Graph Representation (图表示）

- 在使用图模型去解决实际问题时，如何方便有效地表示图是非常重要的
- 实际上，有多种表示方法。这里讲授最常用、最方便的几种表示方法
- A graph is a kind of mathematical structure, sometime it seems abstract. Not just a graph drawn on the plane.
- ***But we can draw a real graph on the plane which represents the abstract graph directly and intuitively***. That is the most intuitive way to represent a graph model (图形表示方法).
- 这也正是这种数学结构被称为图的原因

图可以整体地说是一个二元结构，一个点集和一个边集。代表这一个集合V上的一个二元关系E。那么图的表示，需要表示些什么？

When we represent a graph using a tool, what should be expressed?

(1) all vertices必须表示出所有的结点；

(2) The relation between the vertices点之间（对象之间）的关系（边）表达出来；

- 回想二元关系的表示方法…

Graph Representation

- **adjacency list 邻接表表示法**
- Example:

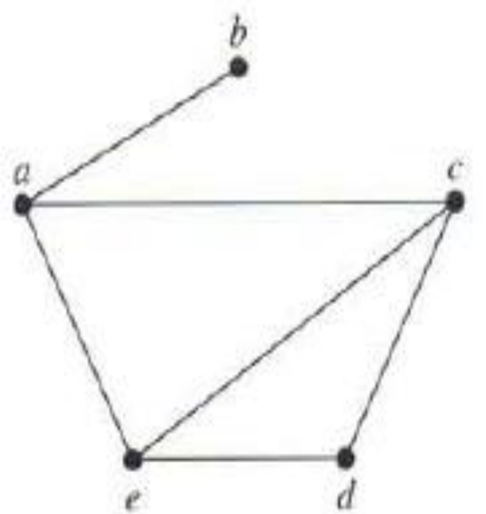


FIGURE 1 A Simple Graph.

TABLE 1 An Adjacency List for a Simple Graph.

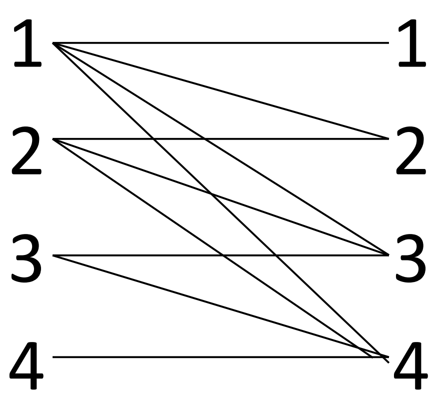
Vertex	*Adjacent Vertices*
a	*b, c, e*
b	*a*
c	*a, d, e*
d	*c, e*
e	*a, c, d*

- 思考问题：如果有多重边，如何表示？

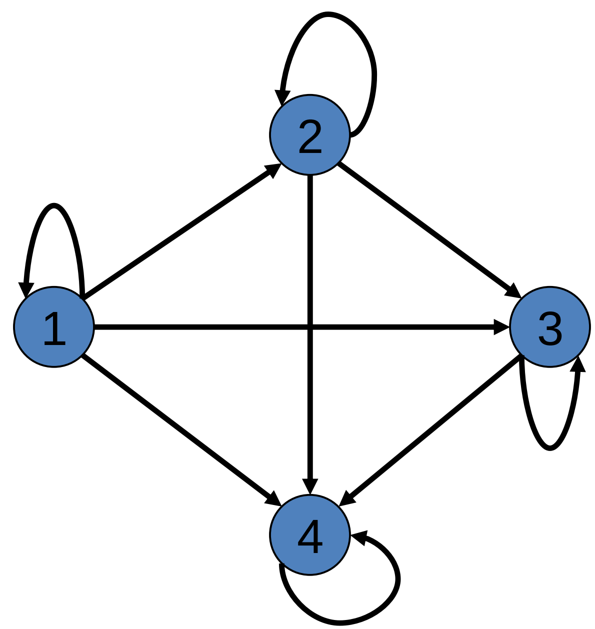
Graph Representation—Adjacency Matrix 邻接矩阵表示法

We already saw a way of representing relations on a set with a Boolean matrix: (曾经的关系矩阵表示法）

R



digraph(*R*)



M_R

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Adjacency Matrix 邻接矩阵表示方法

对于简单有向图，邻接矩阵可以如下这样定义：

For a simple digraph G = (V,E) define matrix A_G =$(a_{ij})_{nxn}$ by:

$$
\begin{array}{c|cccc}
 & v_1 & v_2 & \cdots & v_n \\
\hline
v_1 & * & * & \cdots & * \\
v_2 & * & * & \cdots & * \\
\vdots & * & * & \cdots & * \\
v_n & * & * & \cdots & *
\end{array}
$$

$$
a_{ij} = \begin{cases} 1 & \text{if } \mathrm{v}_\mathrm{i} \rightarrow v_j \in E \\ 0 & \text{otherwise} \end{cases}
$$

Adjacency Matrix -Directed Multigraphs
邻接矩阵表示有向多重图

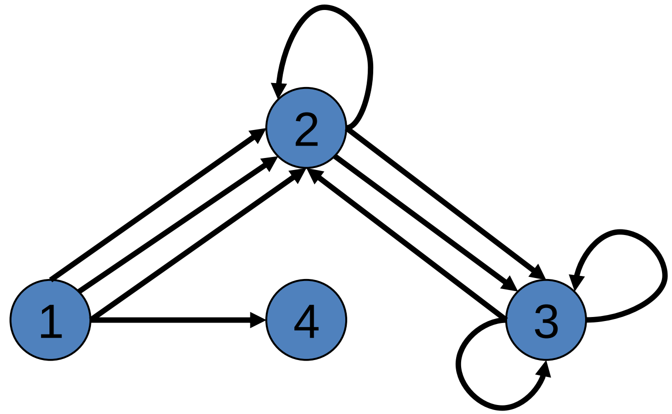
For a directed multigraph $G = (V,E)$ define the

matrix A_G by:

a_{ij} is

– the number of edges from source the i th vertex to target the j th vertex 从第i个结点到第j个结点的边的数目

Adjacency Matrix -Directed Multigraphs



A:

$$\begin{pmatrix} 0 & 3 & 0 & 1 \\ 0 & 1 & 2 & 0 \\ 0 & 1 & 2 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

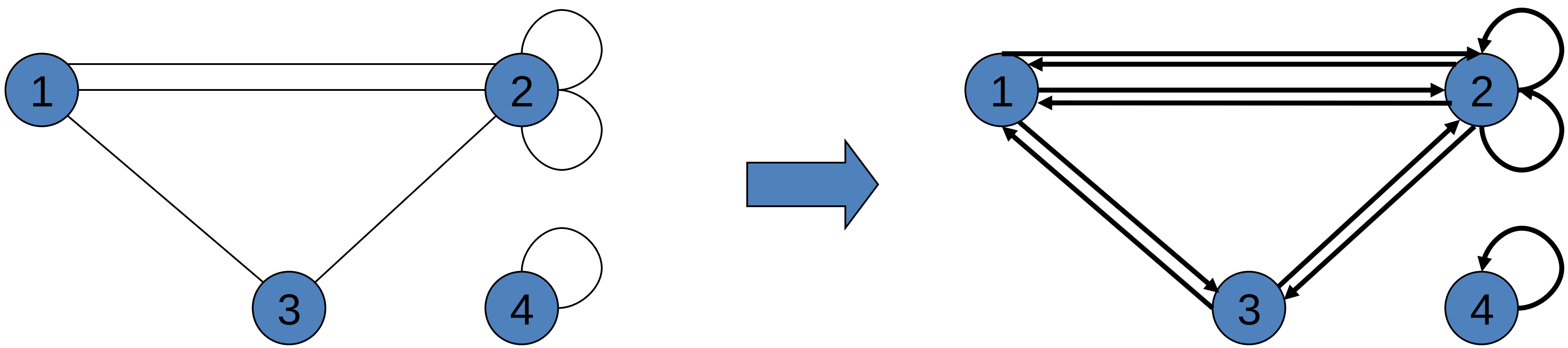
思考

- The definition above is for digraph, what about undirected graph?
- 类似于以上有向图的邻接矩阵表示，思考无向图该如何表示好？

Adjacency Matrix邻接矩阵

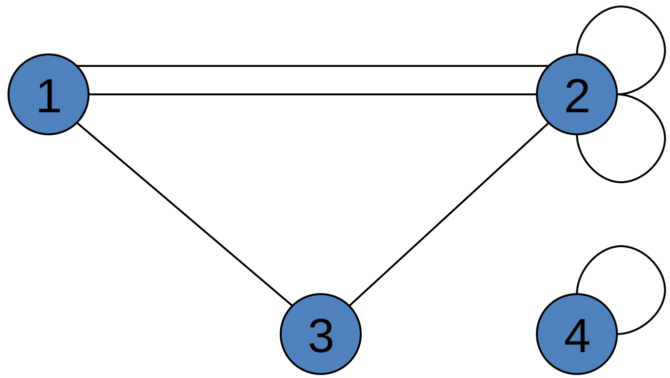
For undirected graph, define the entry a_{ij} as the number of edges between the i^{th} vertex i and j^{th} vertex.

对无向图而言，就是两个点之间的边数定义为相应的矩阵的项；但在计算同一个点之间的单边环时，每一条边（环）只算一个。



Adjacency Matrix-General

A:



$$\begin{pmatrix} 0 & 2 & 1 & 0 \\ 2 & 2 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Notice that matrix is *symmetric*. 为什么会是对称的？

Adjacency Matrix-General

For a simple undirected graph $G = (V,E)$ define the matrix $A_G = (a_{ij})$ by，简单无向图的邻接矩阵定义如下：

$$a_{ij} = \begin{cases} 1 & \text{if } (\mathrm{v}_\mathrm{i}, v_j) \in E \\ 0 & \text{otherwise} \end{cases}$$

For any graph $G = (V,E)$, its adjacency matrix is unique. 唯一

And with an adjacency matrix, we can easily draw its respective graph. 给定邻接矩阵，容易画出相应的图

Adjacency matrix is very useful tool.

Understanding Adjacency Matrix-General

For an **simple undirected graph** $G = (V,E)$ define the matrix A_G by:

- （i,j）项的值为0还是1，表示的就是第i个结点与第j个结点之间是否有边。
- 多重图：表示的是两个结点之间有多少条边
- 有向多重图：表示的是从i点到j点有多少条有向边

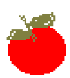
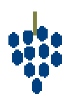
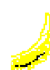
Properties of Adjacency Matrix
--Summary (请同学们自己总结）

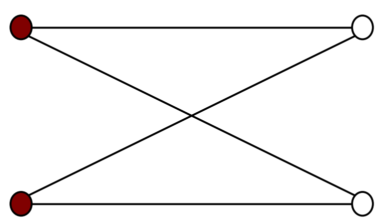
- Properties of the Adjacency Matrix of simple graph
- Properties of the Adjacency Matrix of undirected graph
- Properties of the Adjacency Matrix of multiple graph
- The sum of a row, a column （注意区分有单边环的情况，分开讨论简单图和伪图）
- （在有单边环的伪图中，邻接矩阵的一行的和未必等于相应结点的度）

- Indicent matrix关联矩阵：就是将结点与边的关联关系，用一个矩阵表示出来。用得不多，自己看看该段内容。

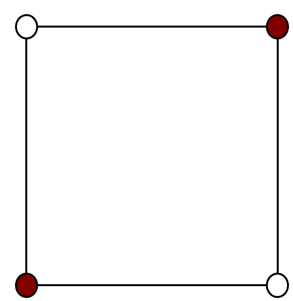
Graph Isomorphism(图的同构)

Various mathematical notions come with their own concept of ***equivalence***, as opposed to equality:

- Equivalence for sets is bi-jectivity:
  - EG { ,   } ≈ {12, 23, 43}
- Equivalence for graphs is **isomorphism**:
  - EG



≈

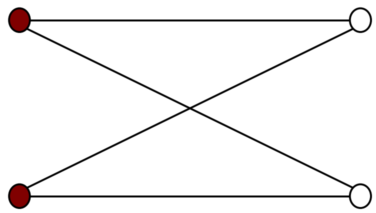


Graph Isomorphism图同构

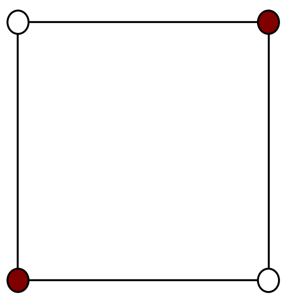
直观地说，两个图的同构是，如果能将一个图重新布局，重新画(redraw)出来（不改变结点之间的关系），变成另一个图，那这两图就是同构的。

Graph *isomorphic* means “same shape”. 同构意味着“形状相同”

例如: we can twist or relabel:



to obtain:



Graph Isomorphism

- Same shape and same structure 相同的形状、相同的结构
- Understanding “Same shape” 好好理解“形状相同、结构相同”

Isomorphism between simple undirected graph 简单无向图同构

Definition: Suppose $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are simple undirected graphs. Let $f : V_1 \rightarrow V_2$ be a function such that:

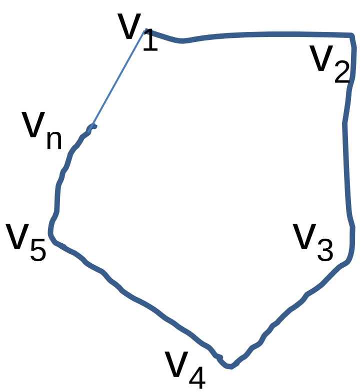
1) f is bijective （双射，点对应）
2) for all vertices u, v in V_1, u and v are adjacent iff $f(u)$ and $f(v)$ are adjacent in G_2. （边对应）

In another word, if there is an edge between u and v, iff there is an edge between f(u) and f(v) in G_2

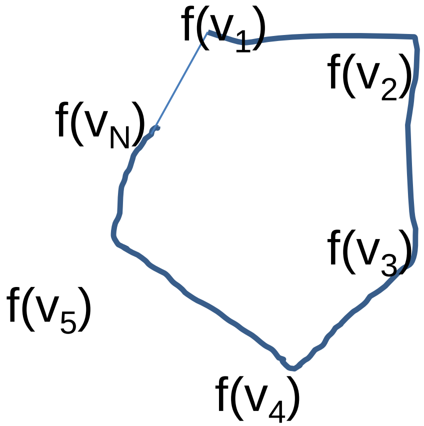
Then f is called an ***isomorphism(*** *同构映射，或简称同构* ***)*** and G_1 is said to be ***isomorphic*** to G_2.

如何理解定义中的第2个条件?

G_1与G_2同构



G_1



G_2

任意无向图的同构

Definition: Suppose $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are pseudographs. Let $f : V_1 \rightarrow V_2$ be a function s.t.:

1) f is bijective （双射， 点对应）
2) for all vertices u,v in V_1, the number of edges between u and v in G_1 is exact same as the number of edges between $f(u)$ and $f(v)$ in G_2. （边对应）

Then f is called an ***isomorphism(同构映射，或简称同构)*** and G_1 is said to be ***isomorphic*** to G_2.

任意有向图的同构

DEF: Suppose $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are directed multigraphs. Let $f : V_1 \rightarrow V_2$ be a function s.t.:

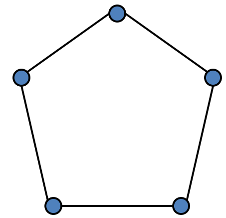
1) f is bijective （双射，结点对应）
2) for all vertices u,v in V_1, the number of edges *from* u to v in G_1 is the same as the number of edges from $f(u)$ to $f(v)$ in G_2. （边对应）

Then f is called an ***isomorphism*** and G_1 is said to be ***isomorphic*** to G_2.

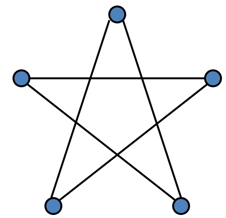
Note: Only difference between two definitions is the italicized "from" in no. 2 (was "between").

图同构举例

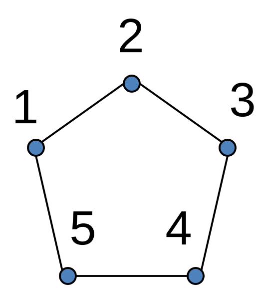
EG: Prove that

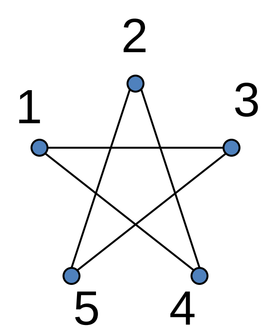


is isomorphic to



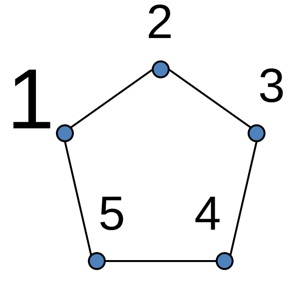
First label the vertices: (to relabel all the vertices 重新标记所以结点)

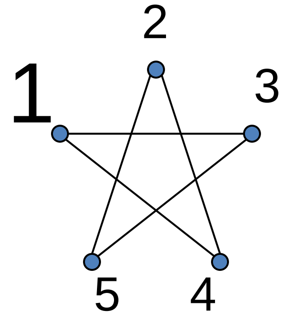




Graph Isomorphism -Example

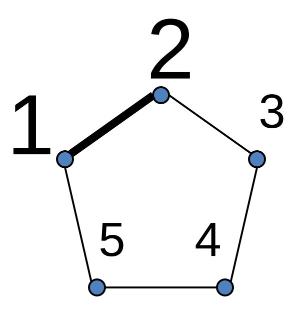
Next, set $f(1) = 1$ and try to walk around clockwise on the star.

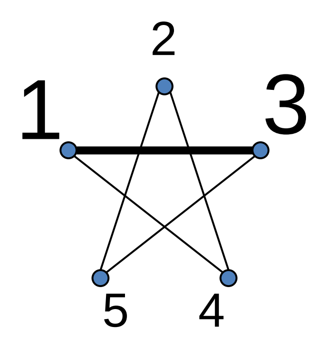




Graph Isomorphism -Example

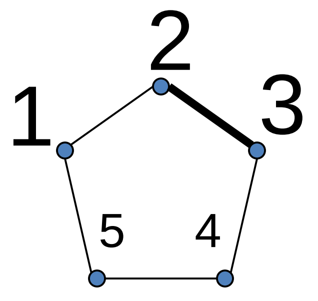
Next, set $f(1) = 1$ and try to walk around clockwise on the star. The next vertex seen is 3, *not* 2 so set $f(2) = 3$.

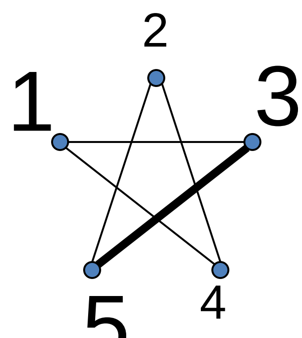




Graph Isomorphism -Example

Next, set $f(1) = 1$ and try to walk around clockwise on the star. The next vertex seen is 3, *not* 2 so set $f(2) = 3$. Next vertex is 5 so set $f(3) = 5$.



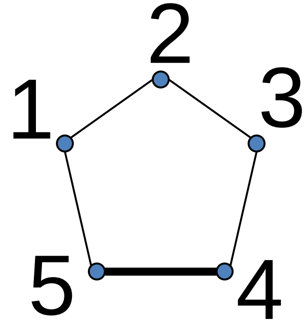


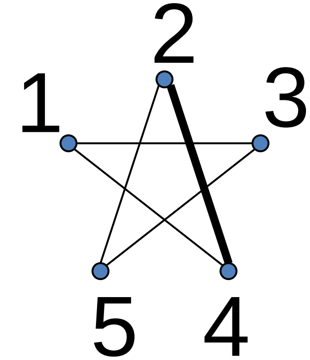
Graph Isomorphism -Example

Next, set *f* (1) = 1 and try to walk around clockwise on the star. The next vertex seen is 3, *not* 2 so set *f* (2) = 3. Next vertex is 5 so set *f* (3) = 5. In this fashion we get *f* (4) = 2

Graph Isomorphism -Example

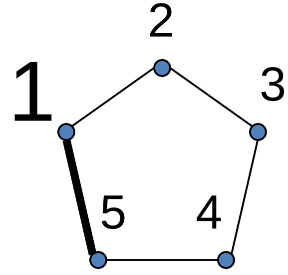
Next, set $f(1) = 1$ and try to walk around clockwise on the star. The next vertex seen is 3, *not* 2 so set $f(2) = 3$. Next vertex is 5 so set $f(3) = 5$. In this fashion we get $f(4) = 2$, $f(5) = 4$.

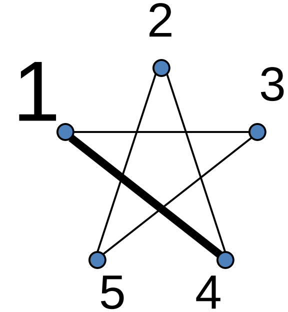




Graph Isomorphism -Example

Next, set $f(1) = 1$ and try to walk around clockwise on the star. The next vertex seen is 3, *not* 2 so set $f(2) = 3$. Next vertex is 5 so set $f(3) = 5$. In this fashion we get $f(4) = 2$, $f(5) = 4$. If we would continue, we would get back to $f(1) = 1$ so this process is well defined and f is a morphism.

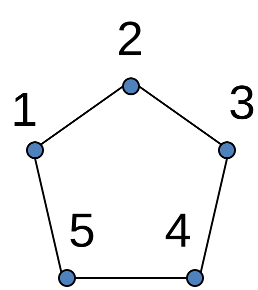


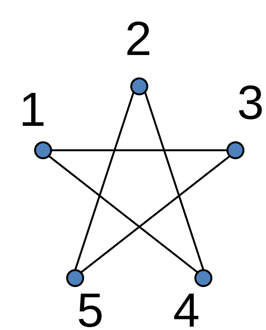


Graph Isomorphism -Example

Next, set *f* (1) = 1 and try to walk around clockwise on the star. The next vertex seen is 3, *not* 2 so set *f* (2) = 3. Next vertex is 5 so set *f* (3) = 5. In this fashion we get *f* (4) = 2, *f* (5) = 4. If we would continue, we would get back to *f* (1) =1 so this process is well defined and *f* is a morphism. Finally since *f* is bijective, *f* is an isomorphism.

f {1,2,3,4,5} → {1, 3, 5, 2, 4}





同构的图之间的特征

由于图完全由它的结点和边决定，所以同构的图之间具有相同的形状结构，必然具有**相同的一切内在性质**，所有的内在不变性都一样。

Isomorphic graphs must have the same intrinsic properties(invariant properties， 内在的不变性 ）

凡是那些不会因为图的画法不同发生变化的特征，或者说即便重画图也不会发生变化的那些特征（内在的不变性 ），都是一样的。

Isomorphic graphs have the same…

…number of vertices and edges

…degrees at corresponding vertices

…types of possible subgraphs

…any other property defined in terms of the basic graph theoretic building blocks!

…If one is bipartite (complete), the other one must be.

…etc. There is more about path

Isomorphims理解同构和意义

在同构的图之间有：

- Any approach/solution used on one, it fits the other as well.
- So for whatever purpose, whenever we know something on one, it could be applied on the other one which is isomorphic
    - *That is why "isomorphic " is important!*
- It is impossible and not necessary to repeat the same research on each of the graphs!
- Unfortunately, it is very difficulty to find out whether the two given graphs are isomorphic or not.

Isomorphic in math

- 数学中的同构是非常重要而且常见的概念
- 代数学中有“代数系统”的同构：指的是代数结构的相同
- 拓扑学中有拓扑同构：拓扑结构的相同
- 计算机领域也有所谓的同构和异构
- 就数学理解而言，同构的系统或结构之间，除了符号（代号）的差异，代表的具体对象和含义可能的差异外，结构方面和有关性质方面都是一样的，没有区别。
- 也可以说实质是一样的，都可以在抽象的意义上（数学意义上）视同，也就是说抽象地认为是相同的，至少是所关心的问题的方方面面是相同的。

不同构的例子-Negative Examples

Once you see that graphs are isomorphic, easy to prove it
一旦知道某两个图是同构的，证明起来一般不是太难。

但要证明两个图不同构，就不是那么简单了。
Proving the opposite, is usually more difficult. To show that two graphs are non-isomorphic need to show that no Function exists that satisfies defining properties (invariant properties) of isomorphism.

在实践中，我们可以去寻找不一样的内在特性，从而作出不同构的判断In practice, you can try to find some intrinsic property that differs between the 2 graphs in question.

Graph Isomorphism
-Negative Examples

Q: Why are the following non-isomorphic?

A: 1st graph has more vertices than 2nd.

Graph Isomorphism -Negative Examples

Q: Why are the following non-isomorphic?

A: 1^{st} graph has more edges than 2^{nd}.

Graph Isomorphism -Negative Examples

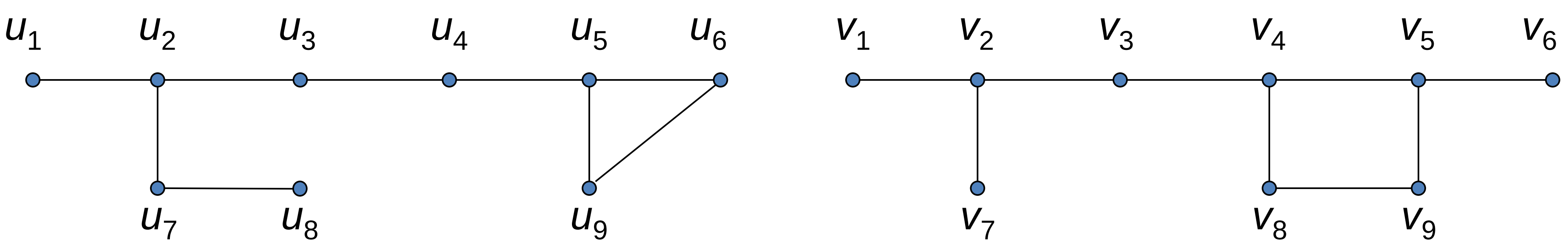
Q: Why are the following non-isomorphic?

A: 2nd graph has vertex of degree 1, 1st graph doesn't.

Graph Isomorphism -Negative Examples

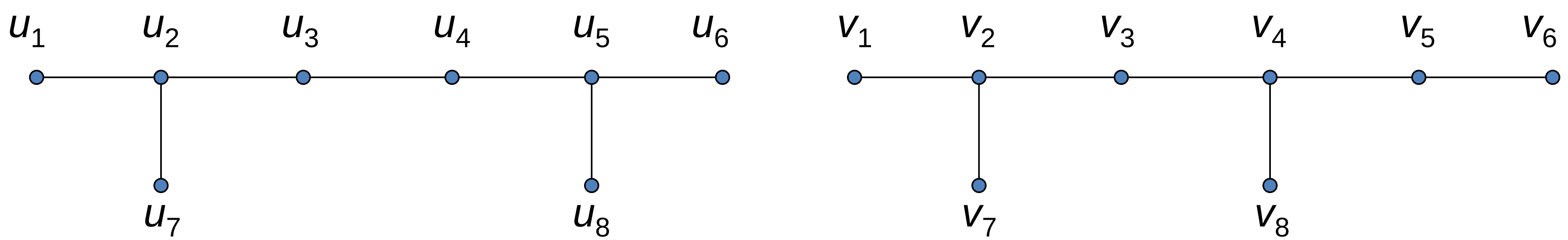
Q: Why are the following non-isomorphic?

A: 1st graph has 2 degree 1 vertices, 4 degree 2 vertex and 2 degree 3 vertices. 2nd graph has 3 degree 1 vertices, 3 degree 2 vertex and 3 degree 3 vertices.



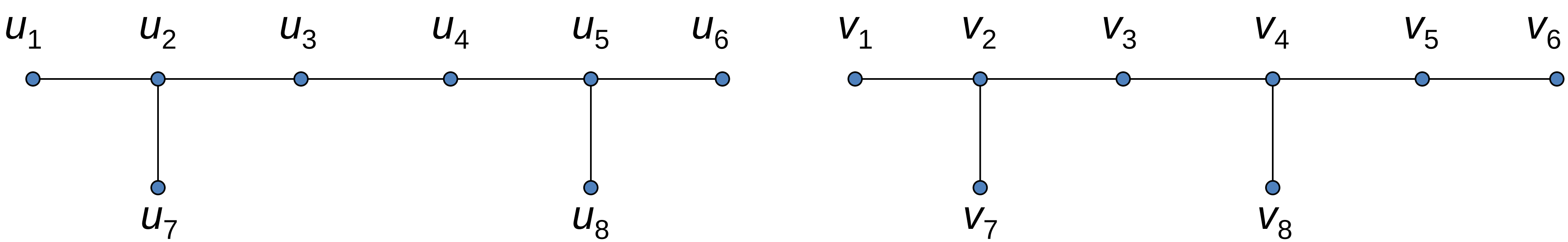
Graph Isomorphism
-Negative Examples

Q: Why are the following non-isomorphic?



Graph Isomorphism-Negative Examples

You can see: None of the previous approaches work as there are the same no. of vertices, edges, and same no. of vertices per degree.



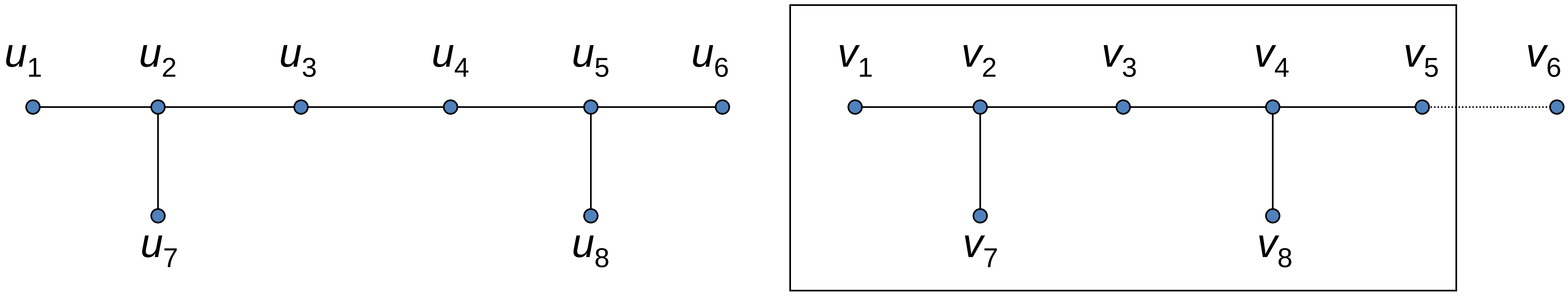
LEMMA: If *G* and *H* are isomorphic, then any

subgraph of *G* will be isomorphic to some subgraph of *H*.

Solution: Find a subgraph of 2nd graph which isn't a subgraph of 1st graph.

Graph Isomorphism -Negative Examples

A: This subgraph is not a subgraph of the left graph.

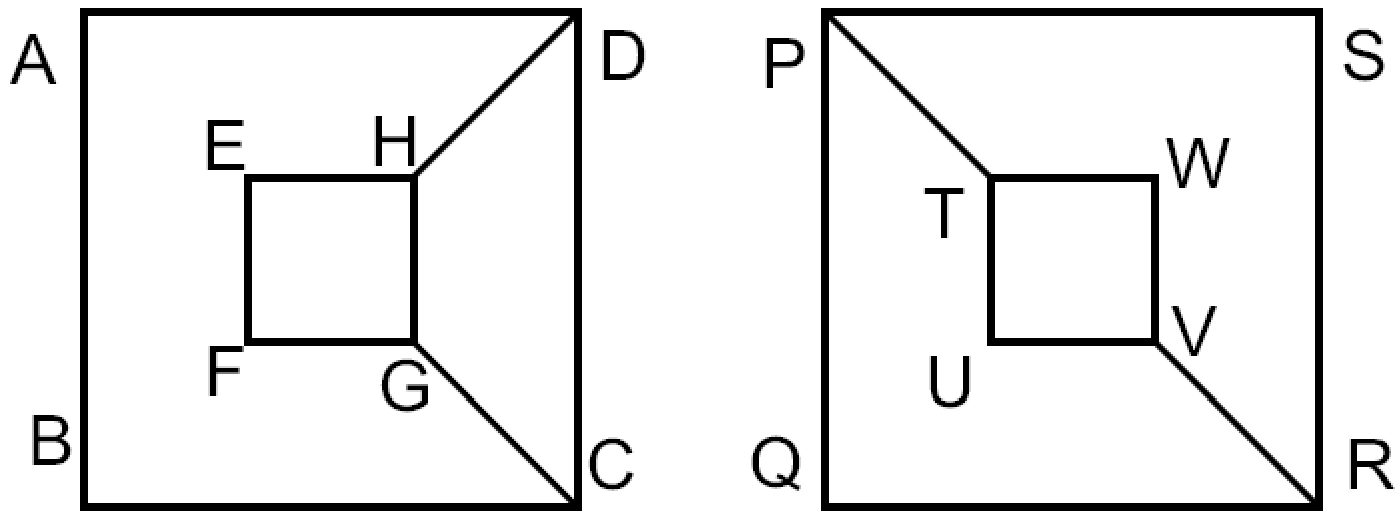


Why not? Deg. 3 vertices must map to deg. 3 vertices. Since subgraph and left graph are symmetric, can assume v_2 maps to u_2. Adjacent deg. 1 vertices to v_2 must map to degree 1 vertices, forcing the deg. 2 adjacent vertex v_3 to map to u_3. This forces the other vertex adjacent to v_3, namely v_4 to map to u_4. But then a deg. 3 vertex has mapped to a deg. 2 vertex→←

还可以考察两个度为3的结点间的距离

Isomorphism

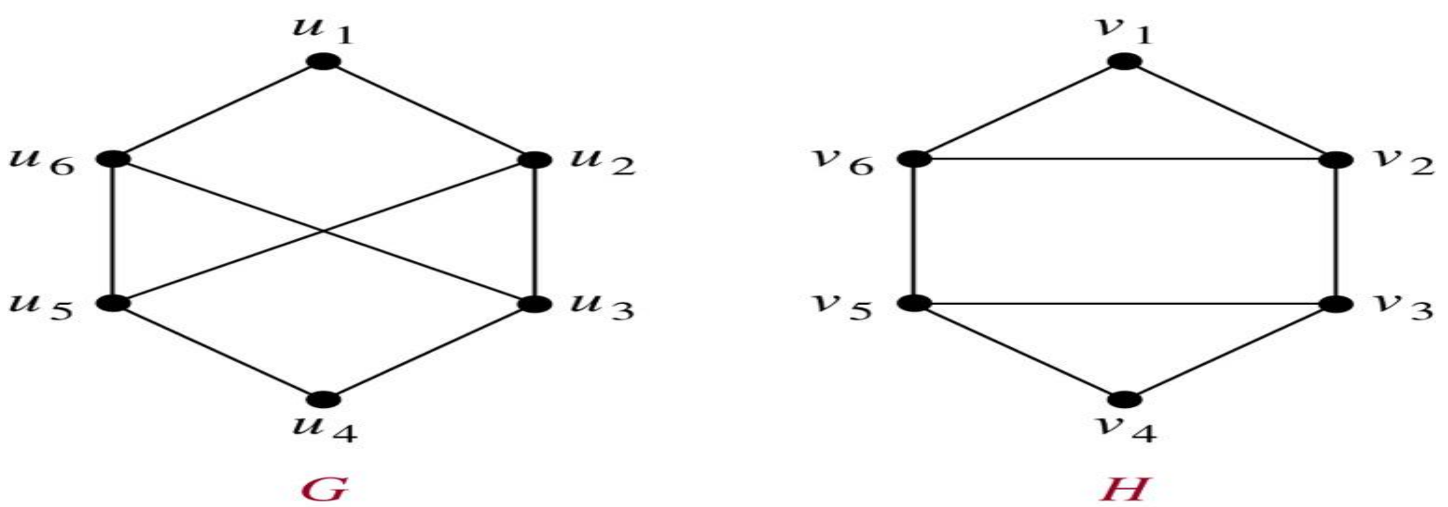
- Could you tell whether these following two graphs are isomorphic or not?



They are not isomorphic because the subgraphs defined by the four vertices of degree 3 are not isomorphic.
Or: there is a cycle with four vertices of degree 3, but not on the right.

Paths and Isomorphism路与图同构

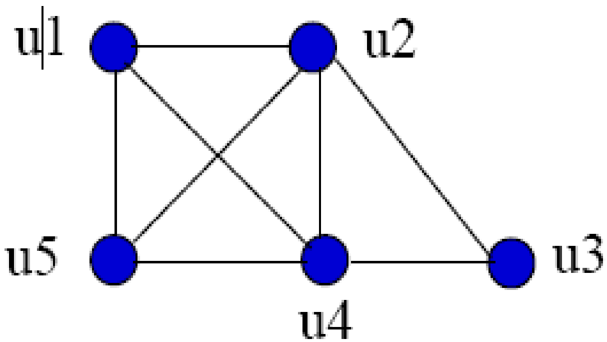
- Mentioned in previous section.
- Isomorphic graphs must have 'isomorphic' paths. E.g: if one has a simple circuit of length k then so must the other. Compare the following two graphs to see whether they are isomorphic.



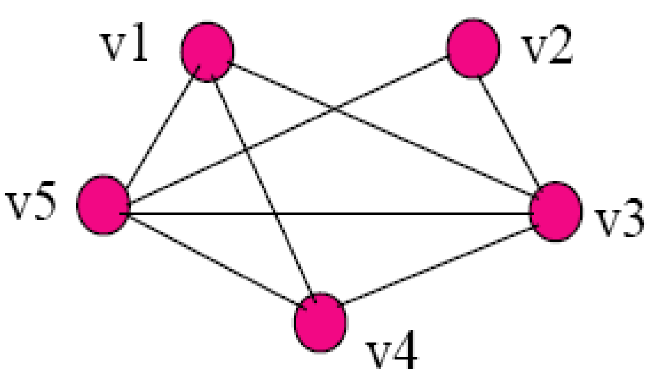
There is simple circuit of length 3 in H, but no in G.

Question about Isomorphism

- 问题：Can you determine whether two given graphs are isomorphic based on their adjacency matrices? Is it possible? If yes, how to?
- Example:



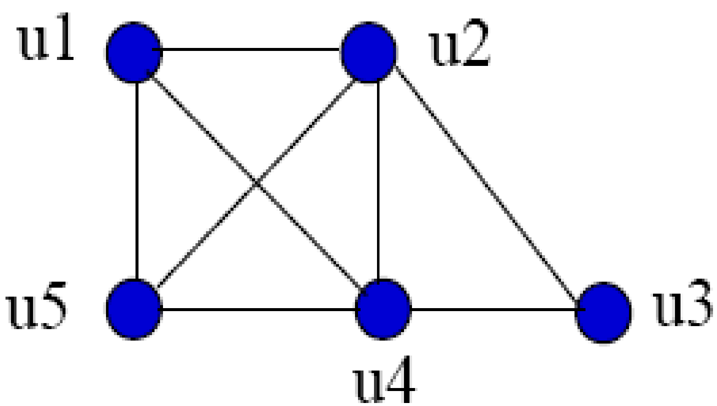
$$G1 = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \end{bmatrix}$$



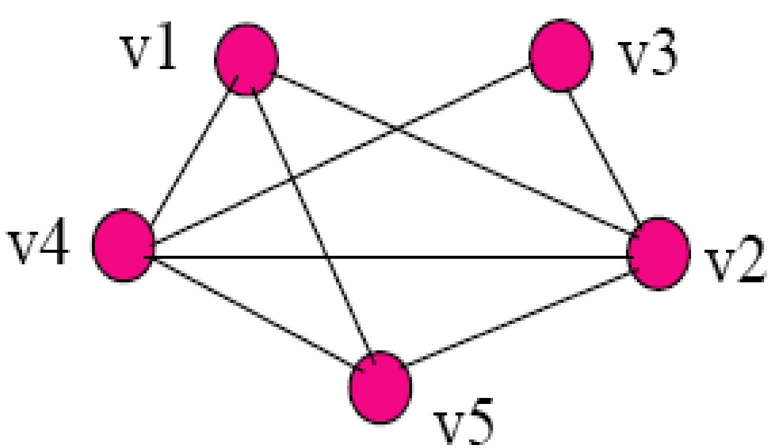
$$G2 = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{bmatrix}$$

Solution of the last example

- change the labels of the graph *G2 to produce the* graph *G2* according to the above permutation and* recalculate the adjacency matrix.



$$G1 = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \end{bmatrix}$$



$$G2^* = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \end{bmatrix}$$

Question about Isomorphism

- Observation: Doing these relabeling by hand is a bummer!

Exercises

- 6.3节
- T13,T17,T41,T43,T53

Solution to the "Crossing River"

- Note: There are 16 combinations of (P,W,L,C), but only 10 status are possible.