. Bipartite graphs and a characterization in terms of odd length cycles.

Reference: Sections 1.1-1.3 of Chapter 1 from Douglas West.

Today, we start with:

Graph representations and naming

Some special graphs and notations



- Complete graphs K_n
- Complete bipartite graphs $K_{i,j}$
- Paths P_n
- Cycles C_n

Some special graphs and notations

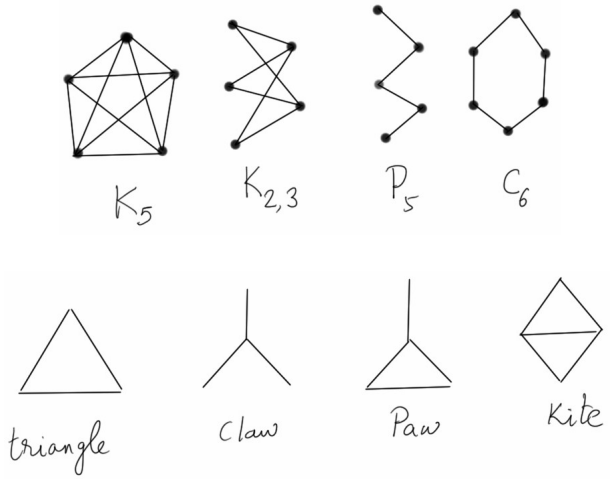


Figure: A whole graph zoo!

Are these graphs the same?



Are these graphs the same?

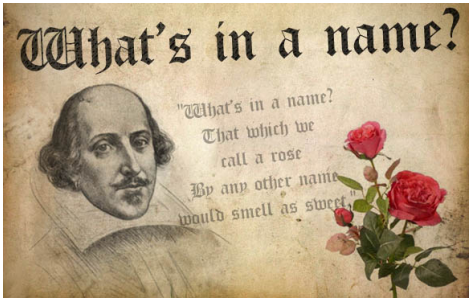


- To compare graphs, we need to name them!

Are these graphs the same?



- To compare graphs, we need to name them!



Representing and comparing graphs

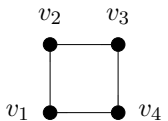
We start with simple graphs...



To represent it, we need to name the vertices...

Representing and comparing graphs

We start with simple graphs...



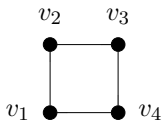
To represent it, we need to name the vertices...

- As an adjacency list:

v_1	v_2, v_4
v_2	v_1, v_3
v_3	v_2, v_4
v_4	v_1, v_3

Representing and comparing graphs

We start with simple graphs...



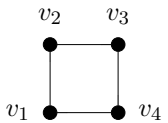
To represent it, we need to name the vertices...

- As an adjacency matrix:

$$\begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \begin{array}{c} \begin{array}{cccc} v_1 & v_2 & v_3 & v_4 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix} \end{array}$$

Representing and comparing graphs

We start with simple graphs...



To represent it, we need to name the vertices...

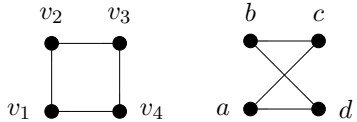
- As an adjacency matrix:

$$\begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \begin{array}{c} \begin{array}{cccc} v_1 & v_2 & v_3 & v_4 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix} \end{array}$$

- But we want to study properties that are independent of the naming, e.g., connectivity.
- Are two given graphs the "same", wrt these properties?

Representing and comparing graphs

We start with simple graphs...



To represent it, we need to name the vertices...

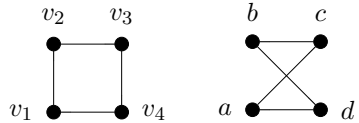
- As an adjacency matrix:

$$\begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \begin{array}{c} \begin{array}{cccc} v_1 & v_2 & v_3 & v_4 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix} \end{array}$$

- But we want to study properties that are independent of the naming, e.g., connectivity.
- Are two given graphs the "same", wrt these properties?

Representing and comparing graphs

We start with simple graphs...



To represent it, we need to name the vertices...

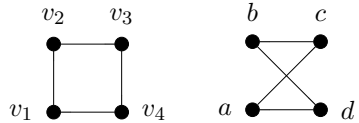
- As an adjacency matrix:

$$\begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \begin{array}{c} \begin{array}{cccc} v_1 & v_2 & v_3 & v_4 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix} \end{array} \qquad \begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix} \end{array}$$

- But we want to study properties that are independent of the naming, e.g., connectivity.
- Are two given graphs the "same", wrt these properties?

Representing and comparing graphs

We start with simple graphs...



To represent it, we need to name the vertices...

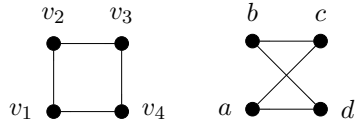
- As an adjacency matrix:

$$\begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \begin{array}{c} \begin{array}{cccc} v_1 & v_2 & v_3 & v_4 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix} \end{array} \qquad \begin{array}{c} \\ v_1 \\ v_3 \\ v_2 \\ v_4 \end{array} \begin{array}{c} \begin{array}{cccc} v_1 & v_3 & v_2 & v_4 \end{array} \\ \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix} \end{array} \qquad \begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix} \end{array}$$

- But we want to study properties that are independent of the naming, e.g., connectivity.
- Are two given graphs the "same", wrt these properties?

Representing and comparing graphs

We start with simple graphs...



To represent it, we need to name the vertices...

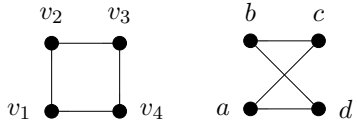
- As an adjacency matrix:

$$\begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \begin{array}{c} \begin{array}{cccc} v_1 & v_2 & v_3 & v_4 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix} \end{array} \qquad \begin{array}{c} \\ v_1 \\ v_3 \\ v_2 \\ v_4 \end{array} \begin{array}{c} \begin{array}{cccc} v_1 & v_3 & v_2 & v_4 \end{array} \\ \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix} \end{array} \qquad \begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix} \end{array}$$

- Reordering of vertices same as applying permutation to rows/colums of $A(G)$.
- So, it seems two graphs are "same" if by reordering and renaming the vertices we get the same graph/matrix.

Representing and comparing graphs

We start with simple graphs...



To represent it, we need to name the vertices...

- As an adjacency matrix:

$$\begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \begin{array}{c} \begin{array}{cccc} v_1 & v_2 & v_3 & v_4 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix} \end{array} \qquad \begin{array}{c} \\ v_1 \\ v_3 \\ v_2 \\ v_4 \end{array} \begin{array}{c} \begin{array}{cccc} v_1 & v_3 & v_2 & v_4 \end{array} \\ \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix} \end{array} \qquad \begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix} \end{array}$$

- Reordering of vertices same as applying permutation to rows/colums of $A(G)$.
- So, it seems two graphs are "same" if by reordering and renaming the vertices we get the same graph/matrix.
- How do we formalize this?

Isomorphism

Definition

An isomorphism from simple graph G to H is a bijection $f : V(G) \to V(H)$ such that $uv \in E(G)$ iff $f(u)f(v) \in E(H)$.

Isomorphism

Definition

An isomorphism from simple graph G to H is a bijection $f : V(G) \to V(H)$ such that $uv \in E(G)$ iff $f(u)f(v) \in E(H)$.

- Thus, it is a bijection that "preserves" the edge relation.
- Can be checked using adjacency matrix by reordering/renaming.

Isomorphism

Definition

An isomorphism from simple graph G to H is a bijection $f : V(G) \to V(H)$ such that $uv \in E(G)$ iff $f(u)f(v) \in E(H)$.

- Thus, it is a bijection that "preserves" the edge relation.
- Can be checked using adjacency matrix by reordering/renaming.
- What are the properties of this function/relation:
  $R = \{(G, H) \mid \exists$ an isomorphism from G to $H\}$.

Isomorphism

Definition

An isomorphism from simple graph G to H is a bijection $f : V(G) \to V(H)$ such that $uv \in E(G)$ iff $f(u)f(v) \in E(H)$.

- Thus, it is a bijection that "preserves" the edge relation.
- Can be checked using adjacency matrix by reordering/renaming.
- What are the properties of this function/relation:
  $R = \{(G, H) \mid \exists$ an isomorphism from G to $H\}$.

Proposition

The isomorphism relation is an equivalence relation.

Isomorphism

Definition

An isomorphism from simple graph G to H is a bijection $f : V(G) \to V(H)$ such that $uv \in E(G)$ iff $f(u)f(v) \in E(H)$.

- Thus, it is a bijection that "preserves" the edge relation.
- Can be checked using adjacency matrix by reordering/renaming.
- What are the properties of this function/relation:
  $R = \{(G, H) \mid \exists$ an isomorphism from G to $H\}$.

Proposition

The isomorphism relation is an equivalence relation.

- The equivalence classes are called isomorphism classes.

Isomorphism

Definition

An isomorphism from simple graph G to H is a bijection $f : V(G) \to V(H)$ such that $uv \in E(G)$ iff $f(u)f(v) \in E(H)$.

- Thus, it is a bijection that "preserves" the edge relation.
- Can be checked using adjacency matrix by reordering/renaming.
- What are the properties of this function/relation:
  $R = \{(G, H) \mid \exists$ an isomorphism from G to $H\}$.

Proposition

The isomorphism relation is an equivalence relation.

- The equivalence classes are called isomorphism classes.
- When we talked about an "unlabeled" graph till now, we actually meant the isomorphism class of that graph!

Properties of isomorphic graphs

Intuitively, if two graphs are isomorphic then all structural properties, i.e., properties that do not depend on the naming of vertices are preserved.

Properties of isomorphic graphs

Intuitively, if two graphs are isomorphic then all structural properties, i.e., properties that do not depend on the naming of vertices are preserved.

Theorem

If G is isomorphic to H, then the following properties are preserved:

1. G, H have same # vertices.
2. G, H have same # edges.

Properties of isomorphic graphs

Intuitively, if two graphs are isomorphic then all structural properties, i.e., properties that do not depend on the naming of vertices are preserved.

- Are C_5 and $P_5 \cup \{e\}$ isomorphic?

Theorem

If G is isomorphic to H, then the following properties are preserved:

1. G, H have same # vertices.
2. G, H have same # edges.

Properties of isomorphic graphs

Intuitively, if two graphs are isomorphic then all structural properties, i.e., properties that do not depend on the naming of vertices are preserved.

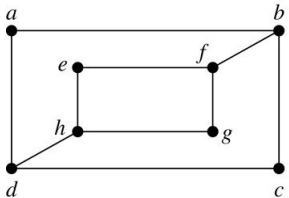
Theorem

If G is isomorphic to H, then the following properties are preserved:

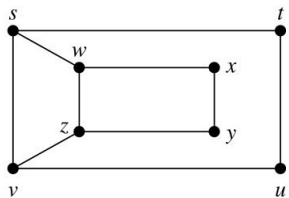
1. G, H have same # vertices.
2. G, H have same # edges.
3. G, H have the same # vertices of degree k, $\forall k \in \mathbb{N}$.

Properties of isomorphic graphs

Intuitively, if two graphs are isomorphic then all structural properties, i.e., properties that do not depend on the naming of vertices are preserved.



G



H

Properties of isomorphic graphs

Intuitively, if two graphs are isomorphic then all structural properties, i.e., properties that do not depend on the naming of vertices are preserved.

Theorem

If G is isomorphic to H, then the following properties are preserved:

1. G, H have same # vertices.
2. G, H have same # edges.
3. G, H have the same # vertices of degree k, $\forall k \in \mathbb{N}$.

Properties of isomorphic graphs

Intuitively, if two graphs are isomorphic then all structural properties, i.e., properties that do not depend on the naming of vertices are preserved.

Theorem

If G is isomorphic to H, then the following properties are preserved:

1. G, H have same # vertices.
2. G, H have same # edges.
3. G, H have the same # vertices of degree k, $\forall k \in \mathbb{N}$.
4. G has k paths/cycles of length r iff H has k paths/cycles of length r.

Properties of isomorphic graphs

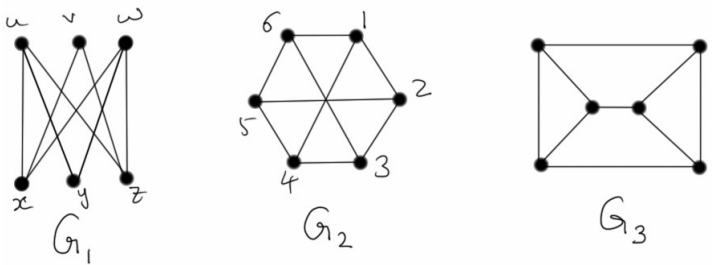
Intuitively, if two graphs are isomorphic then all structural properties, i.e., properties that do not depend on the naming of vertices are preserved.

Theorem

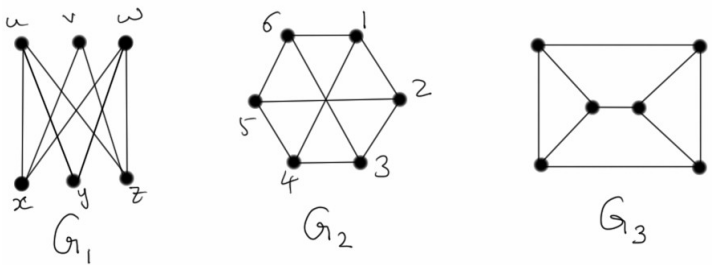
If G is isomorphic to H, then the following properties are preserved:

1. G, H have same # vertices.
2. G, H have same # edges.
3. G, H have the same # vertices of degree k, $\forall k \in \mathbb{N}$.
4. G has k paths/cycles of length r iff H has k paths/cycles of length r.
5. G is bipartite iff H is bipartite.
6. ...

Graph isomorphism

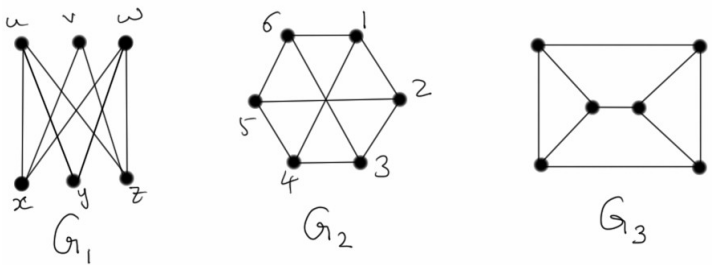


Graph isomorphism



Exercise 1: Which of these graphs are isomorphic? Justify!

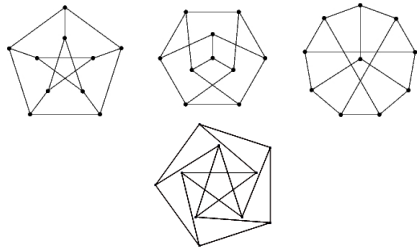
Graph isomorphism



Exercise 1: Which of these graphs are isomorphic? Justify!

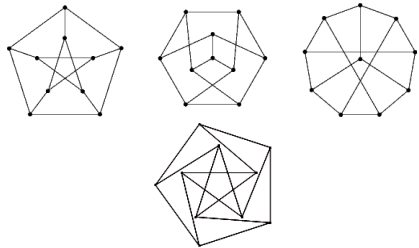
- To show that two graphs are isomorphic, you have to
  1. give names to vertices
  2. specify a bijection
  3. check that it preserves the adjacency relation
- To show that two graphs are non-isomorphic, find a structural property that is different.

Is checking graph isomorphism easy?



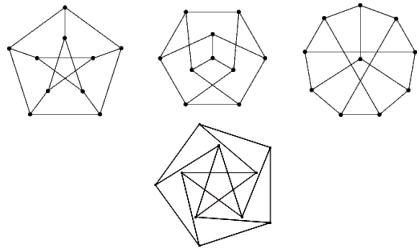
- Exercise 2: Which of these graphs are isomorphic?

Is checking graph isomorphism easy?



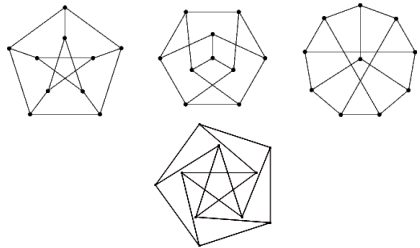
- Exercise 2: Which of these graphs are isomorphic?
- A: All of them!

Is checking graph isomorphism easy?



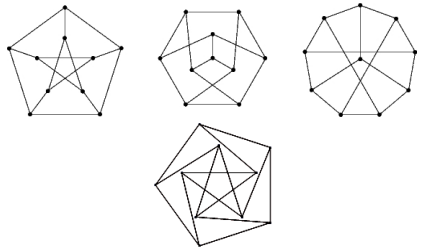
- Exercise 2: Which of these graphs are isomorphic?
- A: All of them!
- This graph is called the Petersen graph and has some very interesting propreties.

Is checking graph isomorphism easy?



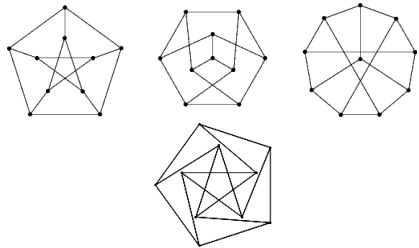
- Exercise 2: Which of these graphs are isomorphic?
- A: All of them!
- This graph is called the Petersen graph and has some very interesting propreties.
  - vertices are 2-element subsets of 5-element set and edges are pairs of disjoint 2-element subsets.

Is checking graph isomorphism easy?



- Exercise 2: Which of these graphs are isomorphic?
- A: All of them!
- This graph is called the Petersen graph and has some very interesting propreties.
  - vertices are 2-element subsets of 5-element set and edges are pairs of disjoint 2-element subsets.
  - 2 vertices that do not share an edge, have exactly 1 common nbr.

Is checking graph isomorphism easy?



- Exercise 2: Which of these graphs are isomorphic?
- A: All of them!
- This graph is called the Petersen graph and has some very interesting propreties.
  - vertices are 2-element subsets of 5-element set and edges are pairs of disjoint 2-element subsets.
  - 2 vertices that do not share an edge, have exactly 1 common nbr.

Further reading: Graph and sub-graph isomorphism problems.

Graph Automorphisms

An isomorphism from simple graph G to H is a bijection $f : V(G) \to V(H)$ such that $uv \in E(G)$ iff $f(u)f(v) \in E(H)$.

Graph Automorphisms

An isomorphism from simple graph G to H is a bijection $f : V(G) \to V(H)$ such that $uv \in E(G)$ iff $f(u)f(v) \in E(H)$. What if $G = H$?

Graph Automorphisms

An isomorphism from simple graph G to H is a bijection $f : V(G) \to V(H)$ such that $uv \in E(G)$ iff $f(u)f(v) \in E(H)$. What if $G = H$?

Definition

An automorphism of G is an isomorphism from G to itself, i.e. a bijection $f : V(G) \to V(G)$ s.t. $uv \in E(G)$ iff $f(u)f(v) \in E(G)$.

Graph Automorphisms

An isomorphism from simple graph G to H is a bijection $f : V(G) \to V(H)$ such that $uv \in E(G)$ iff $f(u)f(v) \in E(H)$. What if $G = H$?

Definition

An automorphism of G is an isomorphism from G to itself, i.e. a bijection $f : V(G) \to V(G)$ s.t. $uv \in E(G)$ iff $f(u)f(v) \in E(G)$.

- What are the automorphisms of P_4?

Graph Automorphisms

An isomorphism from simple graph G to H is a bijection $f : V(G) \to V(H)$ such that $uv \in E(G)$ iff $f(u)f(v) \in E(H)$. What if $G = H$?

Definition

An automorphism of G is an isomorphism from G to itself, i.e. a bijection $f : V(G) \to V(G)$ s.t. $uv \in E(G)$ iff $f(u)f(v) \in E(G)$.

- What are the automorphisms of P_4?
- How many automorphisms does K_n have?

Graph Automorphisms

An isomorphism from simple graph G to H is a bijection $f : V(G) \to V(H)$ such that $uv \in E(G)$ iff $f(u)f(v) \in E(H)$. What if $G = H$?

Definition

An automorphism of G is an isomorphism from G to itself, i.e. a bijection $f : V(G) \to V(G)$ s.t. $uv \in E(G)$ iff $f(u)f(v) \in E(G)$.

- What are the automorphisms of P_4?
- How many automorphisms does K_n have?
- How many automorphisms does $K_{r,s}$ have?

Graph Automorphisms

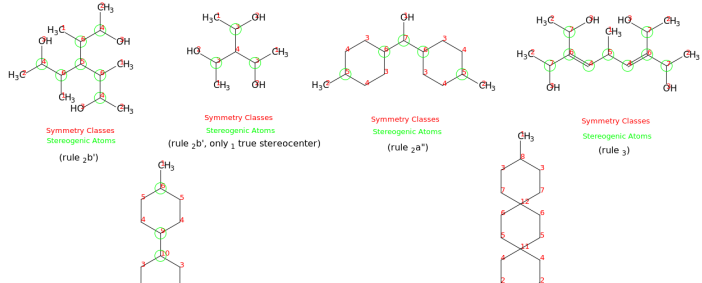
An isomorphism from simple graph G to H is a bijection $f : V(G) \to V(H)$ such that $uv \in E(G)$ iff $f(u)f(v) \in E(H)$.

Definition

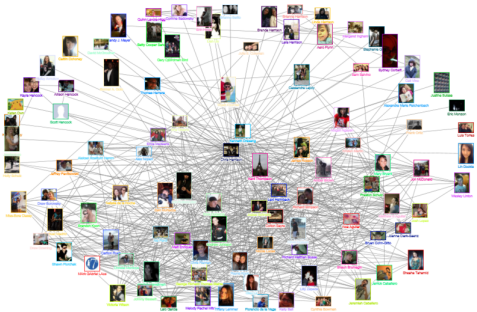
An automorphism of G is an isomorphism from G to itself, i.e. a bijection $f : V(G) \to V(G)$ s.t. $uv \in E(G)$ iff $f(u)f(v) \in E(G)$.

Automorphisms are a measure of symmetry.

Practical applications in graph drawing, visualization, molecular symmetry, structured boolean satisfiability, formal verification

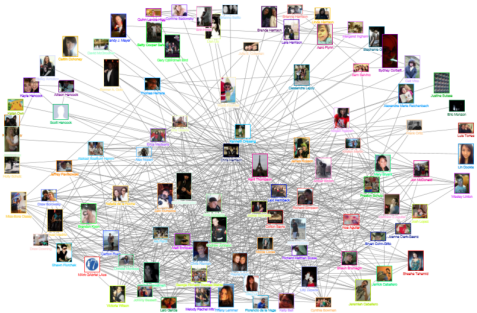


Cliques and independent sets



- Consider a large social network graph where friends are linked by an edge.
- What is the largest clique of friends?
- If we want to spread a youtube video, how many people should we send it to so that we are guaranteed everyone will see it (assuming friends forward to each other)?

Cliques and independent sets



Cliques and independent sets

- A clique in a graph is a set of pairwise adjacent vertices.
- An independent set in a graph is a set of pairwise non-adjacent vertices.

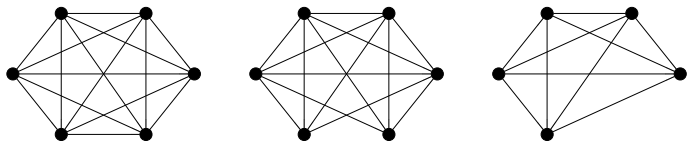
Size of a clique/independent set is the number of vertices in it.

Cliques and independent sets

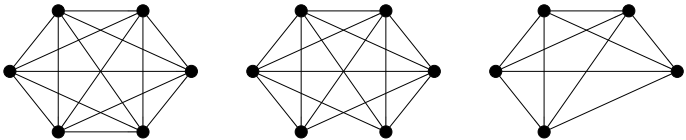
Cliques and independent sets

- A clique in a graph is a set of pairwise adjacent vertices.
- An independent set in a graph is a set of pairwise non-adjacent vertices.

Size of a clique/independent set is the number of vertices in it.



Cliques and independent sets



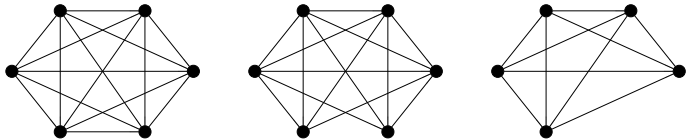
- Thus, a clique in a graph G is a complete subgraph of G.

Subgraphs of a graph G

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

- $V(H) \subseteq V(G)$ and
- $E(H) \subseteq E(G)$
  & assignment of endpoints to edges in H is same as in G.

Cliques and independent sets



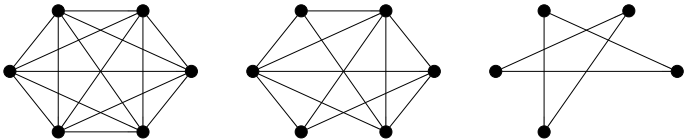
- Thus, a clique in a graph G is a complete subgraph of G.
- An independent set in G is a complete subgraph of $\overline{G}$, where $\overline{G}$ is the complement of G obtained by making all adjacent vertices non-adjacent and vice versa.

Subgraphs of a graph G

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

- $V(H) \subseteq V(G)$ and
- $E(H) \subseteq E(G)$
  & assignment of endpoints to edges in H is same as in G.

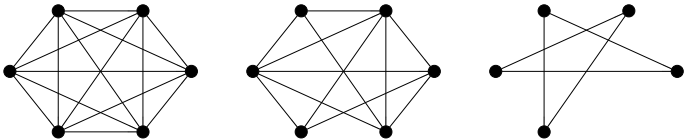
Cliques and independent sets



Questions:

1. What is the size of the largest clique/independent set in each of the above graphs? In any complete graph?

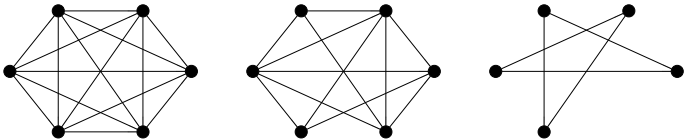
Cliques and independent sets



Questions:

1. What is the size of the largest clique/independent set in each of the above graphs? In any complete graph?
2. Given graph G, integer k, does G have a clique of size k?

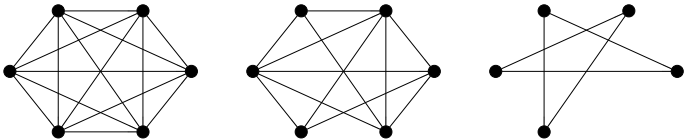
Cliques and independent sets



Questions:

1. What is the size of the largest clique/independent set in each of the above graphs? In any complete graph?
2. Given graph G, integer k, does G have a clique of size k?
3. In a graph with 6 vertices, can you always find a clique or an independent set of size 3?

Cliques and independent sets

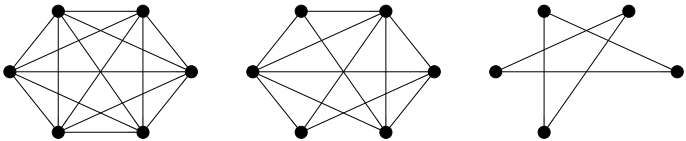


Questions:

1. What is the size of the largest clique/independent set in each of the above graphs? In any complete graph?
2. Given graph G, integer k, does G have a clique of size k?
3. In a graph with 6 vertices, can you always find a clique or an independent set of size 3?

   Yes, because $R(3,3) = 6$!

Cliques and independent sets



Questions:

1. What is the size of the largest clique/independent set in each of the above graphs? In any complete graph?
2. Given graph G, integer k, does G have a clique of size k?
3. In a graph with 6 vertices, can you always find a clique or an independent set of size 3?

   Yes, because $R(3,3) = 6$!

Ramsey's theorem - restated

In any graph with $R(k,\ell)$ vertices, there exists either a clique of size k or an independent set of size ℓ.

Relations between vertices

- We considered a relation between graphs (isomorphism).
- But what about between vertices? Can you think of interesting relations?

Relations between vertices

- We considered a relation between graphs (isomorphism).
- But what about between vertices? Can you think of interesting relations?

1. Adjacency: uRv iff there is an edge between u and v. Any nice properties?

Relations between vertices

- We considered a relation between graphs (isomorphism).
- But what about between vertices? Can you think of interesting relations?

1. Adjacency: uRv iff there is an edge between u and v. Any nice properties?
2. Connectedness: uPv iff there is a path between u and v.

Relations between vertices

- We considered a relation between graphs (isomorphism).
- But what about between vertices? Can you think of interesting relations?

1. Adjacency: uRv iff there is an edge between u and v. Any nice properties?
2. Connectedness: uPv iff there is a path between u and v.

   P, i.e., connectedness is an equivalence relation.

Relations between vertices

- We considered a relation between graphs (isomorphism).
- But what about between vertices? Can you think of interesting relations?

1. Adjacency: uRv iff there is an edge between u and v. Any nice properties?
2. Connectedness: uPv iff there is a path between u and v.

   P, i.e., connectedness is an equivalence relation.

Definition

A (connected) component of G is a maximal connected subgraph, i.e., a subgraph that is connected and is not contained in any other connected subgraph of G.

Thus, equivalence classes of P are the vertex sets of the components of G.

Components and cut-edges

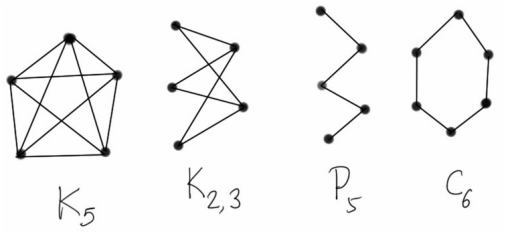
Properties of components

- A component with no edges is called trivial. Thus isolated vertices form trivial components.
- Components are pairwise disjoint.

Components and cut-edges

Properties of components

- A component with no edges is called trivial. Thus isolated vertices form trivial components.
- Components are pairwise disjoint.
- What happens to the number of components when you add or delete an edge?

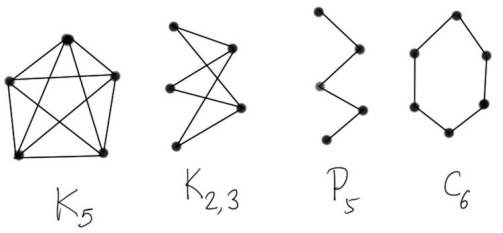


Components and cut-edges

Properties of components

- A component with no edges is called trivial. Thus isolated vertices form trivial components.
- Components are pairwise disjoint.

- What happens to the number of components when you add or delete an edge?
- Edges whose deletion increases # components are called cut-edges.



Components and cut-edges

Properties of components

- A component with no edges is called trivial. Thus isolated vertices form trivial components.
- Components are pairwise disjoint.

- What happens to the number of components when you add or delete an edge?
- Edges whose deletion increases # components are called cut-edges.

Theorem: Characterize cut-edges using cycles

Components and cut-edges

Properties of components

- A component with no edges is called trivial. Thus isolated vertices form trivial components.
- Components are pairwise disjoint.

- What happens to the number of components when you add or delete an edge?
- Edges whose deletion increases # components are called cut-edges.

Theorem: Characterize cut-edges using cycles

H.W. Exercise!

Components and cut-edges

Properties of components

- A component with no edges is called trivial. Thus isolated vertices form trivial components.
- Components are pairwise disjoint.

- What happens to the number of components when you add or delete an edge?
- Edges whose deletion increases # components are called cut-edges.

Theorem: Characterize cut-edges using cycles

H.W. Exercise! An edge is a cut-edge iff it belongs to no cycle.