Paths and Connectivity
路与连通性

Connectivity (连通性）

无向图中的连通性：

DEF: Let *G* be a pseudograph(undirected 无向). Let *u* and *v* be vertices.

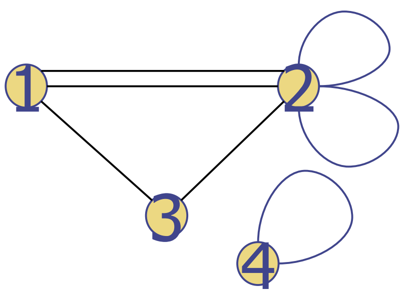
u and *v* are ***connected*（连接的）** *to each other* if there is a path in *G* between *u* and *v*.

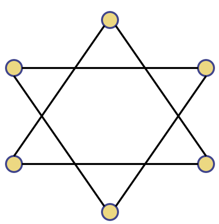
G is said to be ***connected*** (连通的) if all vertices are connected to each other *(pair-wise connected)*. An undirected graph that is not *connected* is called *disconnected*.

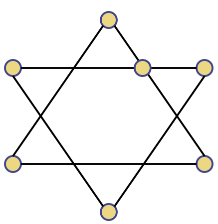
1. Note: Any vertex is automatically connected to itself via the empty path.
2. Note: 后面会有相应的有向图的连通性定义

Connectivity 连通性

Q: Which of the following graphs are connected?

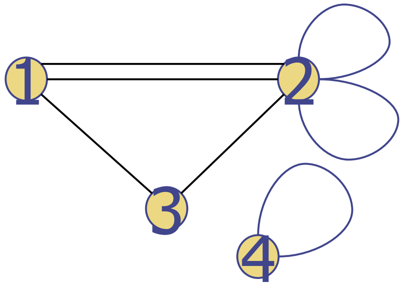


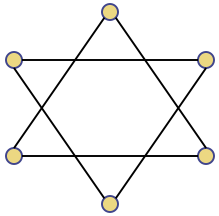


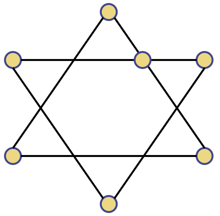


Connectivity

A: The first and second are disconnected. The last is connected.

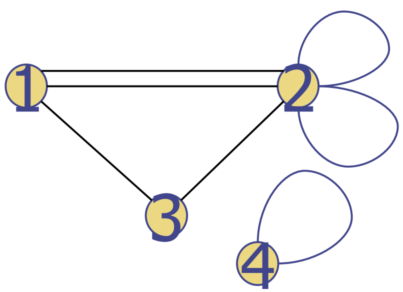


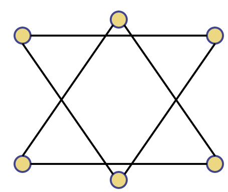


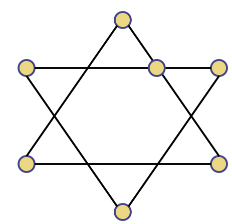


Connectivity

A: First and second are disconnected. Last is connected.

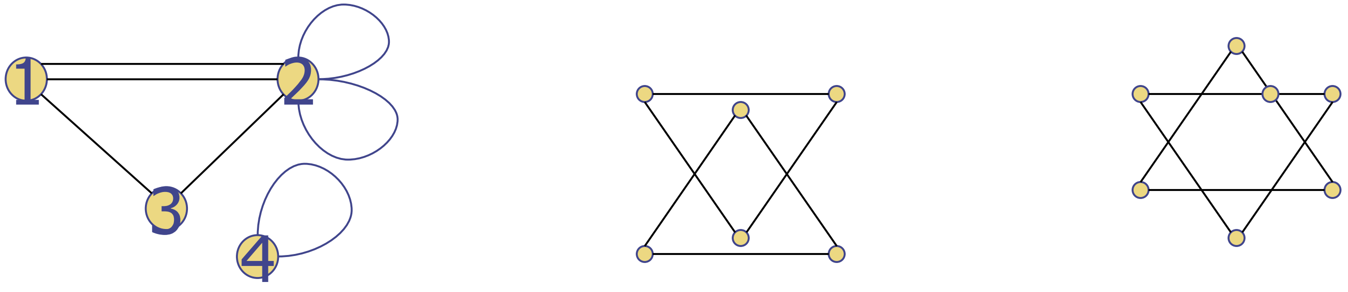






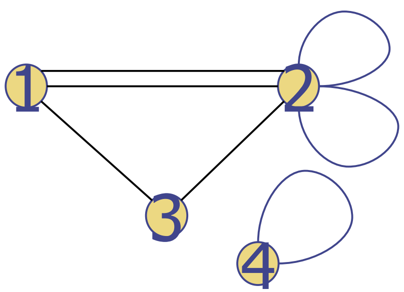
Connectivity

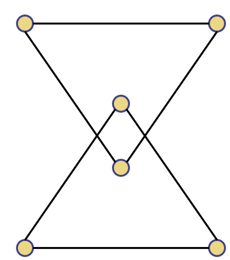
A: First and second are disconnected. Last is connected.

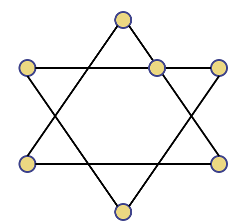


Connectivity

A: First and second are disconnected. Last is connected.

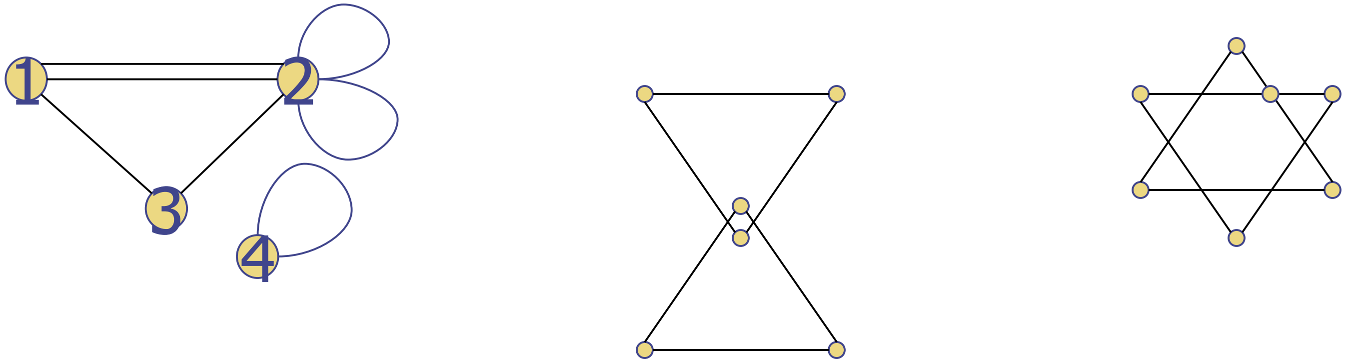






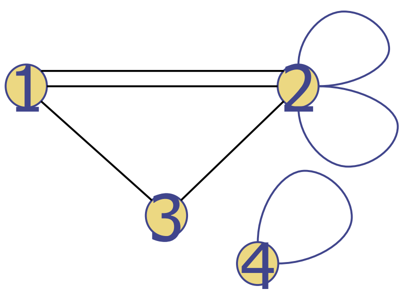
Connectivity

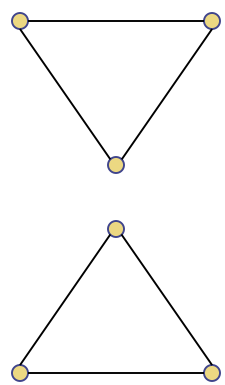
A: First and second are disconnected. Last is connected.

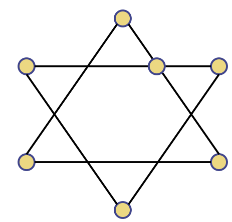


Connectivity

A: First and second are disconnected. Last is connected.







Questions about connectivity

- Question: in a computer network with n terminals, how can the computers communicate message each other?

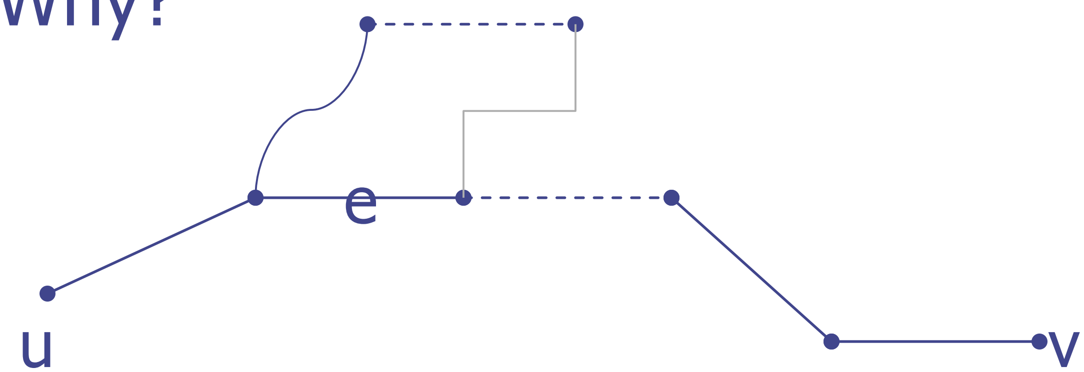
- Question: in a transportation network, can a person get another place from any one of the places?

Connectivity连通性定理

- Theorem: There is a simple path between every pair of distinct vertices of a connected undirected graph.

无向连通图中的任何两个不同的结点之间都一定有一条简单路。

- Why?



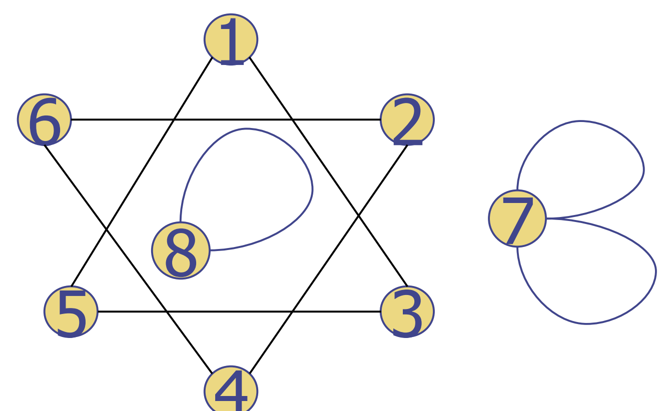
Connected Components
连通分支 or 分图

DEF: 一个连通分支 in a graph *G* is a subgraph of G such that all its vertices in this subgraph are connected to each other and every possible connected vertex is included in this subgraph.

- Or: **the maximally connected subgraphs of G 最大连通子图 (**that is not a proper subgraph of another connected subgraph of *G*. **)**

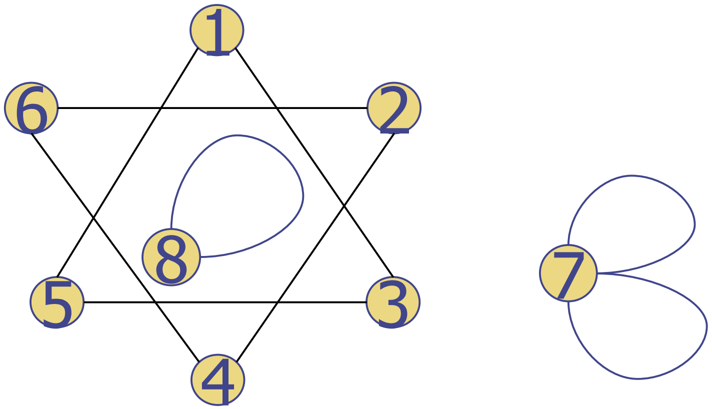
Q: What are the connected components of the following graph?

注：图的每一个结点都必然会在其中的一个分支中。与这个点连接的所有结点以及这些点关联的所有边形成一个子图，该子图就是所在的连通分支。



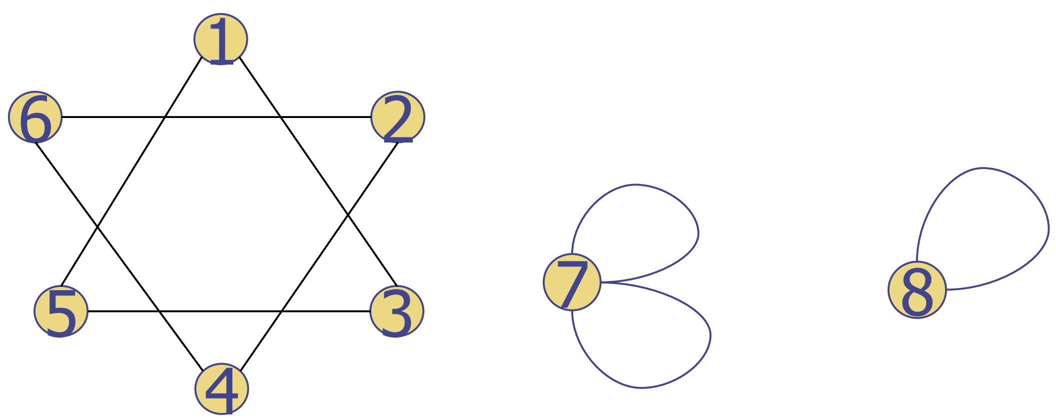
Connected Components

A: The components are {1,3,5},{2,4,6},{7} and {8} as one can see visually by pulling components apart:



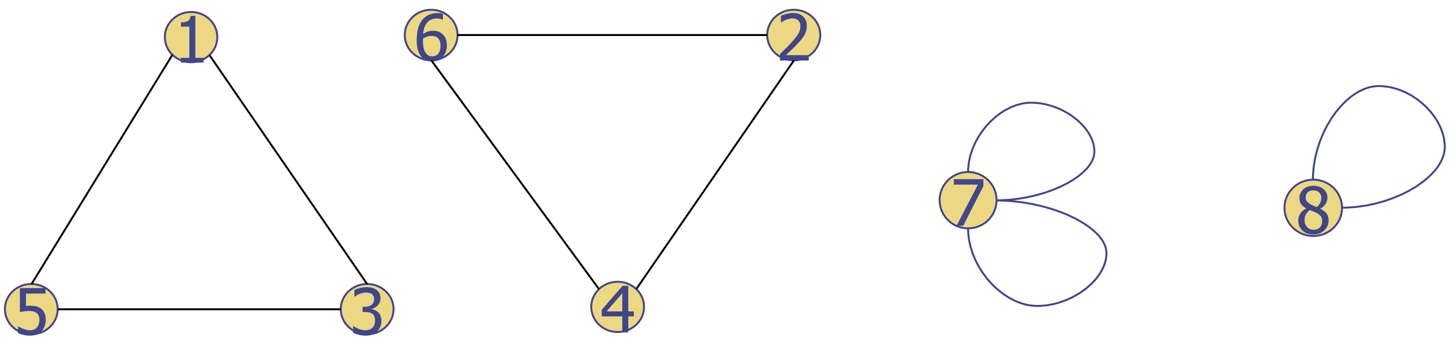
Connected Components

A: The components are {1,3,5},{2,4,6},{7} and {8} as one can see visually by pulling components apart:



Connected Components

A: The components are {1,3,5},{2,4,6},{7} and {8} as one can see visually by pulling components apart:



Question about Connected Components

- Is there any path between two different connected components? Why?

- How many components are there in a connected graph?

团及团问题（补充介绍）

定义（团）：假设G(V,E)是一个无向图，V′ 是V的一个非空子集，如果V′中任何两个不同的结点之间都是邻接的，也即都有直接的边连接起来，则称V′是一个团。

团中的结点数称为团的规模（或者说大小）

显然：团是G的一个完全子图。

有关团的两个著名的问题：

1：图G中是否存在指定正整数K的团？

2：寻找图中最大的团的优化问题称为“团问题”。

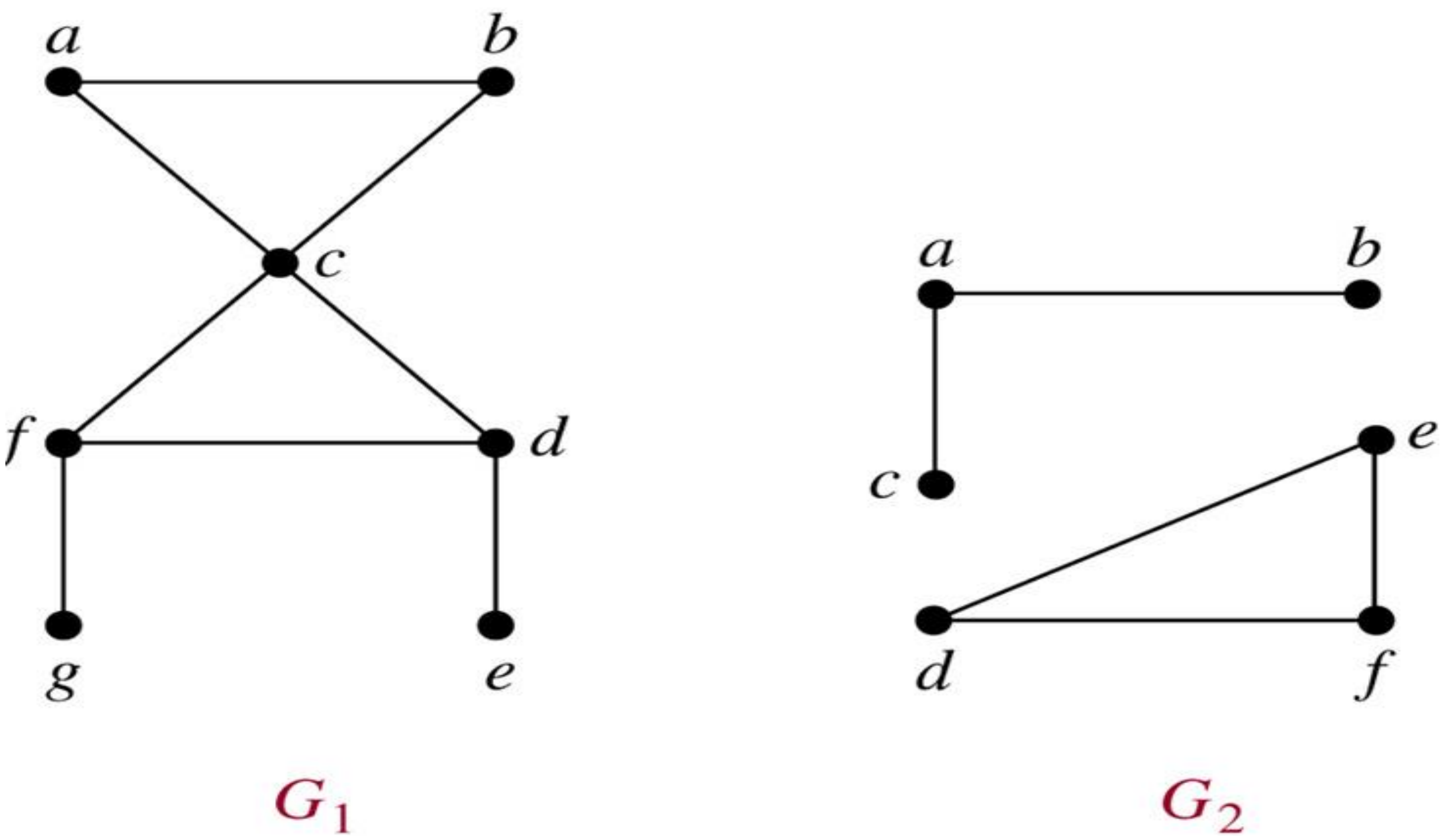
注：已经证明了“团问题”是一个NP完全问题，当然也是NP难问题。

cut vertex and cut edge
割边(弦,桥)和割点

- Def: 去掉该点就会导致图的分支数增加，那么这样的结点称为割点(cut vertex).
- Connected graphs without cut vertices are called **nonseparable graphs**（不可分割图）, and can be thought of as more connected than those with a cut vertex. (for example K_n)
- Def: 类似，去掉该边能导致分图数增加，这样的边称为割边 (**弦,桥**）

Examples

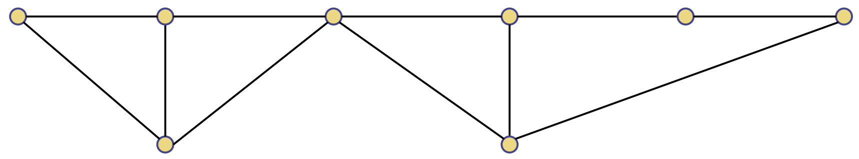
Please find out the cut vertices and cut edges from the following graphs:



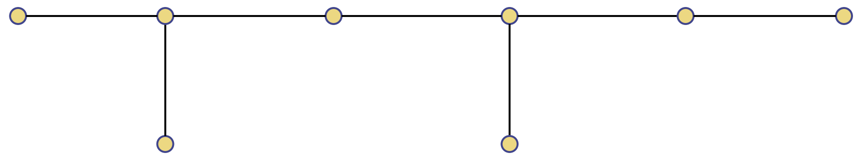
N-Connectivity N-连通

Q: Rate following graphs in terms of their design value for computer networks:

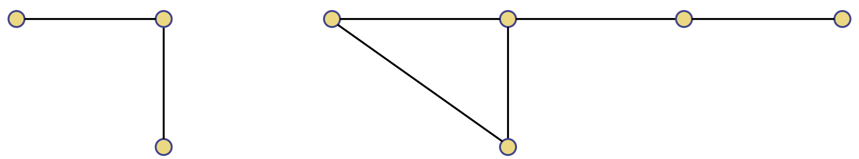
1)



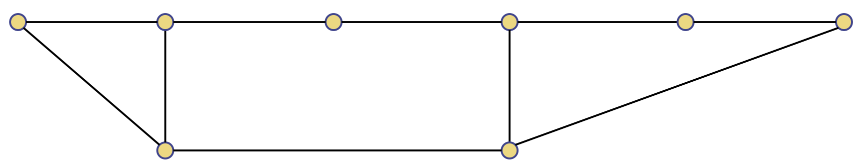
2)



3)



4)

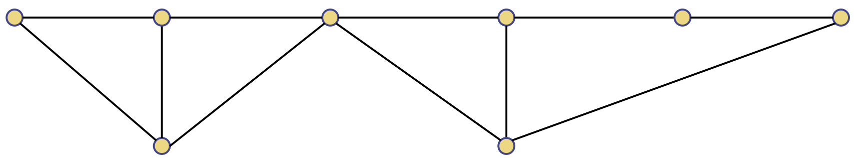


例子：可以把结点想象为路由器，边为连接路由器的链路。

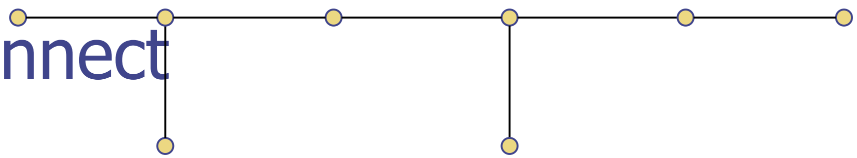
N-Connectivity

A: Want all computers to be connected, even if 1 computer goes down:

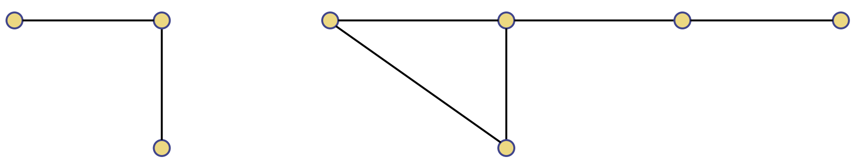
1) 2nd best. However, there's a weak link— "cut vertex"



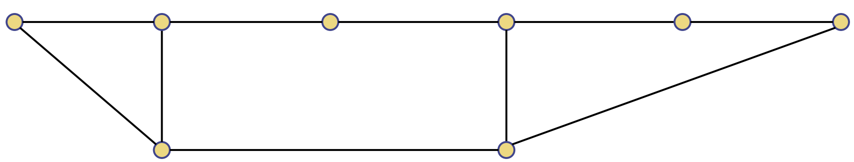
2) 3rd best. Connected but any computer could disconnect



3) Worst! Already disconnected

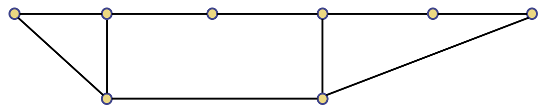


4) Best! Network dies only with 2 bad computers



- Think about the importance to build the redundant network (冗余的网络、冗余的链路)

N-Connectivity N-连通

The network  is best because it can only become disconnected when 2 vertices are removed. In other words, it is 2- connected.

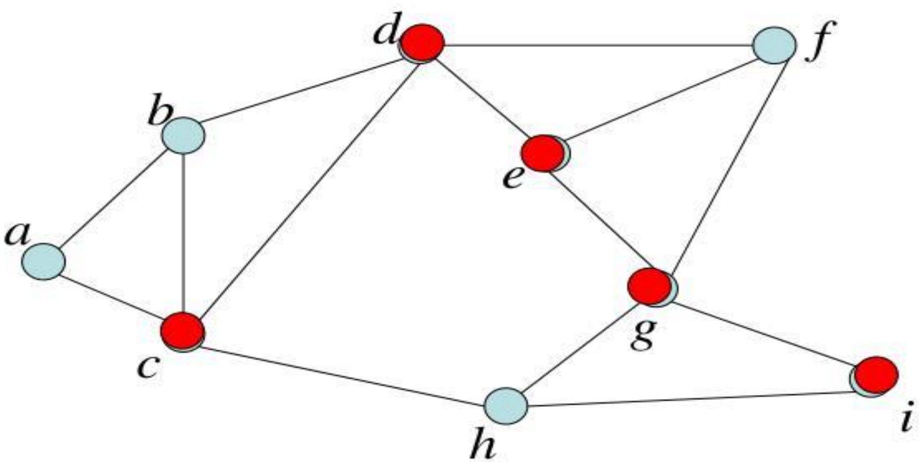
Formally:

DEF: 一个至少3个结点的简单连通图中，如果去掉任何一个点以及与之关联的边，仍然还是连通的，但如果去掉两个点就可能不再连通了，就称为

2***-connected*****。** ***或者说至少要去掉两个点才能不连通。***

Vertex Cut

- **Vertex Cut:** A separating set or vertex cut of a graph G is a set $S \subseteq V(G)$ such that G-S has more than one component.



A subset V′ of the vertex set V of a connected graph G = (V ,E) is a **vertex cut 点割集**, or **separating set**, if subgraph G − V′ is disconnected.

N-Connectivity N-连通

There is also a notion of *N*-Connectivity.

When G is a complete graph, it has no vertex cuts, because removing any proper subset of its vertices and all incident edges still leaves a complete graph

N-Connectivity (点连通度): a connected graph where we require at least *N* vertices to be removed to either disconnect the graph (or to be a graph with only one vertex),这里的N称为图的点连通度，as K(G). （注：图G中有k(G)个结点的点割集，但没有更小的点割集）

N-连通：The graph is called k-connected or k-vertex-connected when k ≤ K(G).

vertex Connectivity 点连通度

- The larger κ(G) is, the more connected we consider.
- Any disconnected graph G and complete graph K_1 has κ(G) = 0
- Any connected graph with vertex cut and K_2 have κ(G)=1
- Graph without cut vertices that can be disconnected by removing two vertices and K_3 have κ(G) = 2, and so on.
- For complete graph $K_{n:}$ $K(K_n) = n-1$
- 注：对于完全图由于没有点割集，我们定义其点连通度为删除的最少点数使其成为一个点。所有$K(K_n) = n-1$
- if G is non-separable and has at least three vertices. Note that if G is a k-connected graph, then G is a j-connected graph for all j with $0 \le j \le k$.

Edge Connectivity边连通度

- We can also measure the connectivity of a connected graph G = (V ,E) in terms of the minimum number of edges that we can remove to disconnect it.

边连通度：将一个连通图变成不连通图需要删除的最少的边数 the minimum number of edges that we can remove to disconnect connected graph G, denoted by λ(G).

- 这个定义适用于所有多余一个结点的连通图.
- If a graph has a cut edge, then we need only remove it to disconnect G.
- If G does not have a cut edge, we look for the smallest set of edges that can be removed to disconnect it.
- 边割集：A set of edges E′ is called an **edge cut边割集**of G if the subgraph G − E′ is disconnected.

点连通度和边连通度的不等式

When G = (V ,E) is a non-complete connected graph with at least three vertices, the minimum degree of a vertex of G is an upper bound for both the vertex connectivity of G and the edge connectivity of G.

$\kappa(G) \leq \lambda(G)$ when G is a connected non-complete graph.

$$\kappa(G) \leq \lambda(G) \leq \min_{v \in V} \deg(v).$$

注：这个部分更多的内容自己看教材。

Connectivity in Directed Graphs 有向图的连通性

In directed graphs may be able to find a path from *a* to *b* but not from *b* to *a*. So how to define directed？

Connectivity is non-obvious:

1) Should we ignore directions?
2) Should we insist that can get from *a* to *b* in actual digraph?
3) Should we insist that can get from *a* to *b* and that can get from *b* to *a as well*?

Connectivity in Directed Graphs

分情况定义：

1) ***Weakly connected*** *弱连接*: can get from *a* to *b* in underlying undirected graph

2) ***Strongly connected*** *强连接*: can get from *a* to *b* AND from *b* to *a* in the digraph

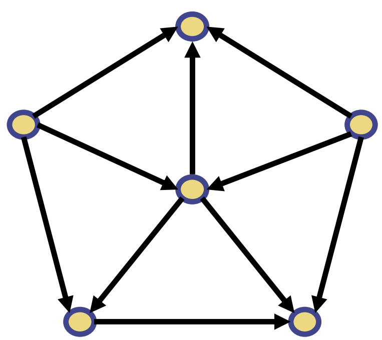
DEF: A graph is ***strongly*** connected if every pair of vertices is strongly connected.

A graph is ***Weakly*** connected if every pair of vertices is weekly connected.

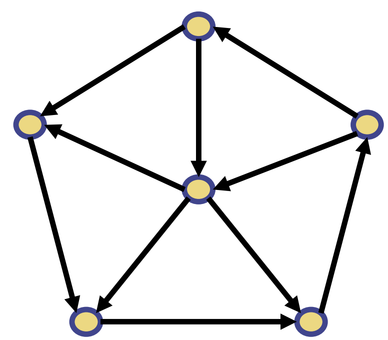
想像交通网络中有些是单行道的情况

有向图连通性

Q: Classify the connectivity of each graph.



weak



strong

The Connected Components of the Web Graph
WEB图的连通分支

想象一下给WEB网进行图建模。如果用结点来表示每个web page，用有向边来表示从一个page到另一个page的链接link，得到一个有向图。根据统计研究，

- 该图在1999年的时候就达到了将近两亿个结点，15亿条边； 到2010年，该网络图有了至少550亿个结点，1万亿条边；

其基本的无向图是不连通的；但是却包含了一个将近90%的结点的连通分支；

- 作为有向图，该图包含了一个巨大的强连通分支，和一些个较小的强连通分支(GSCC)；
- 想象一下，在一个强连通分支内部，从一个PAGE到任何一个其它的PAGE，都会怎么样？

Counting paths between vertices
结点间的路的数目

对图中任意给定的两个结点，思考如下问题：

Q1: are they connected? (or exists path between them)

Q2: how many paths between them?

Q3: which path is the shortest in a regular/weighed graph? (later on)

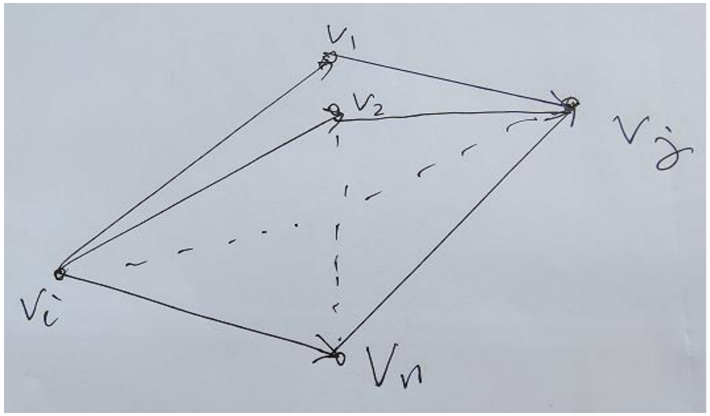
Counting paths between vertices结点间的路数

- **Theorem:** if M is the adjacency matrix of G, then the entry $(i, j)^{th}$ of M^r is the number of paths from i^{th} vertex to j^{th} vertex （或者是第i个结点与第j个结点之间）.

Note: Here is the standard power of M, not the boolean product （矩阵的普通乘积，非布尔积）.

This is a very useful and important theorem.

Proof:…讲解利用数学归纳法的证明思路



Counting paths between vertices

Proof by mathematical induction:

Basis Step: By definition of the adjacency matrix, the number of paths from v_i to v_j of length 1 is the (i,j)th entry of **A**.

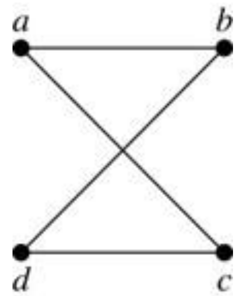
Inductive Step: For the inductive hypothesis, we assume that that the (i,j)th entry of $\mathbf{A}^r$ is the number of different paths of length r from v_i to v_j.

- Because $\mathbf{A}^{r+1} = \mathbf{A}^r\,\mathbf{A}$, the (i,j)th entry of $\mathbf{A}^{r+1}$ equals $b_{i1}a_{1j} + b_{i2}a_{2j} + \cdots + b_{in}a_{nj}$, where b_{ik} is the (i,k)*th* entry of $\mathbf{A}^r$. By the inductive hypothesis, b_{ik} is the number of paths of length r from v_i to v_k.
- A path of length $r + 1$ from v_i to v_j is made up of a path of length r from v_i to some v_k, and an edge from v_k to v_j. By the product rule for counting, the number of such paths is the product of the number of paths of length r from v_i to v_k (i.e., b_{ik}) and the number of edges from v_k to v_j (i.e, a_{kj}). The sum over all possible intermediate vertices v_k *is* $b_{i1}a_{1j} + b_{i2}a_{2j} + \cdots + b_{in}a_{nj}$.

Counting Paths between Vertices

Example: How many paths of length four are there from *a* to *d* in the graph G.

G



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

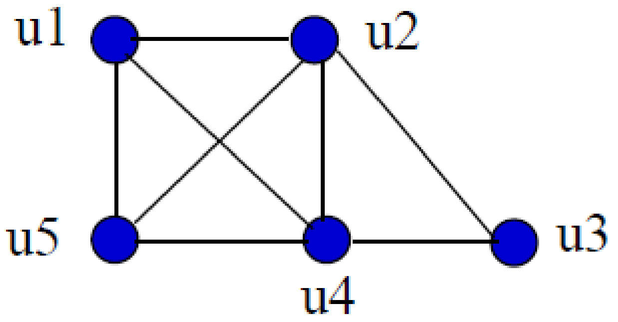
adjacency matrix of G

Solution: The adjacency matrix of *G* (ordering the vertices as *a*, *b*, *c*, *d*) is given above. Hence the number of paths of length four from *a* to *d* is the (1, 4)th entry of $\mathbf{A}^4$. The eight paths are as:

$$\mathbf{A}^4 = \begin{bmatrix} 8 & 0 & 0 & 8 \\ 0 & 8 & 8 & 0 \\ 0 & 8 & 8 & 0 \\ 8 & 0 & 0 & 8 \end{bmatrix}$$

$$\begin{array}{ll} a, b, a, b, d & a, b, a, c, d \\ a, b, d, b, d & a, b, d, c, d \\ a, c, a, b, d & a, c, a, c, d \\ a, c, d, b, d & a, c, d, c, d \end{array}$$

Counting paths--Example



$$M = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \end{bmatrix} \quad M^2 = \begin{bmatrix} 3 & 2 & 2 & 2 & 2 \\ 2 & 4 & 1 & 3 & 2 \\ 2 & 1 & 2 & 1 & 2 \\ 2 & 3 & 1 & 4 & 2 \\ 2 & 2 & 2 & 2 & 3 \end{bmatrix} \quad M^3 = \begin{bmatrix} 6 & 9 & 4 & 9 & 7 \\ 9 & 8 & 7 & 9 & 9 \\ 4 & 7 & 2 & 7 & 4 \\ 9 & 9 & 7 & 8 & 9 \\ 7 & 9 & 4 & 9 & 6 \end{bmatrix}$$

Further question 思考问题

- **Connected**: if there is a path from u to v (or between u and v), u and v are connected.
- （大家想想：什么时候是用from, 何时用between?）

- **Definition 距离**: The distance of two connected vertices u and v is the length of the number of edges of the shortest path from u to v (or between).

- Question: if vertex u and v are connected, is there shortest path between u and v?

Distance 距离

- **Theorem**: u and v are two different connected vertices of an undirected graph G, then there is a shortest path between u and v having length less than number of vertices of G.
- Why?
- **Question**: how to calculate the distance between any two different vertices using the adjacency matrix?

- Solution 解答...
- 考察M^1, M^2, ...M^n

回忆结论：结点间的路的数目

- **Theorem:** 如果M 是图G的邻接矩阵. 那么M^r 的(i, j) 项就是 从结点i到结点j的长度为r的路的数目. *(Could be proved using math induction)*

- *Note: This is the standard power of M, not the boolean product.*

Is a graph connected?
如何判断图的连通性

思考: how to find whether a graph is connected (or not) based on the adjacency matrix?

怎样根据邻接矩阵来判断一个无向图是否是连通的？

图的连通性判断方法

- 假设M是(n,m)图G的邻接矩阵，分别计算M^1,M^2, M^{n-1}. 然后考察路的情况。

- 进一步的问题：想想，能否想出办法用邻接矩阵判断出一个简单无向图G是否是偶图?

- 介绍连接矩阵的定义...

- 偶图判断定理：一个没有单边环的图为偶图的充要条件是任意的回路都是偶数长
- 充分性证明：

课外练习

- 6.4节
- T21，T43，T47(a)