Graph Algorithm I

Instructor: Shizhe Zhou

Course Code:00125401

terminology

- 链、迹、路是图论中三个相似的概念，分别如下:

1、链（chain or walk）：顶点和边交错出现的序列称为链，在序列中边的前后两个顶点正好是边的端点，序列的第一个顶点和最后一个顶点为链的端点，其余的点为内点。
2、迹（trail）：边互不相同的链称为迹。即迹中无重**边**。
3、路（path）：内部点互不相同的链称为路。即路中无重**点**。

闭链（迹、路）：两端点相同的链（迹、路）称为闭链（迹、路）。

从上面的定义知道，三者是有区别的，迹、路也是链，但链不一定是迹、路。同样的闭链、闭迹、闭路也是有相同和区别的。

闭链==闭合的链

4、顶点连通性（connected）:存在连接x到y的路，则称x到y是连通的。
因为链中肯定存在路，所以闭链是连通的，但不能说是环路，不过反过来说就可以了，即环路时连通的闭链

terminology

- 链： v_1到v_k的顶点序列， 通过边连接(v_1, v_2) (v_2, v_3)… (v_{k-1}, v_k)称为v_1到v_k的路径.

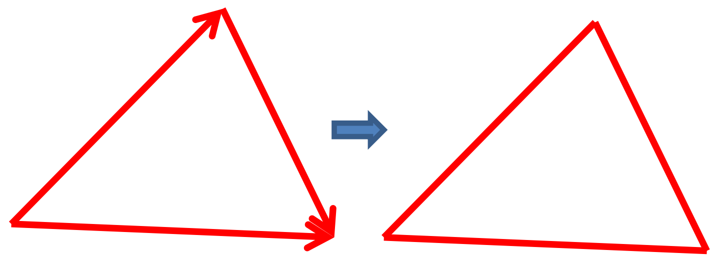
- 每个顶点仅出现一次的链：简单链， 或称路.

- 回路(环路/闭路): v_1=v_k的路径.

回路是简单的闭链, 闭链则不一定是简单的.

terminology

- G=(V,E).
- Undirected form

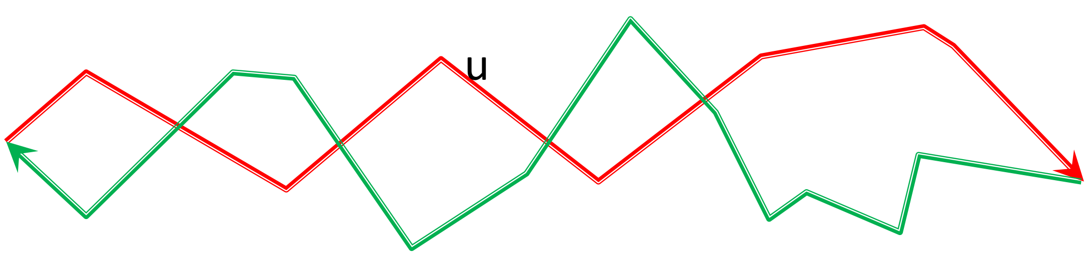


- 如果有向G的Undirected form中任何两个顶点有路径，则称G为弱连通的.
- 如果有向G中任何两个顶点u,v, 存在u到v的有向路径或者v到u有向的路径，则称G为连通的.

path(u,v)||path(v,u)

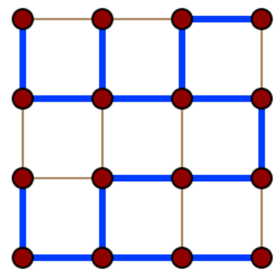
- 如果有向G中任2个顶点的对(u, v)和(v, u)都存在路径相连，则称G为强连通的.

path(u,v)&&path(v,u)



terminology

- H(U,F)是G=(V,E)的子图, 如果:$U \subseteq V, F \subseteq E$. G叫做H的超图(super graph).
  (subgraph)



- 对于连通无向图, 其生成树是一种子图，它的U=V.

- 对于非连通无向图, 其生成森林是一种子图，它的U=V.

- 导出子图(induced subgraph): H(U,F): <由原图的顶点集的一个子集导出的子图>

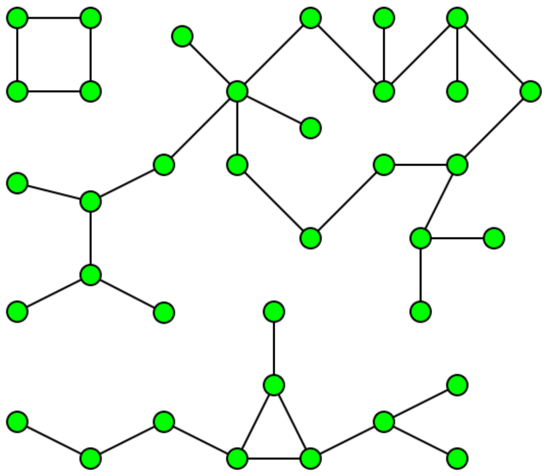
$$U \subseteq V, if\ (u,v) \in E, \text{且} u,v \in U, \text{则边}(u,v) \in F$$

- 生成子图(spanning subgraph): H(U,F):

$$U = V, F \in E$$

terminology

- 对无(有)向图G=(V,E)的一个子图H(U,F)，如果H是(强)连通的，且在G中不再与其他任何点相连通，则是G的一个(强)连通分支/量.
- 非连通图，有超过一个连通分支. ⇨

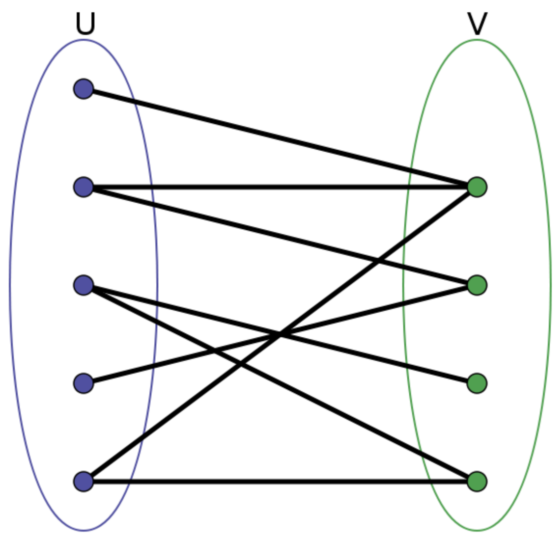


- 连通分支的数量是一个重要的拓扑不变量.

connected component search : a straightforward apps from DFS search.

terminology

- 偶图 Bipartite graph:无向图G =(V,E)的结点集V能够划分为两个子集V1,V2，满足V1∩V2 = F(空集)，且V1∪V2 = V，使得G中任意一条边的两个端点, 一个属于V1, 另一个属于V2，则称G为**偶图(Bipartite Graph)**或二分图(Bigraph). V1和V2称为互补结点子集, 偶图也可记为G = <V1,E,V2>。



欧拉图

问题 给定一个无向连通图 $G=(V,E)$，其所有顶点的度为偶数，寻找一个封闭路径 P，使得 E 中的每一条边在 P 中仅出现一次。

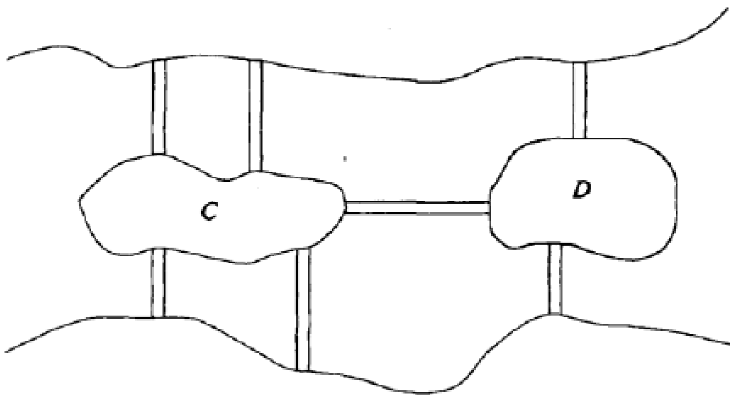
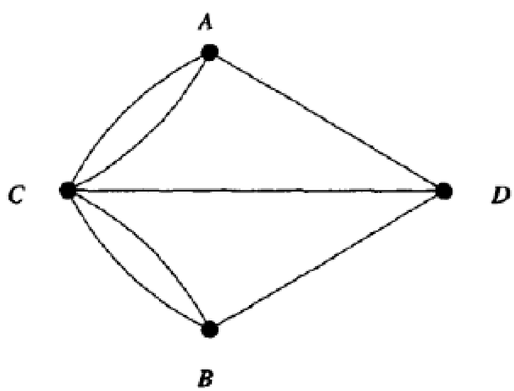


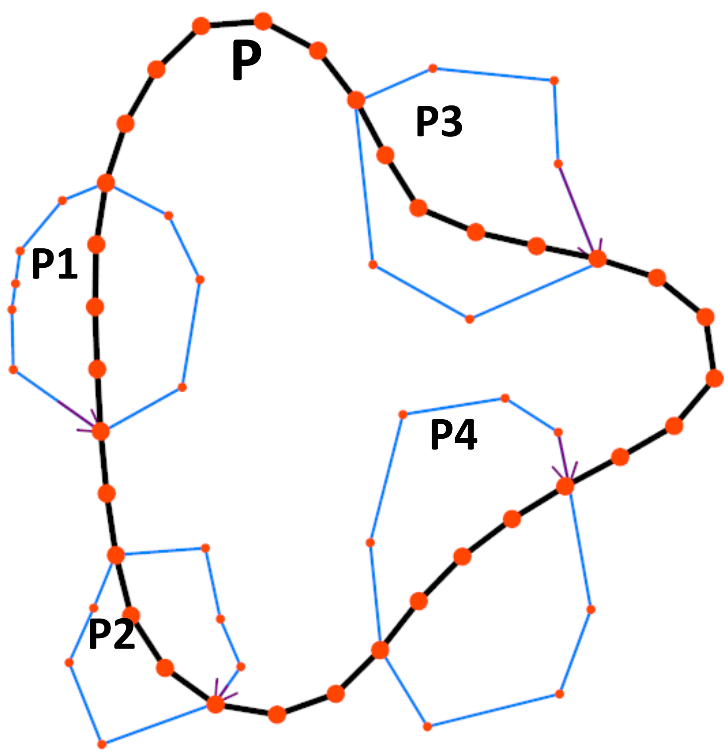
图 7.1 哥尼斯堡桥问题

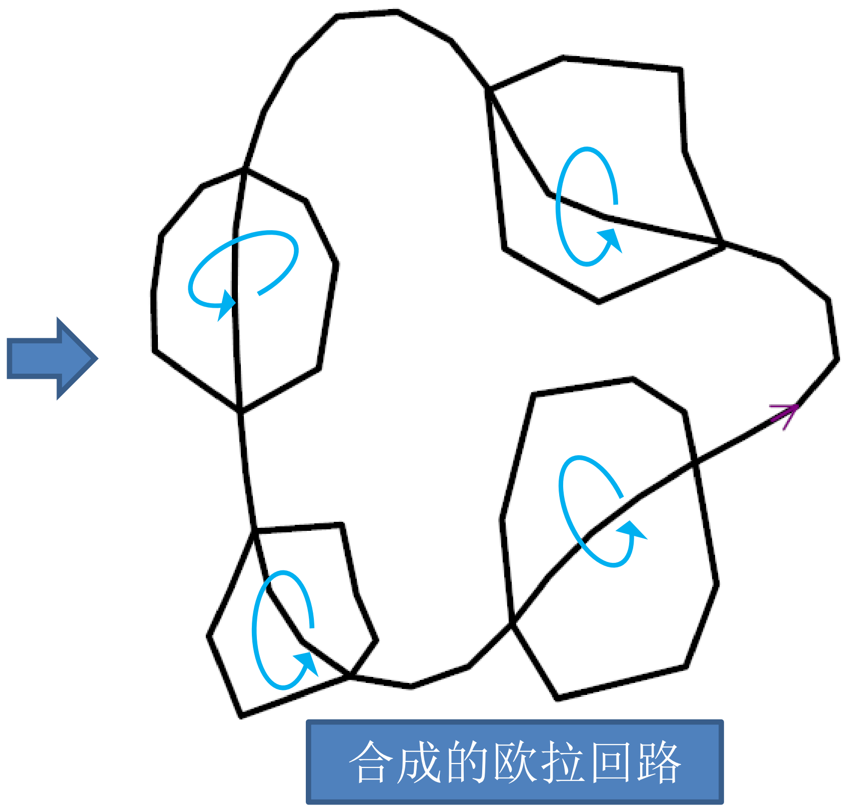


- 必要性：如果满足条件的P存在，则所有顶点的度为偶数.

欧拉图

- 归纳假设：对于边数小于 m 的连通图，其所有顶点的度为偶数，则存在一条包含每条边仅一次的封闭路径，并且我们知道如何找到这条路径。



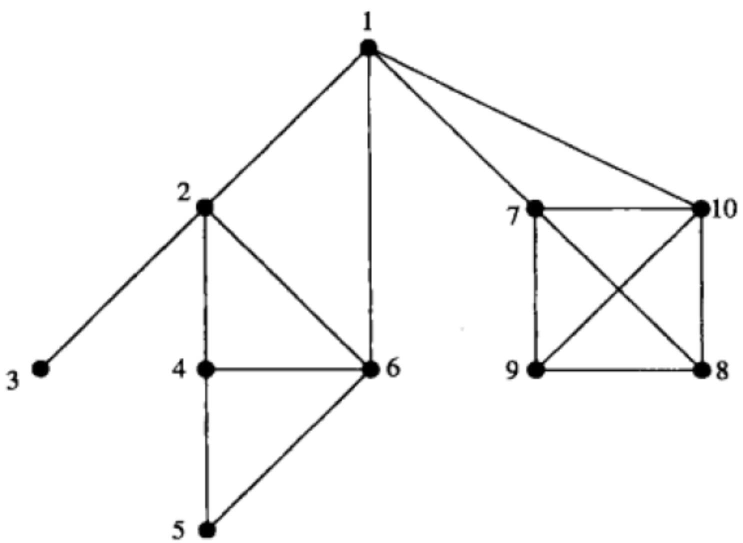


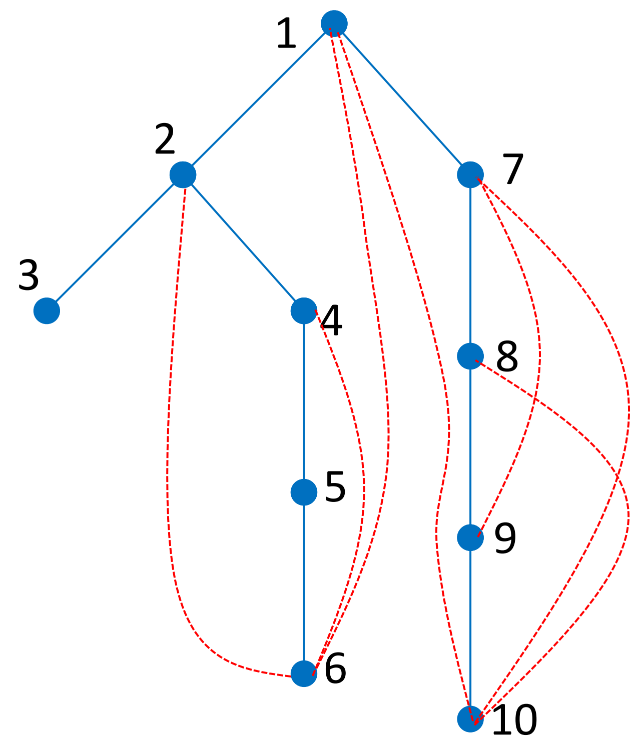
Scan the graph

- Traverse method:
  - Depth-first search.
  - Breath-first search.

Depth-first Search

- 无向图的DFS从任意一个节点开始，都能遍历所有的节点.



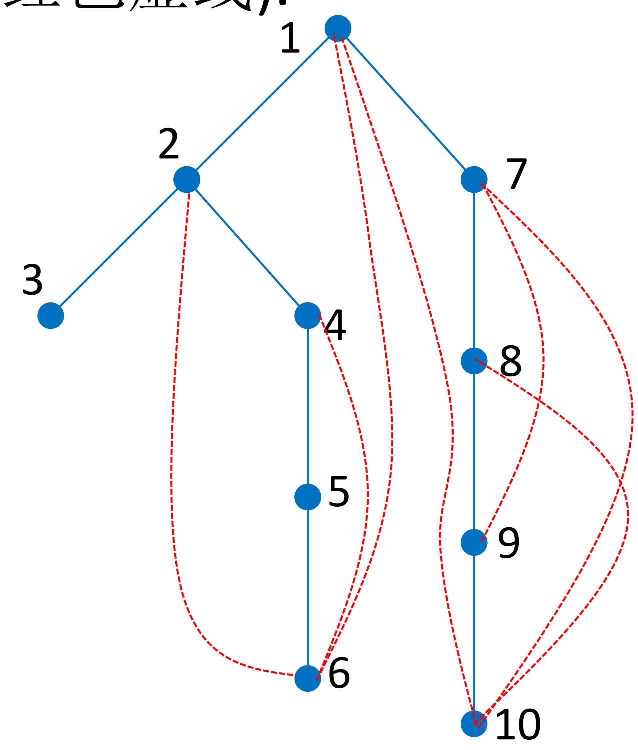


Visited:	1	1	0	0	....					

DFS tree

DFS-tree

- DFS-tree将E划分为两个集合E_T和E_B， 一个集合E_T是DFS-tree上的边，另一个集合E_B是连接DFS-tree上子孙节点和祖先节点的边(下图红色虚线).



Note: 一个DFS确定一个E_B, E_B是不唯一的。

DFS-tree

- 集合E_B不包含”横穿”DFS-tree的边, “横穿”表示连接的顶点在DFS树上不是祖先和后代的关系.
- 习题7.3解答: 对给定生成树T, 从G中减去T的边剩下的边集合必须是某种E_B, 因此检查剩下的边集合中每条边对于T是否满足的E_B属性即可.