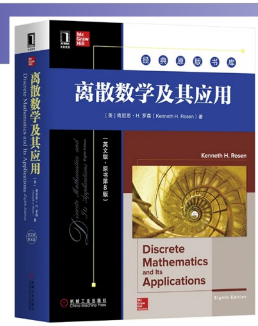
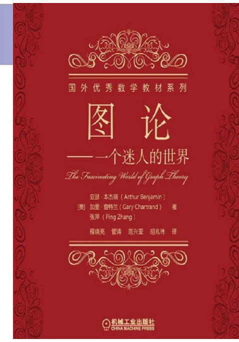
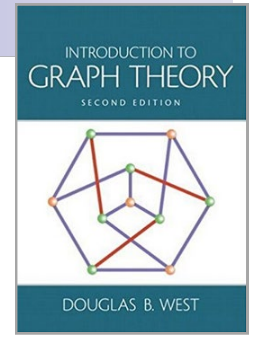
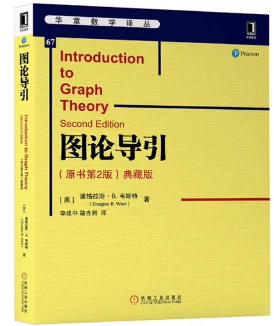
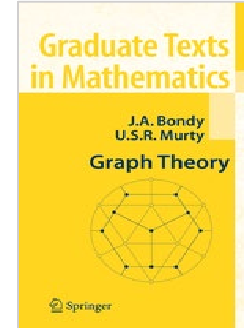










离 散 数 学 2

《图 论》

北航计算机学院

主要内容

1. 图论的基本概念
2. 子图和图的运算
3. 路径、回路和连通图
4. 欧拉图和哈密顿图
5. 图的矩阵表示
6. 树、有向树和有序树

第三章 路径、回路和连通性

知识点：路径、回路和连通性

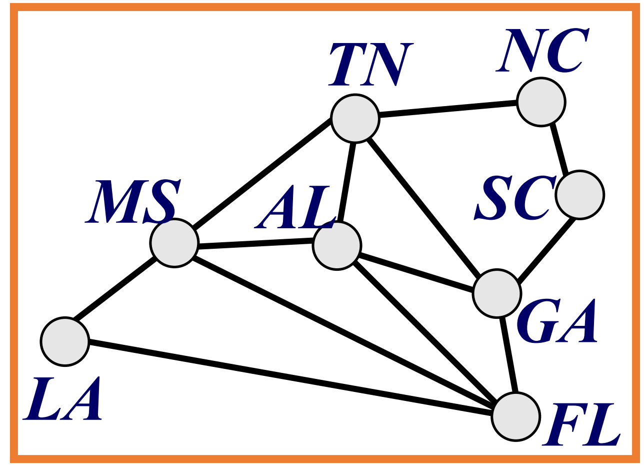
目的： 了解与路径、回路、连通性、分支、非循环图相关的基本概念；掌握求加权路径的算法、判一个图是否有回路、有有向回路、有半回路的过程；

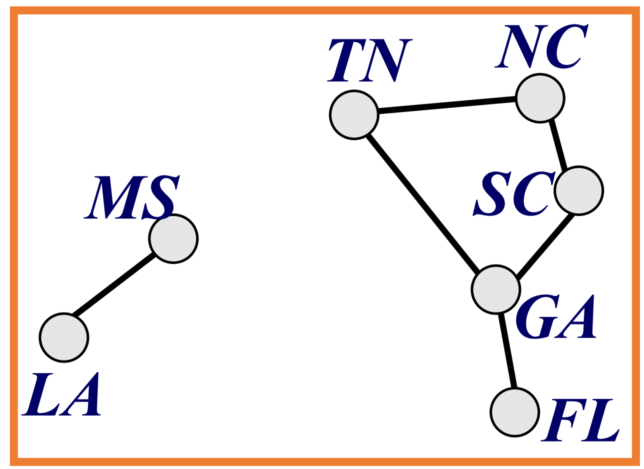
重点： 路径、回路、连通、分支等重要概念；求加权路径的算法；判回路、有向回路、半回路、循环图；

难点： 几种判定方法及其原理。

思考：路径、回路和连通性

- 定义3.1：**路径**
  - 有那几类特殊路径？三个定义
  - 有没有最**基本路径**？定理3.1
  - 路径有什么特性？定理3.2
- 定义3.2：**可达**（基于路径定义）
  - 基本路径和可达关系？定理3.3
- 定义3.3：**距离**（定量化）
  - 图的直径（定义3.4）
  - 算法：最短路径
- 定义3.6：**连通性**（图整体描述）
  - 有向图、无向图
  - **分支数、回路**
  - 算法：W算法





3.1 路径与回路

- **背景：路径的应用**
  - ✓ 无向图的结点和边分别表示城市和连接城市的双轨铁路。
  - ✓ 从城市 v_0 到城市 v_n 的 路径： 由一个结点和边组成的序列来表示：

$$v_0\, e_1\, v_1\, e_2\, v_2 \ldots v_{n\text{-}1}\, e_n\, v_n$$

其中，

- ✓ $e_i\,(1 \le i \le n)$ 表示连接城市的铁路；
- ✓ v_1， v_2， …， $v_{n\text{-}1}$ 表示途经的城市。

（1）路径的定义: 简单路径、基本路径

简单路径: 边不重复的路径

简单回路: 边不重复的回路

基本路径: 点不重复的路径

基本回路: 点不重复的回路

（1）路径的定义: 简单路径、基本路径

定义3.1 设 $n \in \mathbf{N}$，$v_0, v_1, \ldots$，v_n 是图G的结点，$e_1, e_2, \ldots, e_n$ 是图G的边，并且v_{i-1} 和v_i 分别是 e_i 的起点和终点（i=1, 2, …, n），则称序列 $v_0 e_1 v_1 e_2 \ldots v_{n-1} e_n v_n$ 为图 G 中从 v_0 至 v_n 的路径，n 称为该路径的长度。

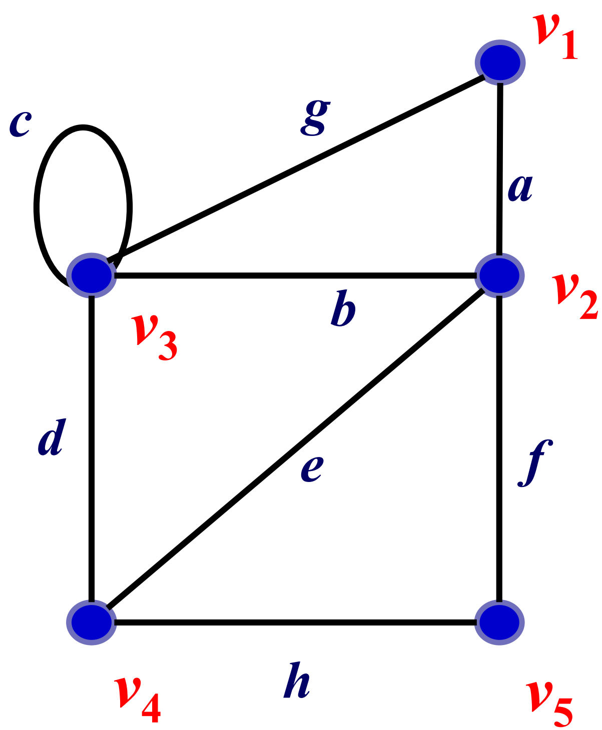
(1) 如果 $v_0 = v_n$，则称该路径为闭的，否则称为开的。

(2)如果 $e_1, e_2, \ldots, e_n$ 互不相同，则称该路径为简单的。

(3) 如果$v_0, v_1, \ldots, v_n$互不相同，则称该路径为基本的。

- 对无向图、有向图同样成立
- 基本路径必为简单路径

路径举例（无向图）



(1) $v_2\, b\, v_3\, d\, v_4\, e\, v_2\, b\, v_3$ 路径

(2) $v_2\, b\, v_3\, c\, v_3\, d\, v_4$ 简单路径

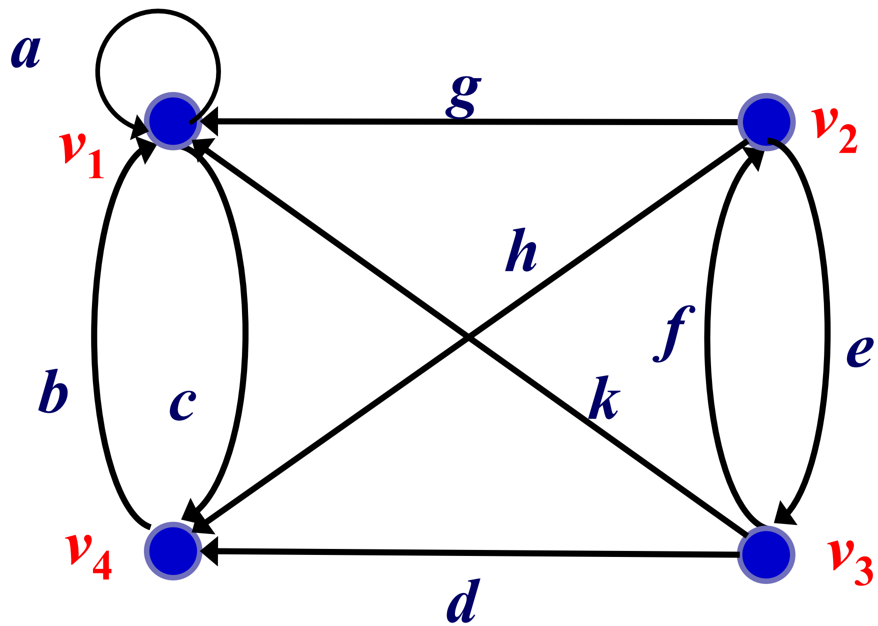
(3) $v_3\, c\, v_3\, c\, v_3$ 闭路径

(4) $v_1\, g\, v_3\, c\, v_3$ 变为一个基本路径？

基本路径：$v_1\, g\, v_3$

直观结论：从路径中去掉闭路径，能够得到基本路径。

路径举例（有向图）



(1) $v_1cv_4bv_1cv_4$ 路径

(2) $v_1av_1cv_4$ 简单路径

(3) v_1c_4 基本路径

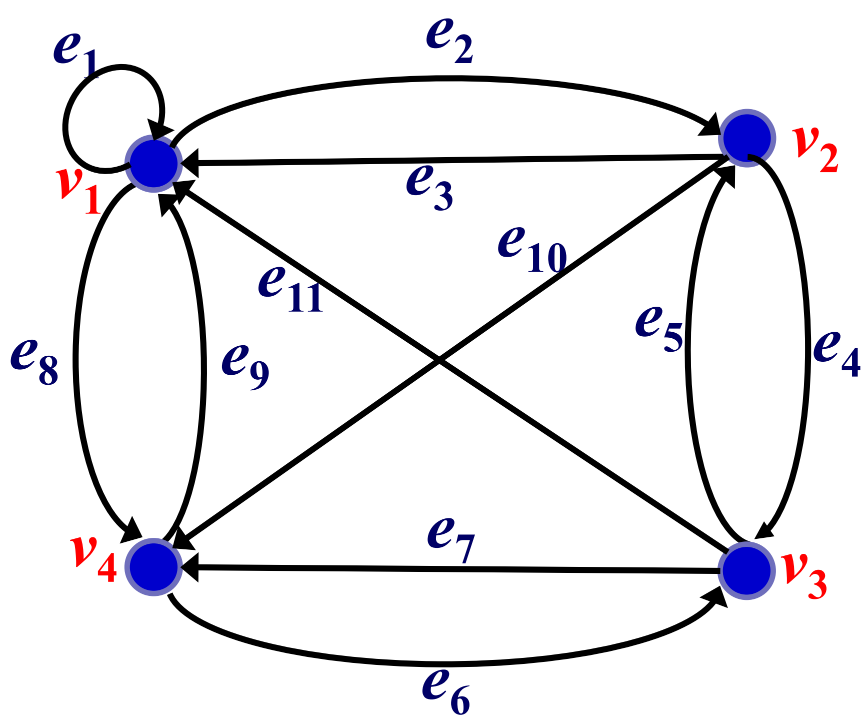
路径的基本性质

- 当 $n = 0$，路径 v_0 的长度为 0，基本路径。
  - 任何结点到自身总存在路径。
- v到 v' 存在路径 $\Rightarrow$ v' 到 v 存在路径?

  （无向图 √ 有向图 ╳ ）

定理3.1 设图 $G=<V, E, \Psi>$， $v, v'\in V$。如果存在从 v 至 v' 的路径，则存在从 v 至 v' 的基本路径。

路径上的结点互不相同

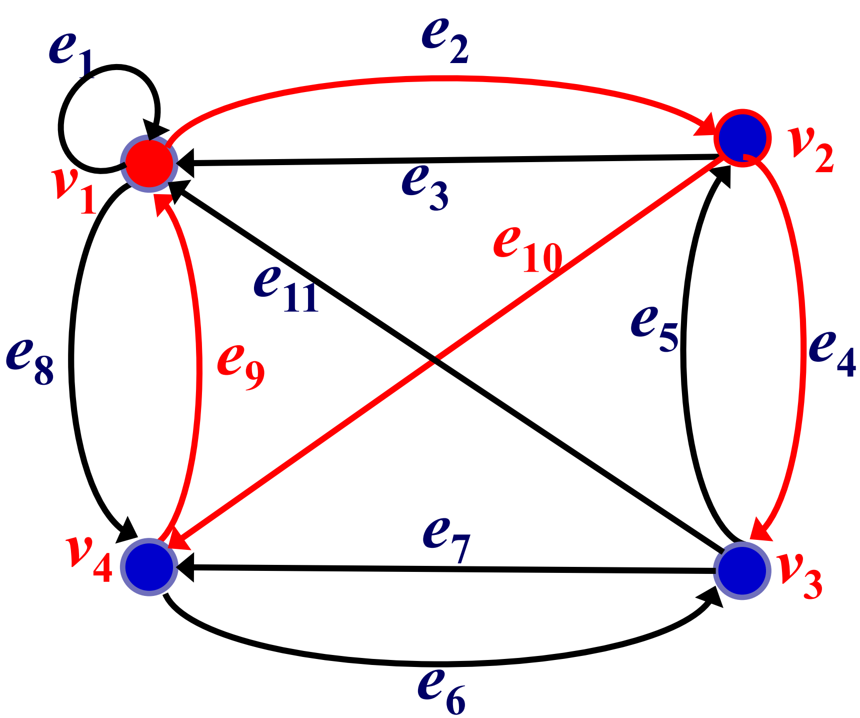


从v_1到v_3的路径：

$v_1\, e_2\, v_2\, e_{10}\, v_4\, e_9\, v_1\, e_2\, v_2\, e_4\, v_3$

定理3.1 设图 $G=<V, E, \Psi>$， $v, v' \in V$。如果存在从 v 至 v' 的路径，则存在从 v 至 v' 的基本路径。

路径上的结点互不相同

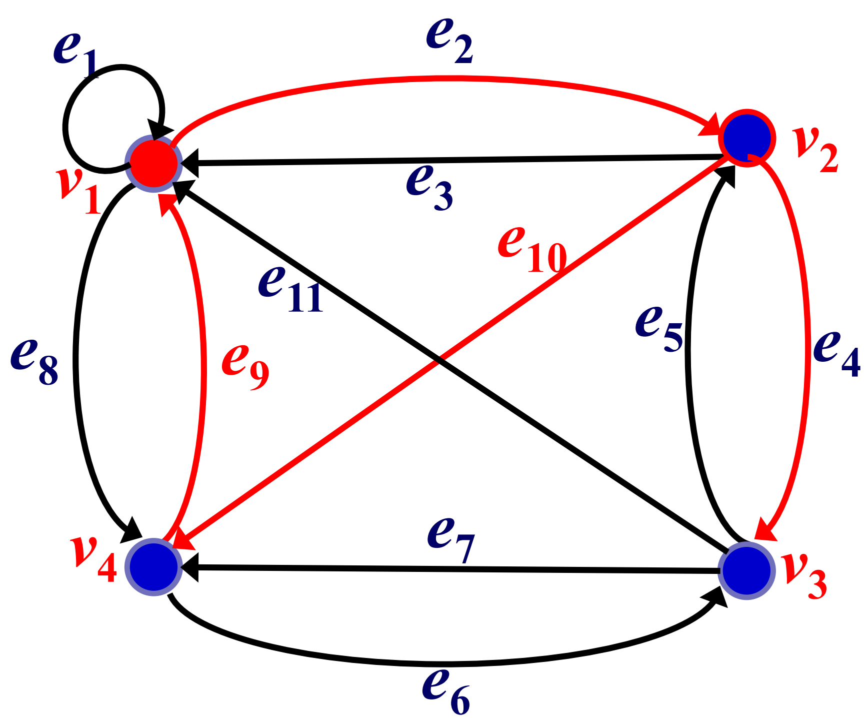


从v_1到v_3的路径：

$v_1\, e_2\, v_2\, e_{10}\, v_4\, e_9\, v_1\, e_2\, v_2\, e_4\, v_3$

定理3.1 设图 $G=\langle V, E, \Psi\rangle$， $v, v'\in V$。如果存在从 v 至 v' 的路径，则存在从 v 至 v' 的基本路径。

路径上的结点互不相同



从v_1到v_3的路径：

$v_1\, e_2\, v_2\, e_{10}\, v_4\, e_9\, v_1\, e_2\, v_2\, e_4\, v_3$

去掉回路

从v_1到v_3的基本路径：

$v_1\, e_2\, v_2\, e_{10}\, v_4\, e_9\, v_1\, e_2\, v_2\, e_4\, v_3$

定理3.1 设图 $G = \langle V, E, \Psi \rangle$， $v, v' \in V$。如果存在从 v 至 v' 的路径，则存在从 v 至 v' 的基本路径。

证明：（第二数学归纳法）

假设从 v 至 v' 的路径长度为n，对n进行归纳证明。

(1) 当$n=0$时，此时$v = v'$，路径长度为 0，是基本路径；

(2) 假设对每个自然数 k，当 $0 \le k < n$时，若存在长度为 k的从 v 到 v' 的路径，则一定存在从存在从v 至 v'的基本路径。

下面证明结论对$k=n$时成立。

定理3.1 设图 $G=\langle V, E, \Psi\rangle$， $v, v'\in V$。如果存在从 v 至 v' 的路径，则存在从 v 至 v' 的基本路径。

证明（续）：（第二数学归纳法）

下面证明结论对 $k=n$ 时成立。

假设路径 $\rho=v_0e_1v_1\ldots v_{n-1}e_nv_n$ 是从 v 至 v' 的路径，且不是基本路径， 其中 $v_0=v, v_n=v'$，则

必有 i 和 j 使 $0\le i<j\le n$ 且 $v_i=v_j$。

故 $\rho'=v_0e_1v_1\ldots v_ie_{j+1}v_{j+1}\ldots v_{n-1}e_nv_n$ 是从 v 至 v' 的长度为 $n-(j-i)$ 的路径。

由归纳假设知，必存在从 v 至 v' 的基本路径。

定理3.1 设图 $G = <V, E, \Psi>$，$v, v' \in V$。如果存在从 v 至 v' 的路径，则存在从 v 至 v' 的基本路径。

路径上的结点互不相同

- n 阶图有 n 个结点
- 基本路径上的结点互不相同
- n 阶图中的基本路径最多含有 n 个结点
- n 阶图中的基本路径最多含有 n-1条边

定理3.2 n 阶图中的基本路径的长度小于 n。

（2）可达

定义3.2 设图 $G=<V,E,\Psi>$，$v_1,v_2\in V$。

(1) 若存在从 v_1 至 v_2 的路径，则称在G中从v_1可达v_2，或从v_1到v_2可达；

(2) 否则称在G中从 v_1 不可达 v_2，或从v_1到v_2不可达。

对于图G的结点v，用 $R(v)$ 表示从v可达的全体结点的集合。

- 在无向图中，若从v_1到v_2可达，则从v_2到v_1必可达
- 在有向图中，从v_1到v_2可达不能保证从v_2到v_1必可达

定理3.3 设图 $G=<V,E,\Psi>$，$v_1,v_2\in V$。

从 v_1 可达 v_2 当且仅当存在从 v_1 至 v_2 的基本路径。

（3）距离与直径

定义3.3 设图 $G=\langle V, E, \Psi\rangle$， $v_1, v_2 \in V$。

(1) 若从 v_1 可达 v_2，则称从 v_1 至 v_2 的路径中长度最短者为从 v_1 至 v_2 的 测地线，并称该测地线的长度为从 v_1 至 v_2 的距离，记作 $d(v_1, v_2)$。

(2) 若从 v_1 不可达 v_2，则称 v_1 至 v_2 的距离 $d(v_1, v_2)$ 为 ∞。

并且规定:

$$\infty+\infty=\infty;\ \forall n\in \mathbf{N},\ \infty>n,\ n+\infty=\infty+n=\infty。$$

定义3.4 图 $G=\langle V, E, \Psi\rangle$ 的直径定义为

$$max_{v,v'\in V}\, d(v, v')$$

例

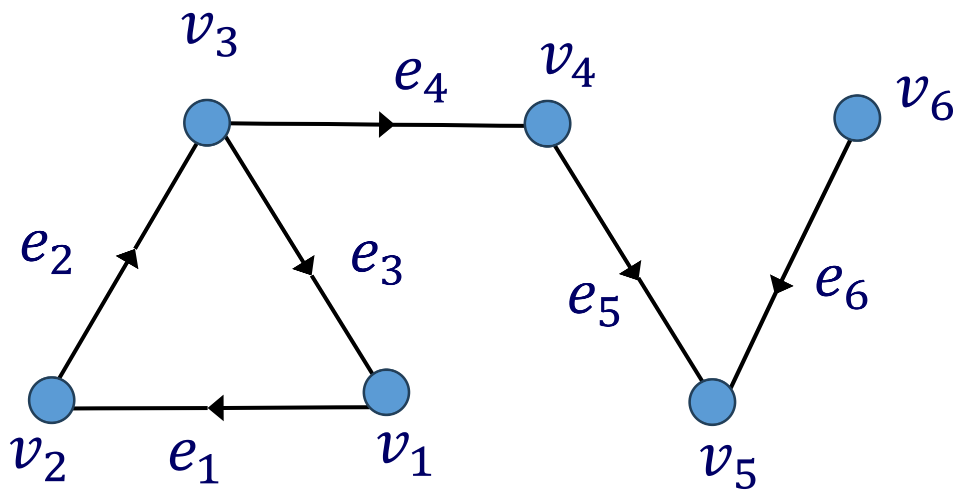


图 G

看作无向图时，直径为4

$R(v_1) = R(v_2) = R(v_3) = \{v_1, v_2, v_3, v_4, v_5\}$

$R(v_4) = \{v_4, v_5\}$

$R(v_5) = \{v_5\}$

$R(v_6) = \{v_5, v_6\}$

$d(v_1, v_2) = 1$

$d(v_2, v_1) = 2$

$d(v_5, v_6) = \infty$

（4）加权图

定义3.5 设图 $G=<V,E,\Psi>$。若 $W:E\to \mathbf{R}_+$（$\mathbf{R}_+$是正实数集），则称<G，W>为加权图。

(1) 若 $e\in E$，称 $W(e)$ 为边 e 的加权长度。

(2) 路径中所有边的加权长度之和称为该路径的加权长度。

(3)从结点v至结点 v'的路径中，加权长度最小的称为从 v 至 v'的 最短路径。

(4) 若从v可达v'，则称从v 至 v'的最短路径的加权长度为从v至v'的加权距离。

(5) 若从 v 不可达 v'，则称从 v 至 v' 的加权距离为 ∞。

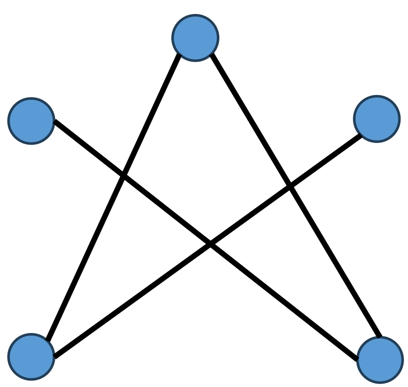
3.2 连通性

1、无向图的连通性

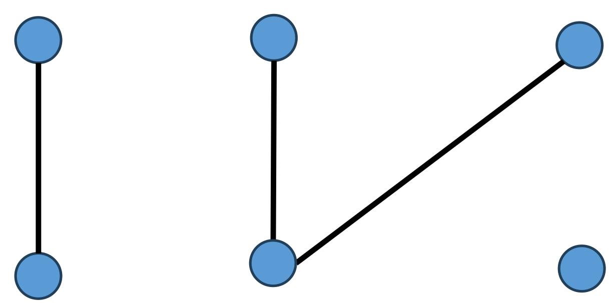
2、有向图的连通性

(1) 无向图的连通性

定义3.6 如果无向图 G 的任意两个结点都互相可达，则称G是连通的；否则称G是非连通的。



连通图

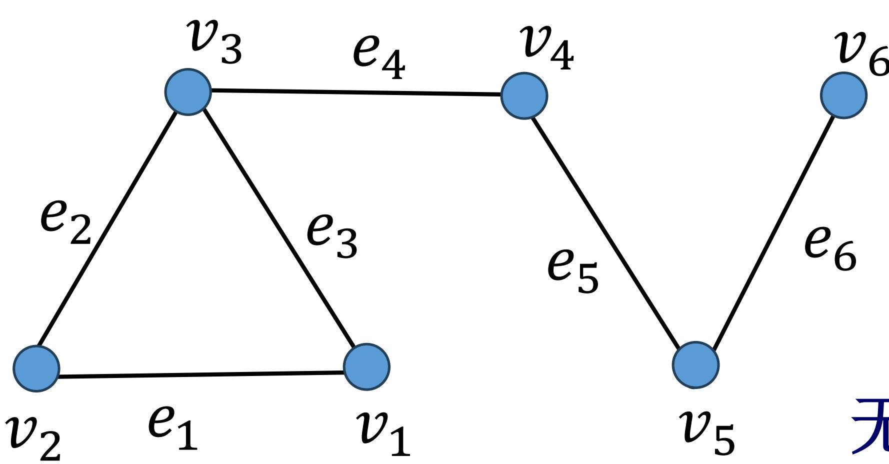


非连通图

(1) 无向图的连通性

定义3.6 如果无向图 G 的任意两个结点都互相可达，则称G是连通的；否则称G是非连通的。

- 无向图 $G = <V, E, \Psi>$是连通的 当且仅当对于任意$v \in V$，皆有 $R(v) = V$。
  - $R(v)$是从 v 可达的全体结点的集合

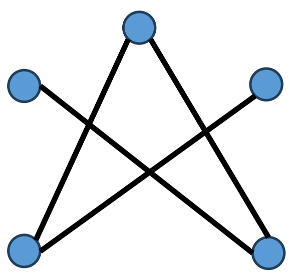


无向连通图

连通分支(connected component)

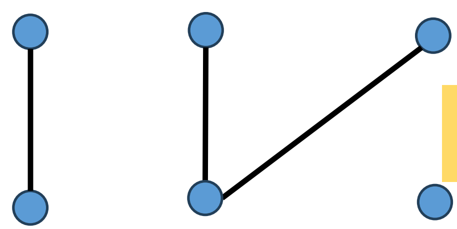
定义3.7 设G'是图G的具有某性质 P 的子图，并且对于G的具有该性质的任意子图 G''，只要$G' \subseteq G''$就有 $G' = G''$，则称 G'相对于该性质是 G 的极大子图。

定义3.8 无向图G的极大的连通子图称为G 的连通分支，简称分支。



1 个连通分支

连通图



3 个连通分支

非连通图

连通分支(connected component)

定义3.7 设G'是图G的具有某性质 P 的子图，并且对于G的具有该性质的任意子图 G''，只要$G' \subseteq G''$就有 $G' = G''$，则称 G'相对于该性质是 G 的极大子图。

定义3.8 无向图G的极大的连通子图称为G 的连通分支，简称分支。

定理3.4

(1) 连通无向图恰有一个分支。

(2) 非连通无向图的分支多于一个。

连通分支与等价关系

- 给定无向图 $G=<V，E，\Psi>$。
  - ✓ 定义关系：$R_V=\{<u,v>|\ u,v\in V$且从u到v可达$\}$

    R_V是V上的等价关系。
  - ✓ 分支：V关于R_V的等价类的导出子图。

    商集$V/R=\{V_1, V_2,\ldots, V_k\}$,

    $G[V_1], G[V_2], \ldots, G[V_k]$为$G$的分支。

例：设无向图 $G = \langle V, E, \Psi \rangle$，证明：$G$是连通图当且仅当对 V的每个划分 $\{V_1, V_2\}$，必存在一条边，它的两个端点分别属于V_1和 V_2。

证明：（充分性）

用反证法进行证明。

假设 G 不连通，则 G 至少有两个连通分支。

设 G 有m个连通分支 G_1, G_2，…, G_m，（$n>1$），

且分支G_1的结点集合为 V_1。

令$V_2=V-V_1$，则$\{V_1, V_2\}$构成一个划分。

显然，在V_1与V_2之间不可能存在两个端点分别属于V_1和V_2的边，否则与 G_1是连通分支矛盾。

故假设不成立，即 G 是连通图。

例：设无向图 $G=<V, E, \Psi>$，证明：G是连通图当且仅当对 V的每个划分 $\{V_1, V_2\}$，必存在一条边，它的两个端点分别属于V_1和 V_2。

证明： （必要性）

当 $|V|=1$ 时，V 没有划分，因此，假设 $|V|>1$。

对 V的每个划分 $\{V_1, V_2\}$，设$v_1 \in V_1$，$v_2 \in V_2$，由连通性知，从v_1到v_2存在一条简单路径 P 且长度设为 n。

下面对 n进行数学归纳证明：P 上必存在一条端点分别在 V_1和 V_2的边。

(1) 当$n=1$时， P 为一条边，显然成立。

(2) 设$n=k$时结论成立。当$n=k+1$时，

考虑路径 P上与v_1 邻接的点 u。

若$u \in V_2$，则边$\{v_1, u\}$ 的端点合乎要求。

若$u \in V_1$，则从 P 中删去v_1 可得u 至v_2 的简单路径，且长度为 k。

由归纳假设知，结论成立。

（2）有向图的连通 — 基础图

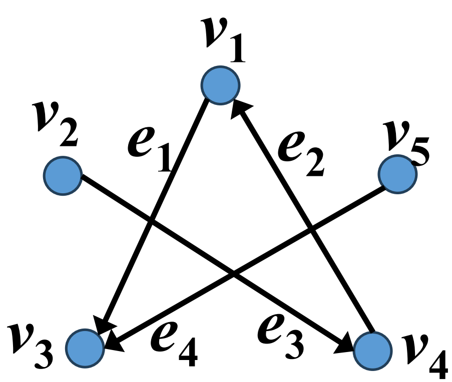
定义3.9 设有向图 $G = <V, E, \Psi>$， 如下定义

$$\Psi' : E \rightarrow \{ \{v_1, v_2\} \mid v_1 \in V \wedge v_2 \in V \},$$

使得，对任意 $e \in E$ 和 $v_1, v_2 \in V$,

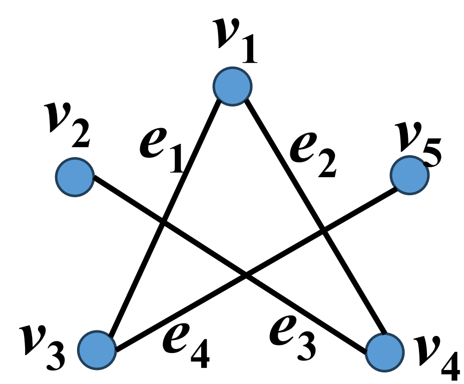
若 $\Psi(e) = <v_1, v_2>$，则 $\Psi'(e) = \{v_1, v_2\}$。

称无向图 $G' = <V, E, \Psi'>$为有向图 G 的**基础图**。



$G = <V, E, \Psi>$

有向边改为无向边



$G' = <V, E, \Psi'>$

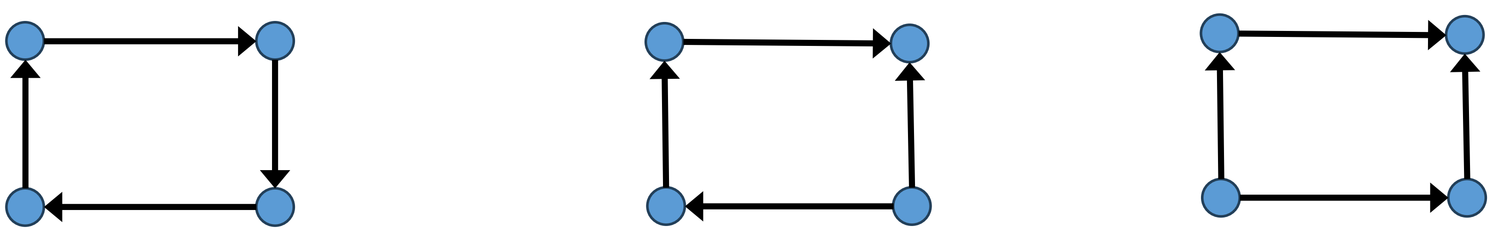
（2）有向图的连通 — 3种连通性

定义3.10 设 G 是有向图。

(1) 如果G中任意两个结点都互相可达，则称G是强连通的；

(2) 如果对于G的任意两结点，必有一个结点可达另一结点，则称 G 是单向连通的；

(3) 如果 G 的基础图是连通的，则称 G 是弱连通的。



强连通图 → 单向连通图 → 弱连通图

（2）有向图的连通 — 3种连通性

定义3.10　设 G 是有向图。

(1) 如果G中任意两个结点都互相可达，则称G是强连通的；

(2) 如果对于G的任意两结点，必有一个结点可达另一结点，则称 G 是单向连通的；

(3) 如果 G 的基础图是连通的，则称 G 是弱连通的。

- 是否有满足以上连通性的极大子图？

定义3.7　设G'是图G的具有某性质 P 的子图，并且对于G的具有该性质的任意子图 G''，只要$G'\subseteq G''$就有 $G'= G''$，则称 G'相对于该性质是 G 的极大子图。

有向图的分支

定义3.11 设 G 是有向图。

(1) G 的极大强连通子图称为 G 的强（连通）分支。

(2) G 的极大单向连通子图称为 G 的单向分支。

(3) G 的极大弱连通子图称为 G 的弱分支。

定理 3.5 (1) 强连通（单向连通，弱连通）有向图恰有一个强分支（单向分支，弱分支）。

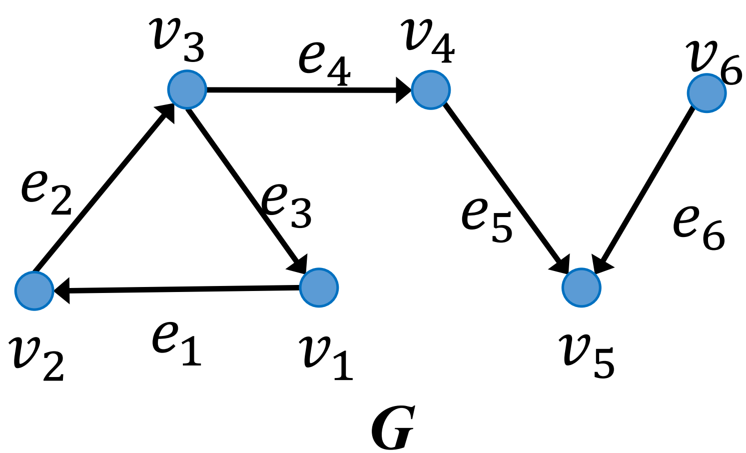
(2)非强连通（非单向连通，非弱连通）有向图有一个以上强分支（单向分支，弱分支）。

定义3.11 设 G 是有向图。

(1) G 的极大强连通子图称为 G 的强（连通）分支。

(2) G 的极大单向连通子图称为 G 的单向分支。

(3) G 的极大弱连通子图称为 G 的弱分支。



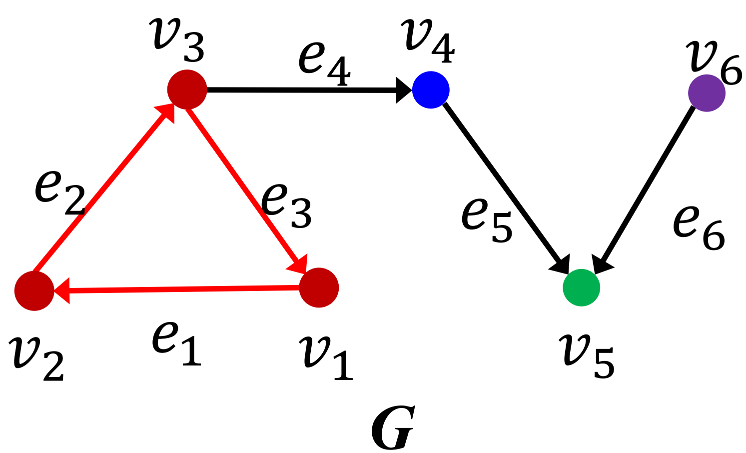
- 个强分支:
- 个单向分支:
- 个弱分支

定义3.11 设 G 是有向图。

(1) G 的极大强连通子图称为 G 的**强（连通）分支**。

(2) G 的极大单向连通子图称为 G 的**单向分支**。

(3) G 的极大弱连通子图称为 G 的**弱分支**。



- 4 个强分支：

$G[\{v_1,v_2,v_3\}]$, $G[\{v_4\}]$, $G[\{v_5\}]$, $G[\{v_6\}]$

- 个单向分支：

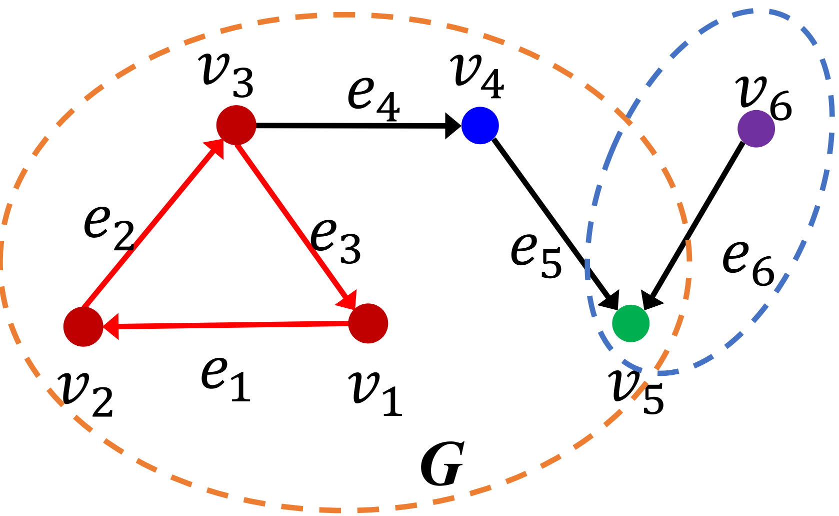
- 个弱分支

定义3.11 设 G 是有向图。

(1) G 的极大强连通子图称为 G 的强（连通）分支。

(2) G 的极大单向连通子图称为 G 的单向分支。

(3) G 的极大弱连通子图称为 G 的弱分支。



- **4** 个强分支：

$G[\{v_1,v_2,v_3\}]$, $G[\{v_4\}]$, $G[\{v_5\}]$, $G[\{v_6\}]$

- **2** 个单向分支：

$G[\{v_1, v_2, v_3, v_4, v_5\}]$，$G[\{v_5, v_6\}]$

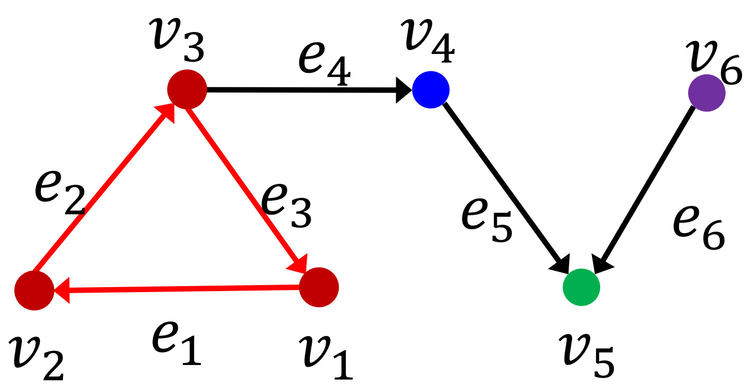
- 个弱分支

定义3.11 设 G 是有向图。

(1) G 的极大强连通子图称为 G 的**强（连通）分支**。

(2) G 的极大单向连通子图称为 G 的**单向分支**。

(3) G 的极大弱连通子图称为 G 的**弱分支**。



G

- 4 个强分支：

$G[\{v_1,v_2,v_3\}]$, $G[\{v_4\}]$, $G[\{v_5\}]$, $G[\{v_6\}]$

- 2 个单向分支：

$G[\{v_1, v_2, v_3, v_4, v_5\}]$，$G[\{v_5, v_6\}]$

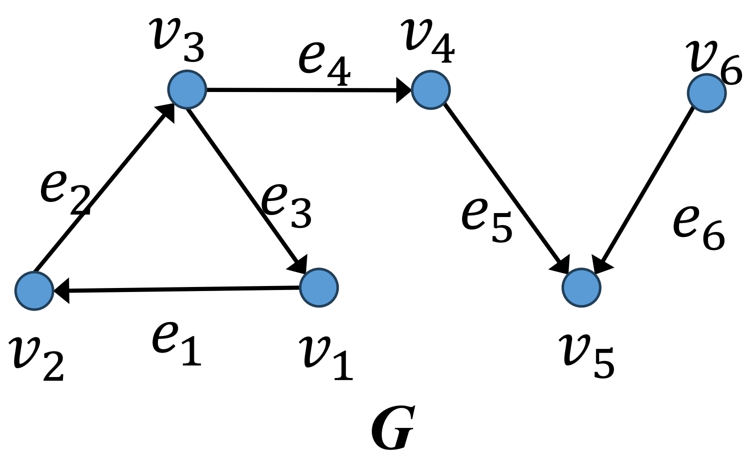
- 1 个弱分支：G

定义3.11 设 G 是有向图。

(1) G 的极大强连通子图称为 G 的**强（连通）分支**。

(2) G 的极大单向连通子图称为 G 的**单向分支**。

(3) G 的极大弱连通子图称为 G 的**弱分支**。



- 4 个强分支：

$G[\{v_1,v_2,v_3\}]$, $G[\{v_4\}]$, $G[\{v_5\}]$, $G[\{v_6\}]$

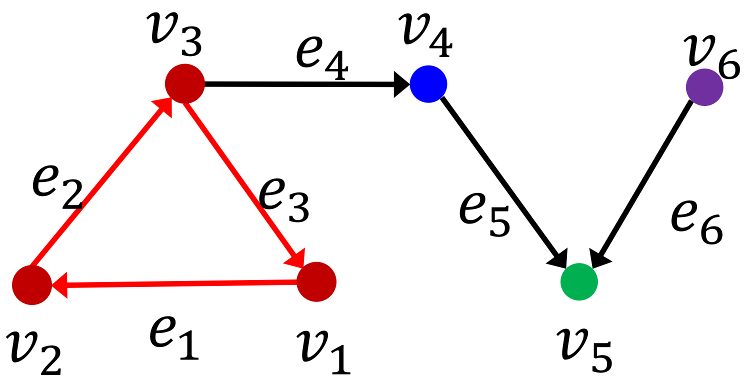
- 2 个单向分支：

$G[\{v_1, v_2, v_3, v_4, v_5\}]$， $G[\{v_5, v_6\}]$

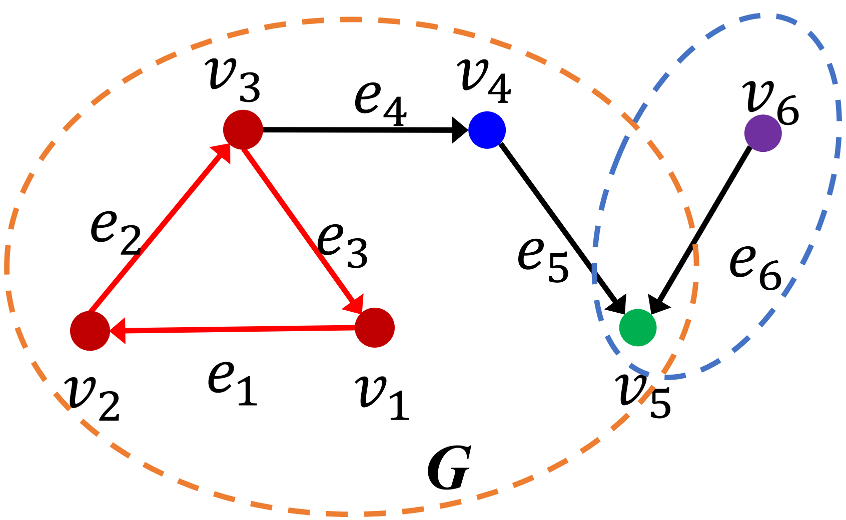
- 1 个弱分支：G

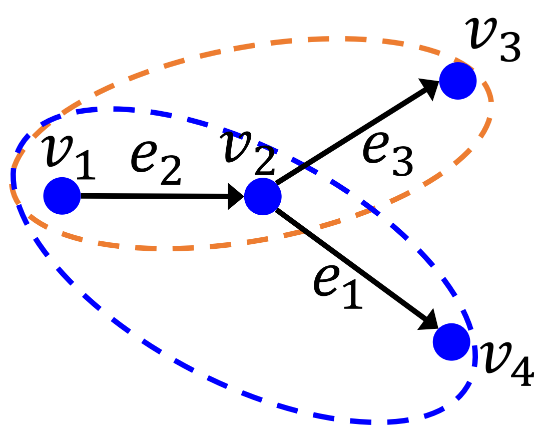
问题：给定有向图 $G =<V，E，\Psi>$，能否定义V上的等价关系使得等价类的导出子图构成强分支 (单向分支，弱分支)?

例：有向图的每个结点（每条边）是否处于一个强分支中？是否恰处于一个单向分支中？



强分支：
结点是，边不一定





单向分支：
结点与边都不一定

边出现在两个单向分支

3.3 回路、半回路、有向回路

定义3.12

1) 设 G'是有向图 $G=<V, E, \Psi>$的基础图，G'中的路径称为 G 中的半路径。

2) 设 $v_0e_1v_1\ldots v_{m-1}e_mv_m$是$G$中的半路径。对每个$i$ $(1\le i\le m)$，

- 若$\Psi(e_i)=<v_{i-1}, v_i>$，则称 e_i是该半路径中的正向边；
- 如果$\Psi(e_i)=<v_i, v_{i-1}>$，则称 e_i是该路径中的反向边。

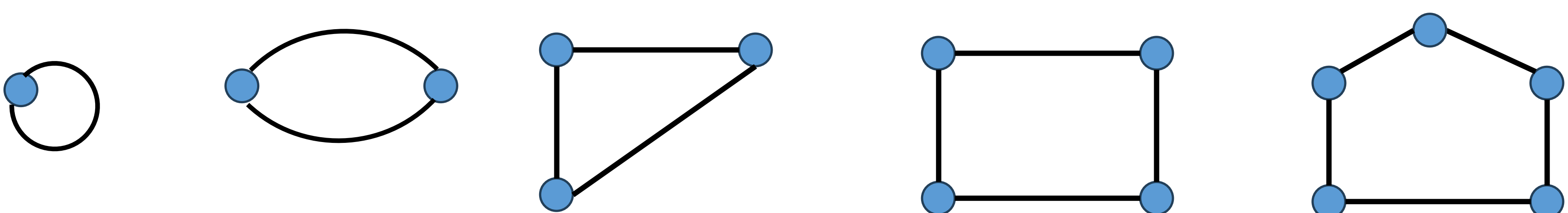
- 有向图 G 中的路径一定是 G 中的半路；但 G 中的半路径未必是 G 中的路径。
- 有向图中的半路径是路径当且仅当该半路径中的边都是正向边

3.3 回路、半回路、有向回路

定义3.13

所有结点的度均为自然数 d 的无向图称为 d 度正则图

1) 连通2度正则图称为回路；
2) 基础图是回路的有向图称为半回路；
3) 每个结点的出度和入度均为 1 的弱连通有向图称为有向回路；
4) 回路 (半回路，有向回路)边的数目称为回路 (半回路，有向回路)的长度。

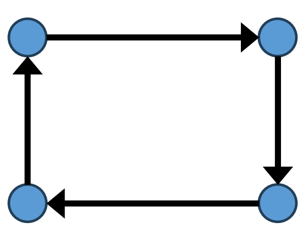


回路

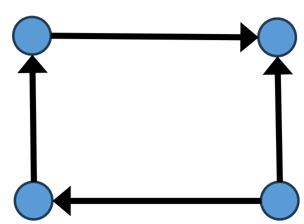
3.3 回路、半回路、有向回路

定义3.13

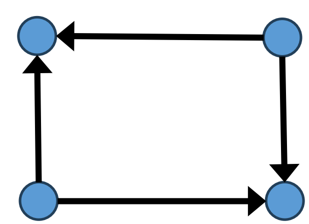
1) 连通2度正则图称为回路；
2) 基础图是回路的有向图称为半回路；
3) 每个结点的出度和入度均为 1 的弱连通有向图称为有向回路；
4) 回路 (半回路，有向回路)边的数目称为回路 (半回路，有向回路)的长度。



有向回路



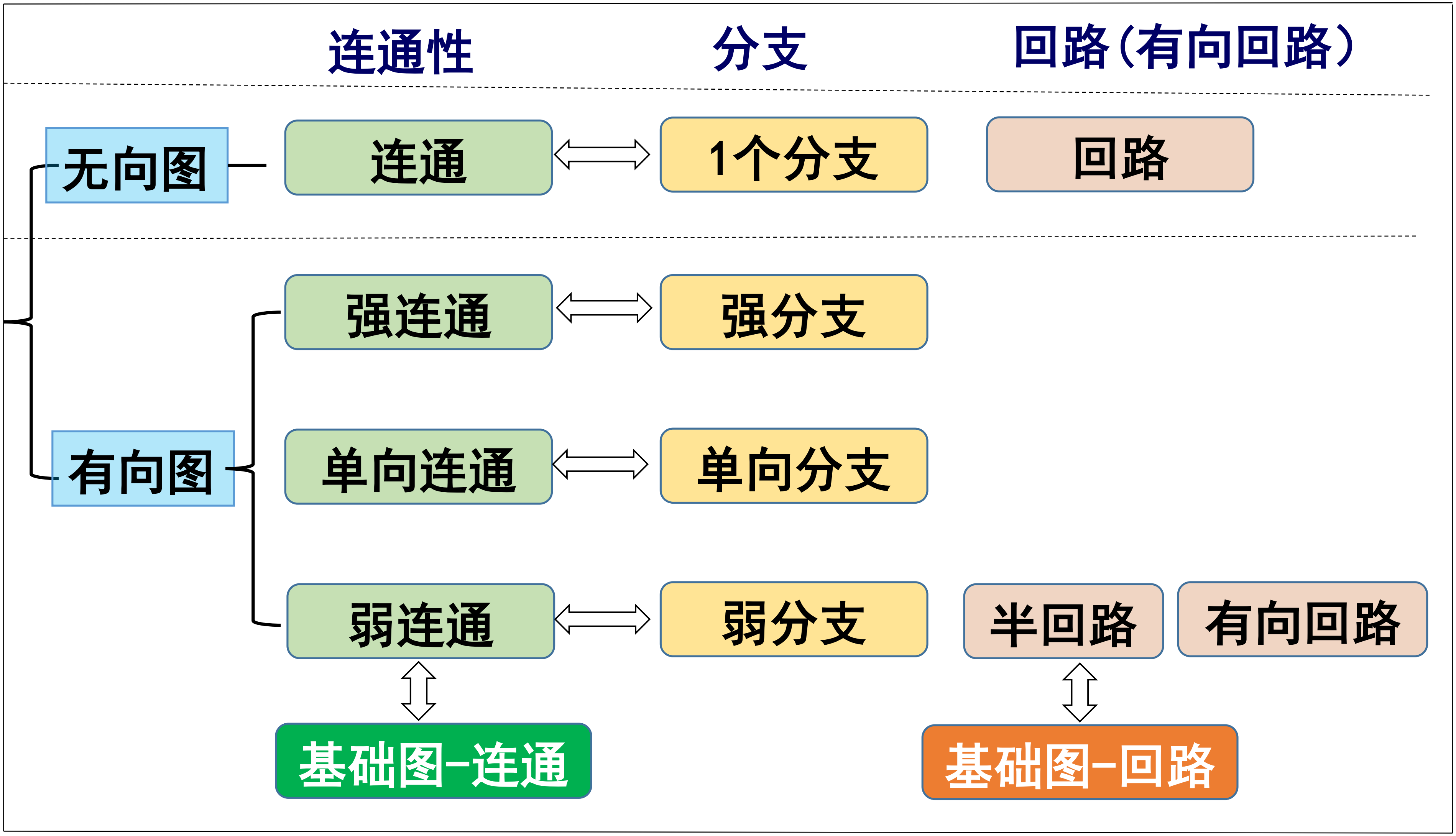
半回路



半回路

长度均为 4

部分概念关系图



连通性
分支
回路(有向回路)
无向图
连通
1个分支
回路
有向图
强连通
强分支
单向连通
单向分支
弱连通
弱分支
半回路
有向回路
基础图-连通
基础图-回路

定理3.6 设 v 是图G的任意结点，G是回路（或有向回路），当且仅当

(1) G的阶与边数相等，且

(2) 在G中存在一条 v到v的闭路径，使得除了v在该闭路径中出现两次外，其余结点和每条边都在该闭路中恰出现一次。

证明： 充分性显然，下面证明必要性。

设 $G= <V, E, \Psi>$，$V=\{v_1,\ldots, v_n\}$，$E=\{e_1,\ldots, e_n\}$ 。

(1) 首先证明 G 的阶与边数相等。

若 G 是回路（或有向回路），则G（或G的基础图）是连通2度正则图，得 $\sum_{v\in V} d_G(v) = 2|V|$。

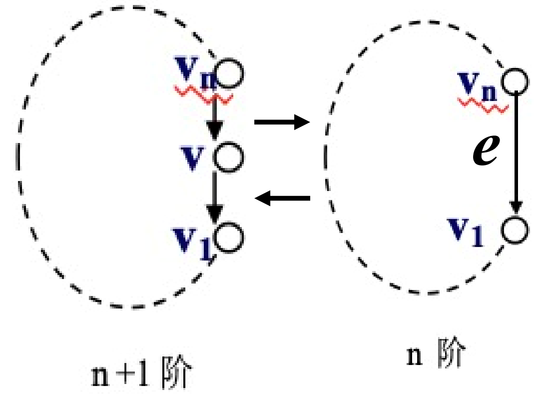
又由握手定理知，$\sum_{v\in V} d_G(v) = 2|E|$。

因此，G的阶与边数相等。

证明（续）(2) 对G的阶n用第一归纳法。

(a) 若G是1阶有向回路，则G只有一个自圈e_1，

且 $v_1e_1v_1$是一条闭路径。



(b) 设G是n阶有向回路时结论成立。

若 G为$n+1$阶有向回路，

由于$d_G^+(v)= d_G^-(v)=1$，一定存在$v_1, v_n \in V$ 和$e_1, e_{n+1} \in E$，

使得$\Psi(e_1) = <v, v_1>$且$\Psi(e_2) =<v_n, v>$。

设边$e \notin E$, $\Psi'(e)=\{<v_n,v_1>\}$，

定义图$G' = (G-\{v\})+\{e\}_{\Psi'}$，则$G'$上$n$ 阶有向回路。

由归纳假设知，G'存在闭路径 $v_1e_2v_2....v_{n-1}e_nv_nev_1$。

此时， $v\ e_1v_1e_2v_2....v_{n-1}e_nv_ne_{n+1}v$ 是G中满足条件的闭路径。

有回路、非循环图

定义3. 14 (1) 如果回路（有向回路，半回路）C 是图 G 的子图，则称 G 有回路 (有向回路，半回路) C。

(2)没有回路的无向图和没有半回路的有向图称为非循环图。

回路：连通2度正则图；

半回路：基础图是回路的有向图

- 树和有向树都是非循环图
- 问题：如何判断一个图是非循环图？

有向回路的充分条件

定理3.7 如果有向图 G 有子图 G'，使得 对于 G'的任意结点 v，皆有 $d_{G'}^{+}(v)>0$，则 G 有有向回路。

定理3.8 如果有向图 G 有子图 G'，使得 对于 G'中的任意结点 v，$d_{G'}^{-}(v)>0$，则 G 有有向回路。

证明：只证明定理 3.8，定理3.7同样可证。

有向回路的充分条件

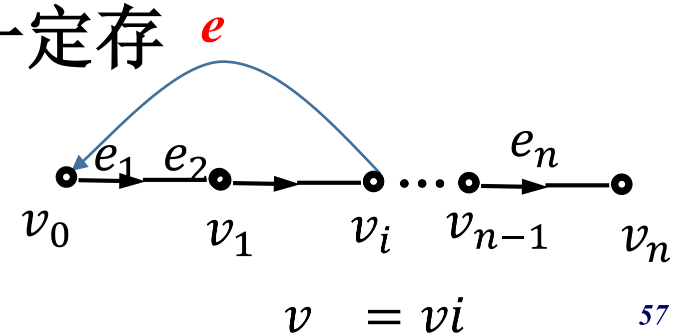
定理3.8 如果有向图 G 有子图 G'，使得对于 G'中的任意结点 v， $d^-_{G'}(v) > 0$，则 G 有有向回路。

证明：设 $G'=<V', E', \Psi'>$，$v_0e_1v_1\ldots v_{n-1}e_nv_n$ 是 G' 中最长的基本路径。

由于$d^-_{G'}(v_0)>0$，必可找到边 $e \in E'$ 和 $v \in V'$，使得

$$ vev_0e_1v_1\ldots v_{n-1}e_nv_n $$ 是G'中简单路径。

由于$v_0e_1v_1\ldots v_{n-1}e_nv_n$是最长的基本路径，则一定存在$i \in [0, n]$，使得 $v = v_i$。



因此，G的以$\{v_0, \ldots, v_i\}$为结点集合，

以 $\{e, e_1, \ldots, e_i\}$为边集合的子图是有向回路。

W-过程：判断一个有向图是否有有向回路

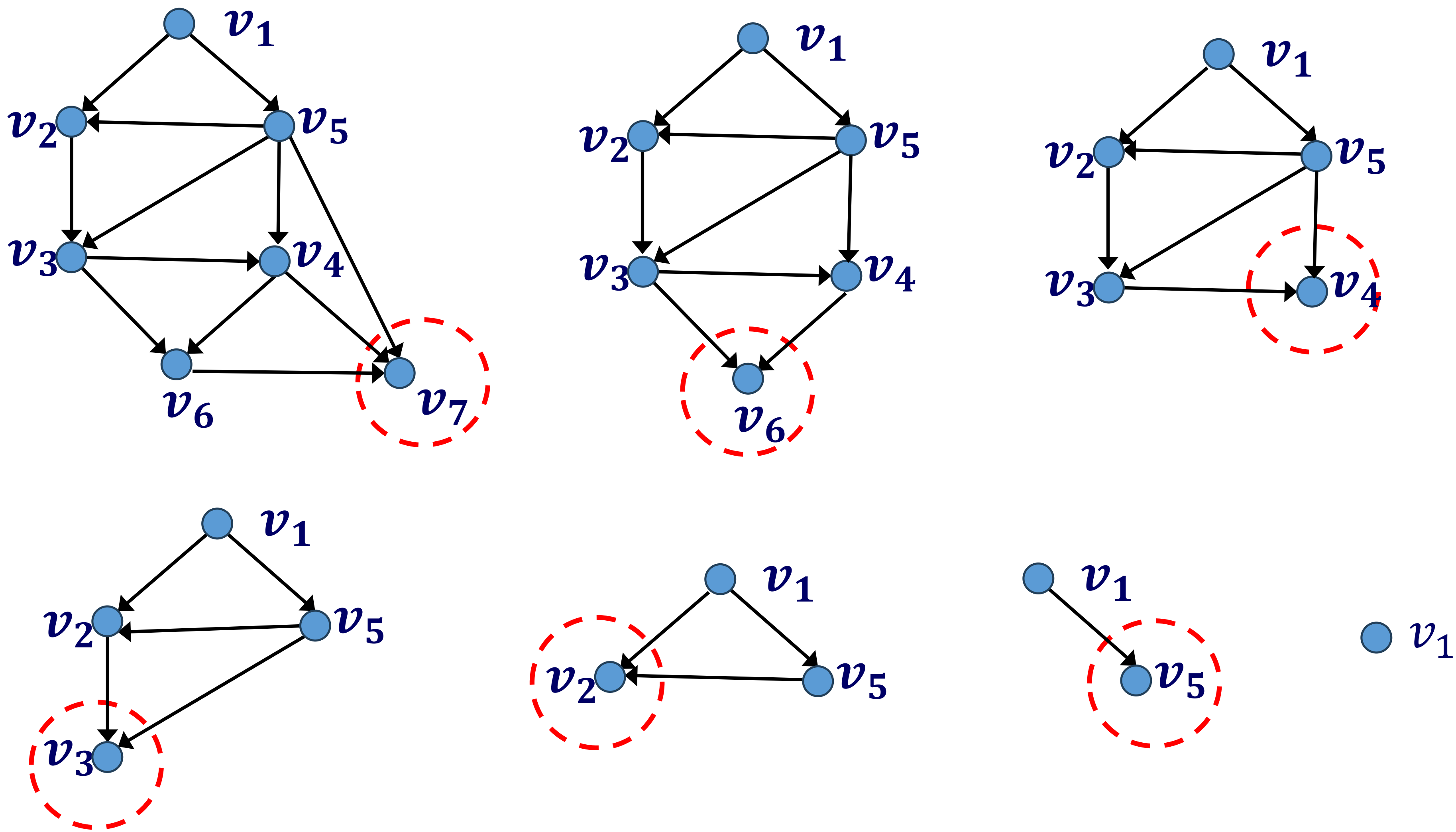
定理3.7 如果有向图 G 有子图 G'，使得 对于 G'的任意结点 v，皆有 $d^{+}_{G'}(v) > 0$，则 G 有有向回路。

定理3.8 如果有向图 G 有子图 G'，使得 对于 G'中的任意结点 v，$d^{-}_{G'}(v) > 0$，则 G 有有向回路。

设 v 是有向图G的结点且$d^{+}_{G'}(v)=0$ 或 $d^{-}_{G'}(v)=0$。

- 从G中去掉 v 和与之关联的边得到有向图 $G-\{v\}$的过程称为*W*-过程。
  - ✓ G 有有向回路当且仅当 $G-\{v\}$ 有有向回路；
  - ✓ 若 n 阶有向图 G 没有有向回路，则经过 $n-1$ 次*W*-过程得到平凡图。

例：判断一个有向图是否有有向回路的W-过程

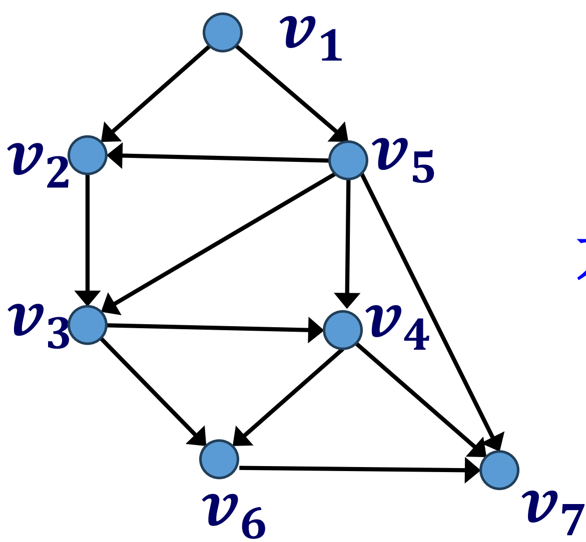


没有有向回路

非循环图的充分必要条件

定理3.9 图 G 不是非循环图当且仅当 G 有子图 G'，使得对于 G' 的任意结点 v，皆有 $d_{G'}(v) > 1$。

- 等价于：图 G是非循环图当且仅当 对于G的任意子图 G'，一定存在一个结点 v，使得 $d_{G'}(v) \leq 1$。



不是非循环图

判断一个图是否是非循环图

定理3.9 图 G 不是非循环图当且仅当 G 有子图 G'，使得对于G' 的任意结点 v，皆有 $d_{G'}(v) > 1$。

- 给定 n阶图有向图 G，令$G_0 = G$。
- 对于小于n-1的自然数 i，如下归纳定义 G_i:
  - ✓ 若 G_i 有结点v_i满足$d_{G_i}(v) \le 1$，则令$G_{i+1} = G_i - \{v_i\}$。

得到图序列 G_0，G_1，…，G_m，其中$0 \le m \le n-1$。

- 若G_m是平凡图，则 G 是非循环图，否则不然。

连通的充分条件

例：设 G 为 n 阶简单无向图，

(1) 若 G 的任意两个结点的度数之和大于等于 $n-1$，则 G 是连通的。

(2) 若对 G 的任意结点 v, 皆有 $d_G(v) \geq (n-1)/2$，则 G 是连通的。

证明：(1) 设 u, v 是图 G 中任意两个结点，则

$$d_G(u) + d_G(v) \geq n-1\text{。}$$

若 u 与 v 邻接，则 u 从 v 可达。

若 u 与 v 不邻接，由于 G 为简单无向图, 因此 G 没有自圈，所以 u, v 只能与剩下的 n-2个结点相连。

由抽屉原理得，至少存在一个结点既与 u 相连，又与 v 相连，得，u 从 v 可达。

因此，G 是连通无向图。

连通的充分条件

例：设 G 为 n 阶简单无向图，

(1) 若 G 的任意两个结点的度数之和大于等于 $n-1$，则 G 是连通的。

(2) 若对G的任意结点 v, 皆有$d_G(v) \geq (n-1)/2$，则G是连通的。

证明：(2) 对任意的两个结点u, v， 有

$$d_G(u)+d_G(v) \geq (n-1)/2+(n-1)/2 = n-1。$$

由(1)得，G是连通的。

例：设 G为 n 阶简单无向图，且G有k个分支，m条边，则有 $m \le (n-k)(n-k+1)/2$。

扩展：如何定义连通度

- 问题：如何定量地比较无向图的连通性的强与弱？
  - ✓ 点连通度：为了破坏连通性，至少需要删除多少个顶点？
  - ✓边连通度：为了破坏连通性，至少需要删除多少条边？
  - ✓破坏连通性：是指“变得更加不连通”

扩展：点割集的定义

- 设无向图$G=<V, E, \Psi>$为连通图，若有点集$V_1 \subset V$, 使
  - ✓ 图G删除了V_1的所有结点后，所得的子图是不连通图，
  - ✓ 而删除了V_1的任意真子集后，所得到的子图仍是连通图，

则称V_1是G的一个点割集。

- 若某一个结点构成一个点割集，则称该结点为割点。

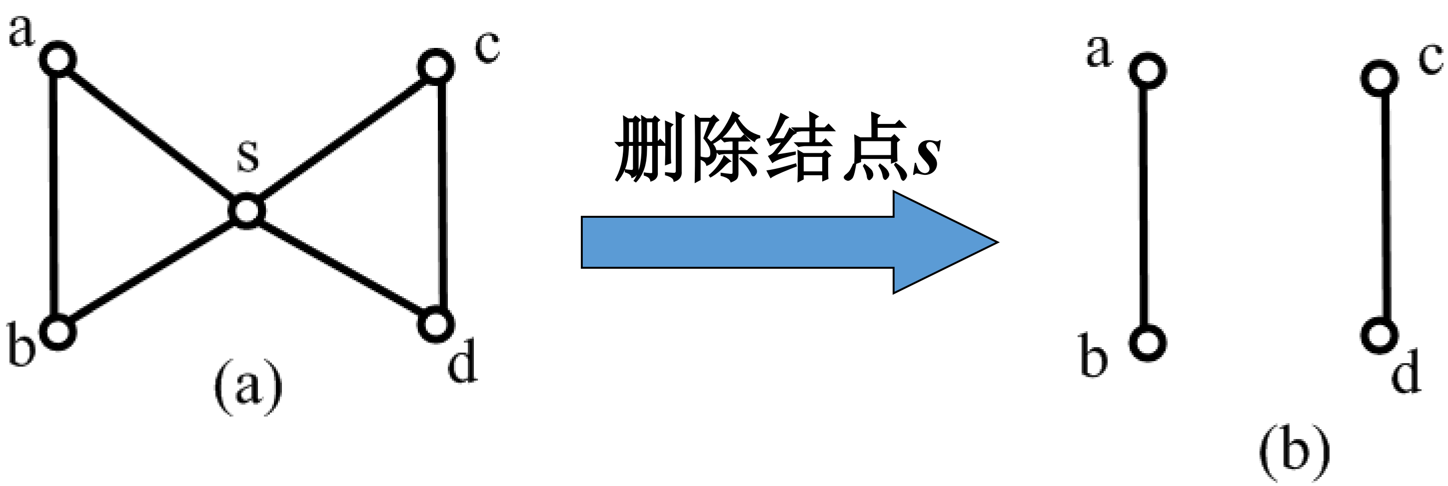
形式化为：

若分支数$W(G-V_1) > W(G)$且$\forall V' \subset V_1$, $W(G-V') = W(G)$, 则称 V_1为G的点割集。

极小性

若$\{v\}$为点割集, 则称v为割点.

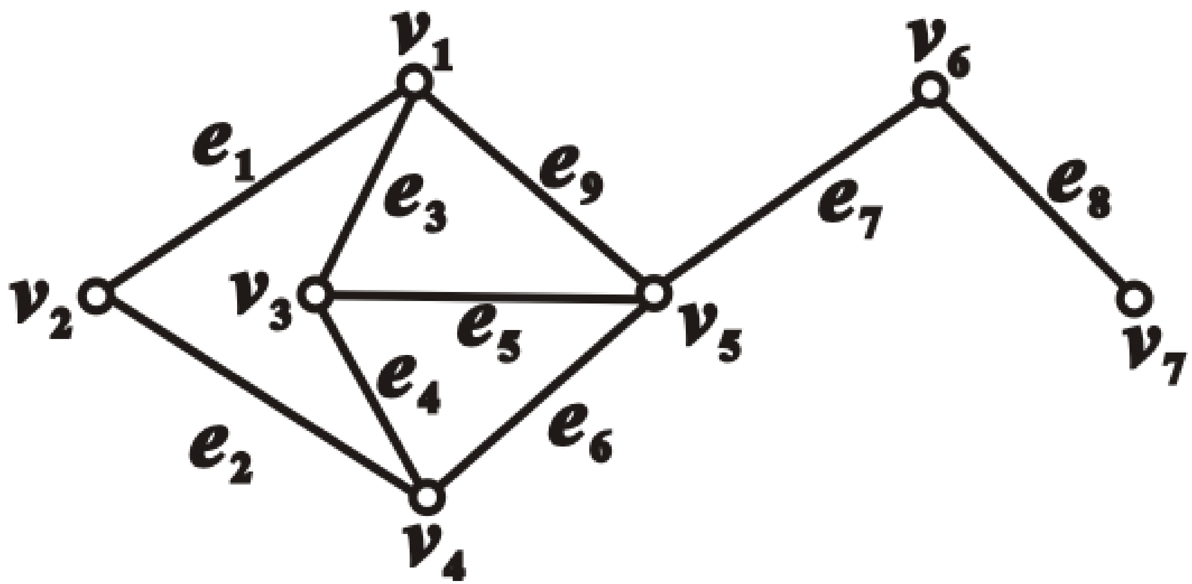
例： 求下图的割点.



连通图， W=1 非连通图， W=2

因此*s*是割点。

例：在下图所示的图中，找出点割集和割点。



点割集：$\{v_1,v_4\}$, $\{v_6\}$，$\{v_5\}$，

割点：v_6， v_5

v_2、v_3与 v_7不在任何点割集中。

扩展：无向图的点连通度

- 设G是无向图，$k(G)$ = min{ $|V_1|$ | V_1是 G 的点割集 }是G的**点连通度**，也称作连通度。
- 几点说明：
  - ✓ 连通度$k(G)$表示为了产生一个不连通图所需要删除的点的最少数目。
  - ✓ 非连通图的连通度等于0，存在割点的连通图的连通度为1，n阶完全图的连通度为n-1。
  - ✓ 连通度$k(G)$表示图G的连通程度，$k(G)$大表示连通性强，即需要删除更多的点才能使图从连通变为非连通。

扩展：边割集

◆ 设无向图 $G = <V, E>$ 为连通图, 若有边集 $E_1 \subset E$, 使

- ✓ 图 G 删除了 E_1的所有边后，所得的子图是不连通图，
- ✓ 而删除了E_1的任意真子集后，所得到的子图仍是连通图，

则称E_1是G的一个边割集。

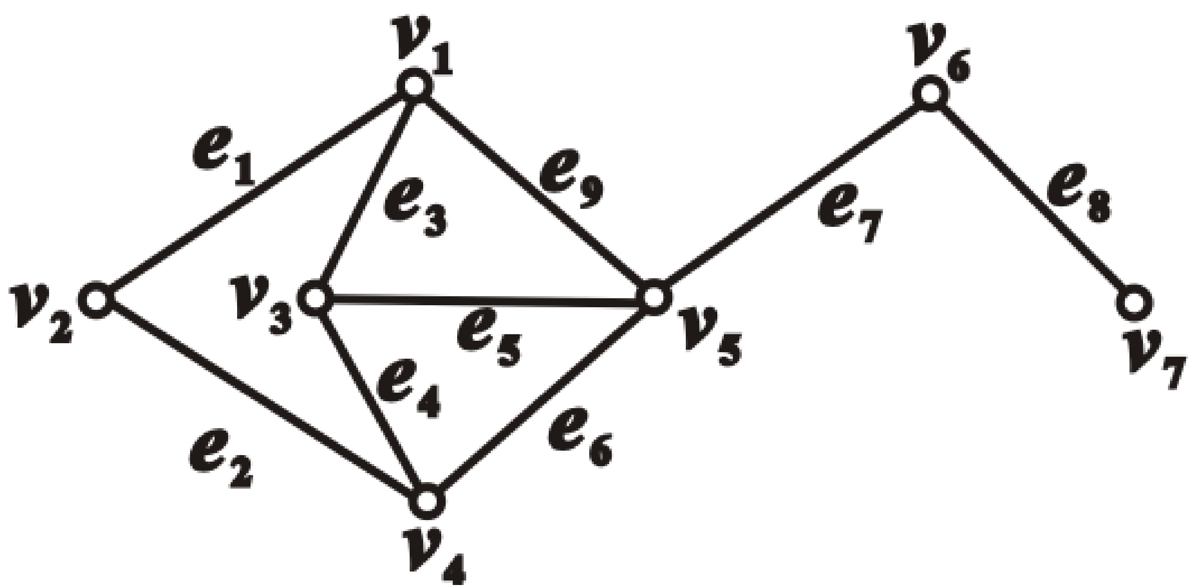
极小性

◆若某一条边构成一个边割集，则称该边为割边（或桥）。

更一般定义为：

若$W(G-E_1)>W(G)$且$\forall E'\subset E_1$，$W(G-E')=W(G)$，则称E_1为G的边割集. 若$\{e\}$为点割集, 则称e为割边.

例3 在下图所示的图中，举出边割集和桥的例子。



边割集：$\{e_1, e_2\}$, $\{e_3, e_4, e_5\}$, $\{e_1, e_3, e_9\}$, $\{e_7\}$，$\{e_8\}$等

割边：e_7，e_8

扩展：边连通度

- 设G是无向图，称

$$\lambda(G)=\min\{\,|E_1|\mid E_1\text{是}G\text{的边割集}\,\}$$

是G的边连通度。

- 几点说明：
  - ✓ 边连通度 $\lambda(G)$ 是为了产生一个不连通图所需要删除的边的最少数目。
  - ✓ 非连通图的边连通度等于0，存在桥的连通图的边连通度为 1，平凡图的边连通度为 0。
  - ✓ 边连通度 $\lambda(G)$ 表示图G的边连通程度， $\lambda(G)$大表示边连通性强，即需要删除更多的边才能使图从连通变为非连通。

割点、割边的充分必要条件

- 一个连通无向图G中的结点v是割点 $\Leftrightarrow$ 存在结点u和w，使得连接u和w的每条路都经过v。

- 一个连通无向图G中的边e是割边 $\Leftrightarrow$ 存在结点u和w，使得连接u和w的每条路都经过e。

迪克斯特拉（Dijkstra）

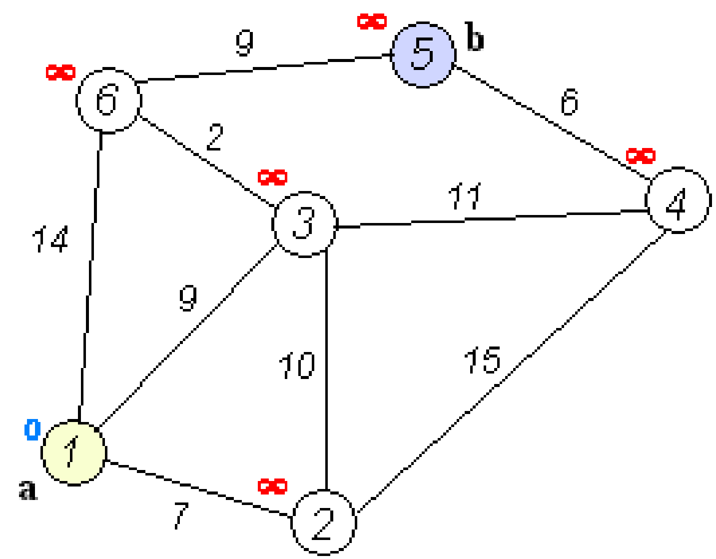
- 艾兹格·W·迪克斯特拉（Edsger Wybe Dijkstra，1930年5月11日~2002年8月6日）
- 荷兰人。 计算机科学家，毕业就职于荷兰Leiden大学，早年钻研物理及数学，而后转为计算学。
- 1972年获得图灵奖

迪克斯特拉（Dijkstra）

- 提出“goto有害论”；
- 提出信号量和pv原语;
- 解决了“哲学家聚餐”问题;
- Dijkstra最短路径算法和银行家算法的创造者;
- 第一个Algol 60编译器的设计者和实现者;
- THE操作系统的设计者和开发者;
- 与D. E. Knuth并称为我们这个时代最伟大的计算机科学家的人。
- 与癌症抗争多年，于2002年8月6日在荷兰Nuenen自己的家中去世，享年72岁

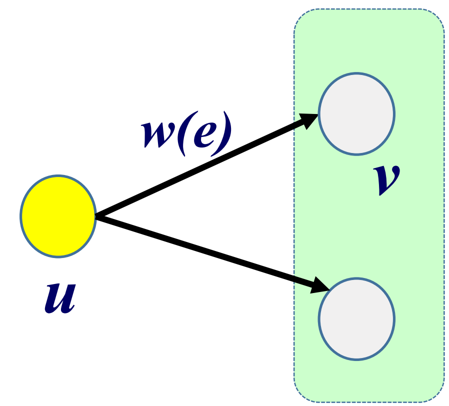
迪克斯特拉 (Dijkstra)算法

- 1959年, 最短路径算法
- 应用产物
  - ✓ 单源路径计算（Single-source shortest paths problem）
  - ✓（连通）有权（有向）图
  - ✓边的权值非负数
- 贪心算法(Greedy Algorithm)



(2)戴克斯特拉（Dijkstra）算法

- **初始化**
  - **设置两个集合**
    - S---已设置权值集
    - T=V-S---未设置权值节点集
  - **初始化**
    - $\lambda(s) \leftarrow 0$，其余为$\forall\ v \in V - \{s\}$，$\lambda(v) \leftarrow \infty$
- **算法流程**
  - **循环：n次**
    - 选择S中最小权值点u
    - 对所有T中e=(u, v)，
      更新$\lambda(v) = \min\{\lambda(v), \lambda(u)+w(e)\}$



(2) 迪克斯特拉 (Dijkstra)算法

算法（求从结点 s 至 t 的加权距离）

1) $\lambda(s) \leftarrow 0$，且 $\forall v \in V - \{s\}$，$\lambda(v) \leftarrow \infty$;

距离数组λ(v)

2) $S \leftarrow V$;

3) 任取 $u \in \{ u' \mid$ 若 $v' \in S$，则 $\lambda(u') \leq \lambda(v') \}$;

4) 如果 $u = t$，则 算法结束。

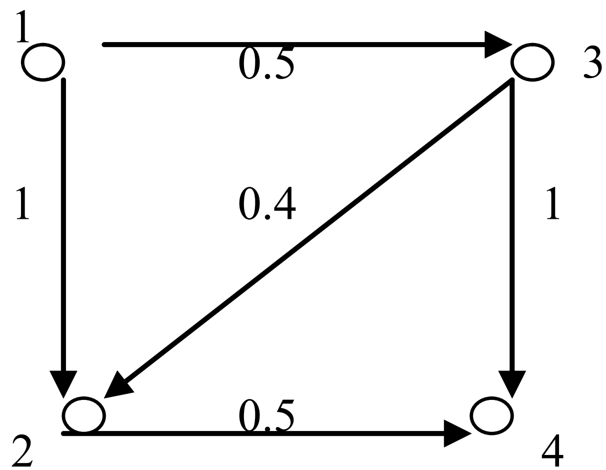
5) 对于以 u 为起点的每条边 e，如果 e 的终点 $v \in S$ 并且 $\lambda(v) > \lambda(u) + W(e)$，则 $\lambda(v) \leftarrow \lambda(u) + W(e)$;

6) $S \leftarrow S - \{u\}$， 且 转向 3) 。

当算法结束时，$\lambda(t)$ 即为从 s 至 t 的加权距离。

Single-source shortest paths problem

例子（加权距离）



当前点\λ\结点	1	2	3	4
	0	∞	∞	∞
第1步	/	1	0.5	∞
第2步	/	0.9	/	1.5
第3步	/	/	/	1.4
4				

从 1 到 4 的加权距离为 1.4。

思考题

- **如何修改算法，以便输出最短路径?**
  - 更新λ时，同时记录min{λ(v)}，是通过那个λ(u)+w(u, v)得到的
- **如何记录最短路条数?**
  - 如果λ(v)= λ(u)+w(e)，则num(v)+=num(u)；
  - 如果λ(v)> λ(u)+w(e)，则λ(v)=λ(u)+w(e)，且num(v)=num(u)

(2)戴克斯特拉（Dijkstra）算法

Input: A graph G, a matrix w representing the weights between vertices in G, source vertex s

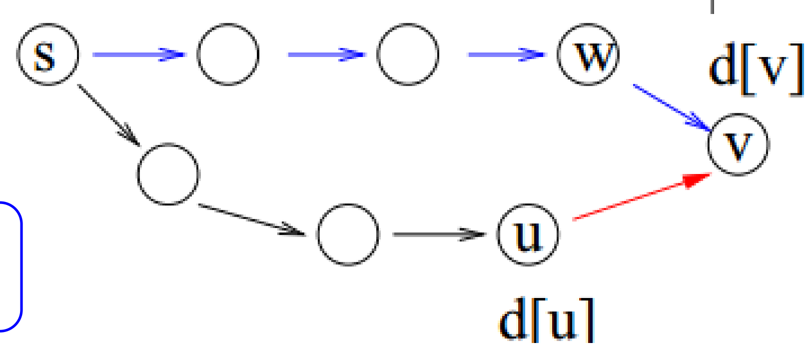
Output: None

for $u \in V$ **do**
  $d[u] \leftarrow \infty, color[u] \leftarrow$ WHITE;// Initialize
end
$d[s] \leftarrow 0$;
$pred[s] \leftarrow$ NULL;
$Q \leftarrow$queue with all vertices;
while *Non-Empty(Q)* **do**
  // Process all vertices
  $u \leftarrow$Extract-Min(Q);// Find new vertex
  for $v \in Adj[u]$ **do**
    if $d[u] + w(u,v) < d[v]$ **then**
      // If estimate improves
      $d[v] \leftarrow d[u] + w(u,v)$;// relax
      Decrease-Key($Q, v, d[v]$);
      $pred[v] \leftarrow u$;
    end
  end
  $color[u] \leftarrow$BLACK;
end

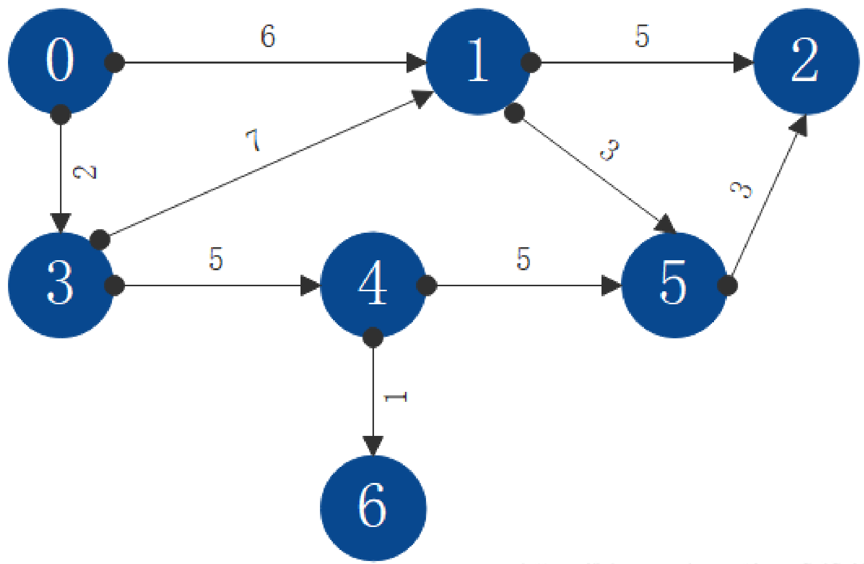
- d[u]：结点u到源点s最短距离
- Q：优先队列

v.key=d[v]

怎么构造最短路径？

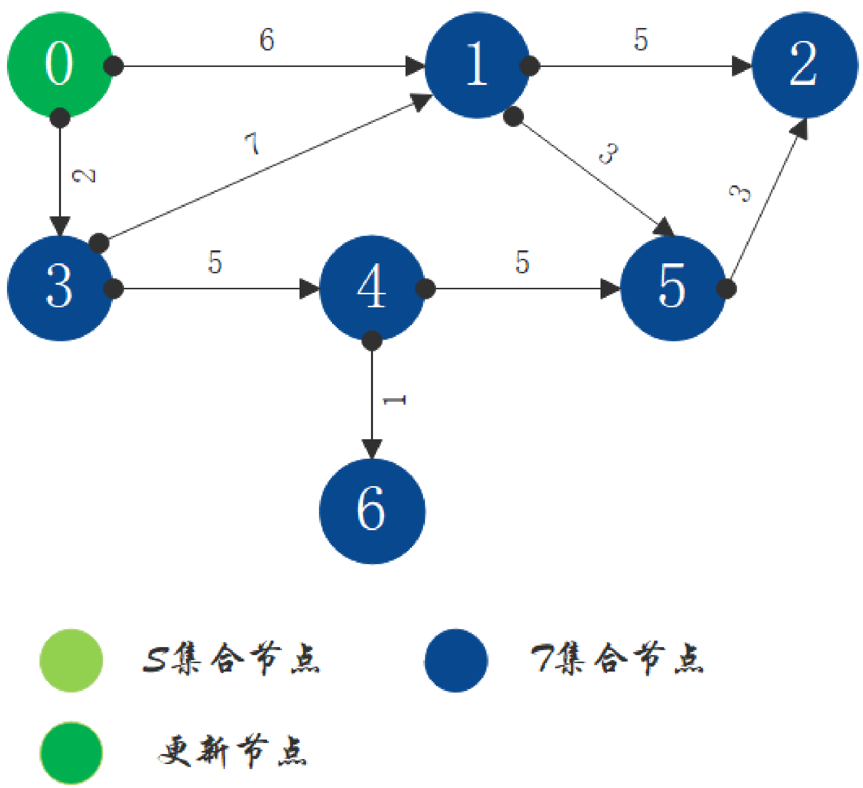


设置Graph为包含7个顶点和9条边的有向无环图，源点为0，计算从源点0到剩余节点的最短路径。每个节点将维护shortest和visited两个数据结构，shortest存储v0到该节点的最短路径，visited存储v0到该节点的最短路径是否求出。S为已求出最短路径的节点，T为未求出最短路径的节点。源节点只允许将S中的节点作为中间节点来计算到达其它节点的最短路径，不允许将T中的节点作为中间节点来计算到达其它节点的最短路径。随着S中节点的增加，源节点可达的节点才会增加。初始状态下，源节点只可达节点1和节点3。



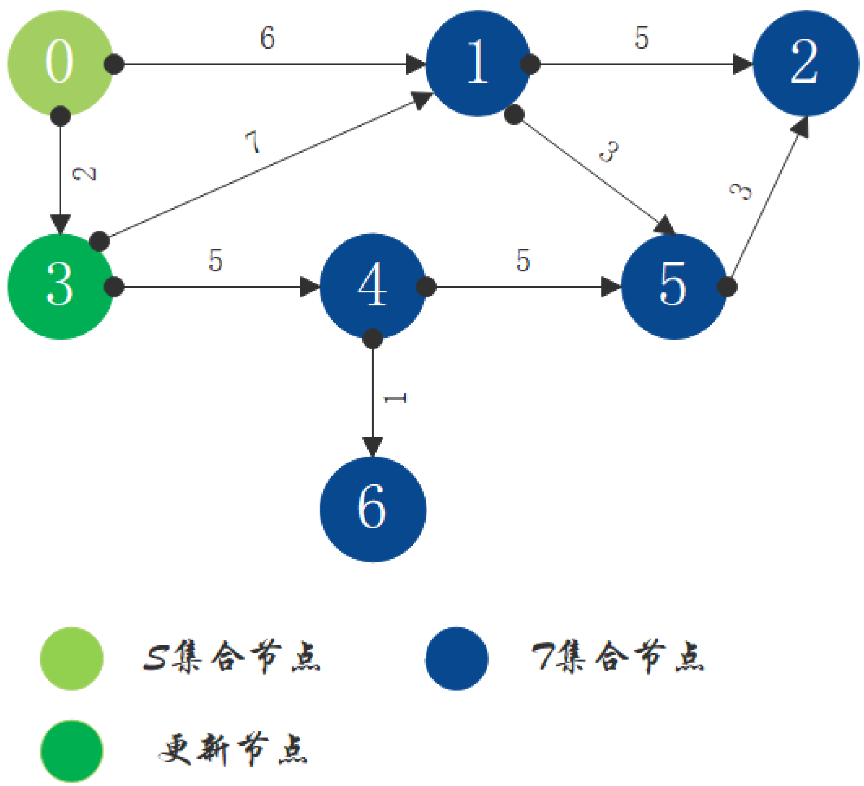
https://blog.csdn.net/qq_34842671/article/details/90083037

1、将源节点（即节点0）加入S中，对shortest和visited数组进行更新。



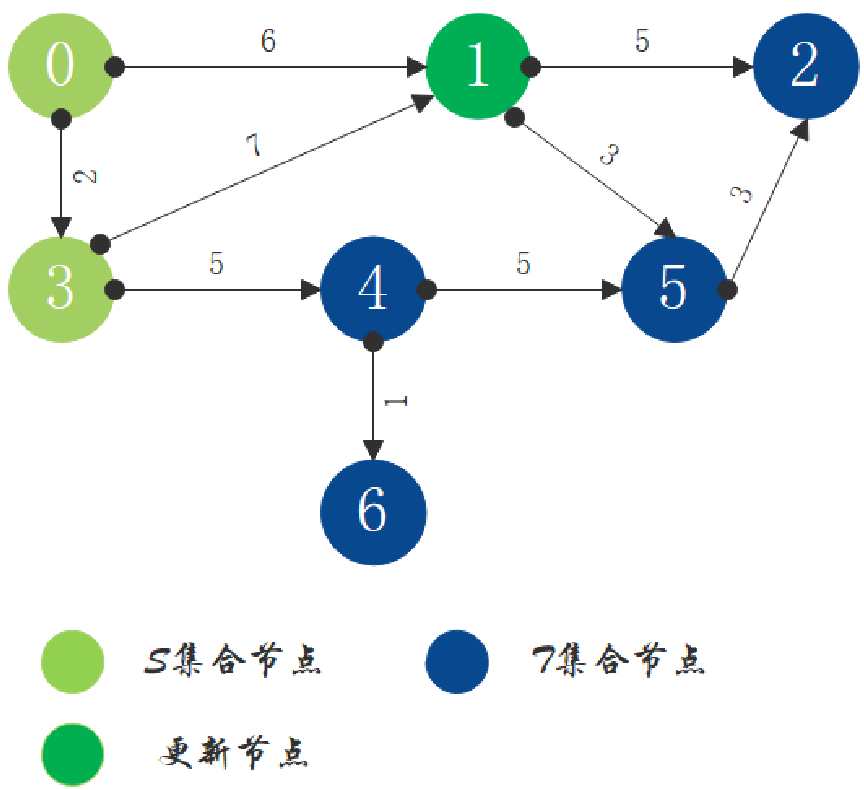
节点	shortest	visited
0	0	true
1	0	false
2	0	false
3	0	false
4	0	false
5	0	false
6	0	false

2、S中现有节点0，源节点可达T中的节点1和节点3，节点0->节点1距离为6，节点0->节点3距离为2，按距离从小到大排序，因此选择将节点3加入S中。更新源点将节点3作为中间节点到达其它节点的距离。



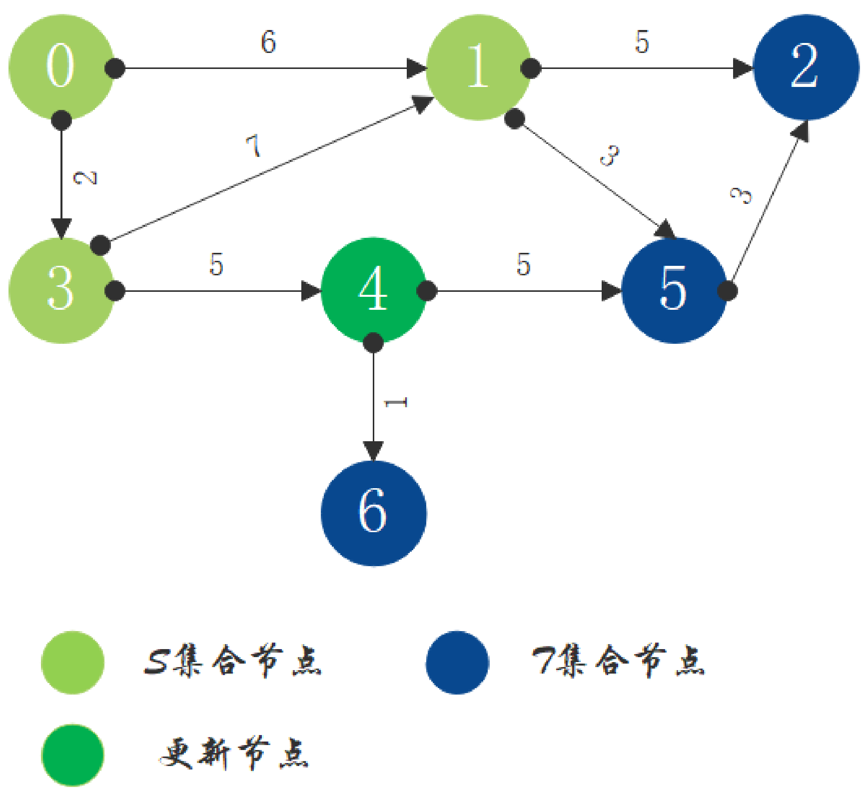
节点	shortest	visited
0	0	true
1	0	false
2	0	false
3	2	true
4	0	false
5	0	false
6	0	false

3、S中现有节点0和节点3，源节点可达T中的节点1和4，节点0->节点1距离为6，节点0->节点4距离为7，按距离从小到大排序，因此选择将节点1加入S中。更新源点将节点1作为中间节点到达其它节点的距离。



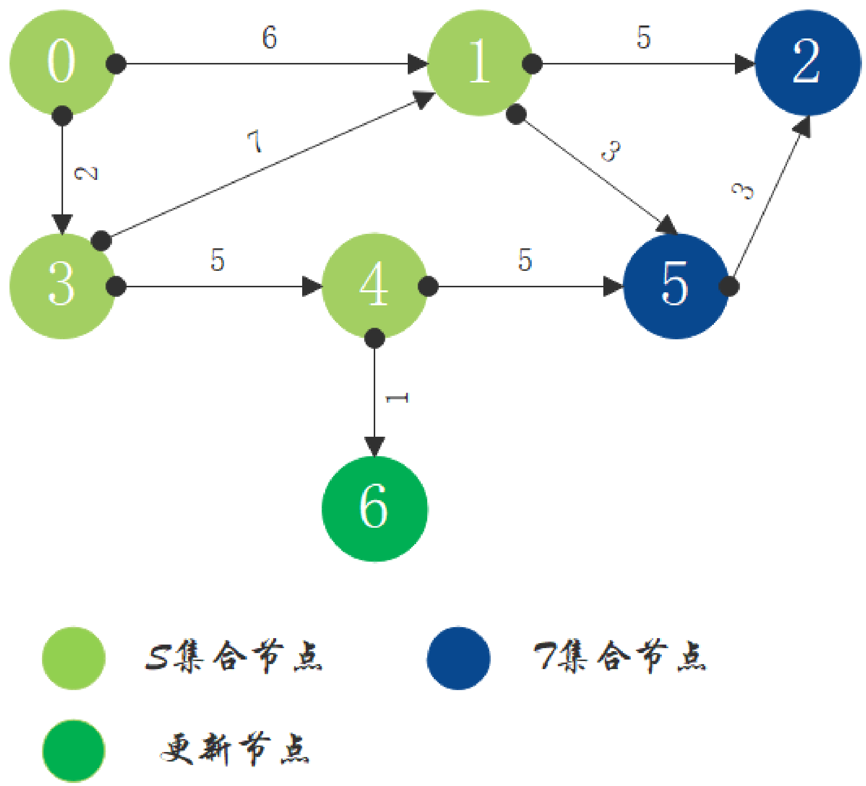
节点	shortest	visited
0	0	true
1	6	true
2	0	false
3	2	true
4	0	false
5	0	false
6	0	false

4、S中现有节点0、1、3，源节点可达T中的节点2、4、5，0->2距离为11，0->4距离为7，0->5距离为9，按距离从小到大排序，因此选择将节点4加入S中。更新源点将节点4作为中间节点到达其它节点的距离。



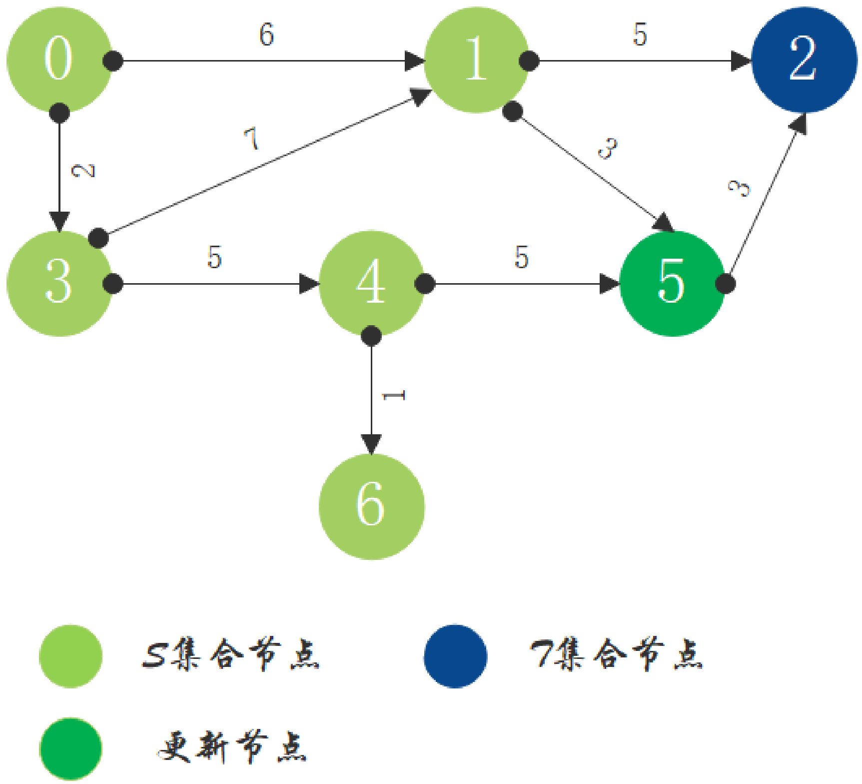
节点	shortest	visited
0	0	true
1	6	true
2	0	false
3	2	true
4	7	true
5	0	false
6	0	false

5、S中现有节点0、1、3、4，源节点可达T中的节点2、5、6，0->2距离为11，0->5距离为9，0->6距离为8，按距离从小到大排序，因此选择将节点6加入S中。更新源点将节点6作为中间节点到达其它节点的距离。



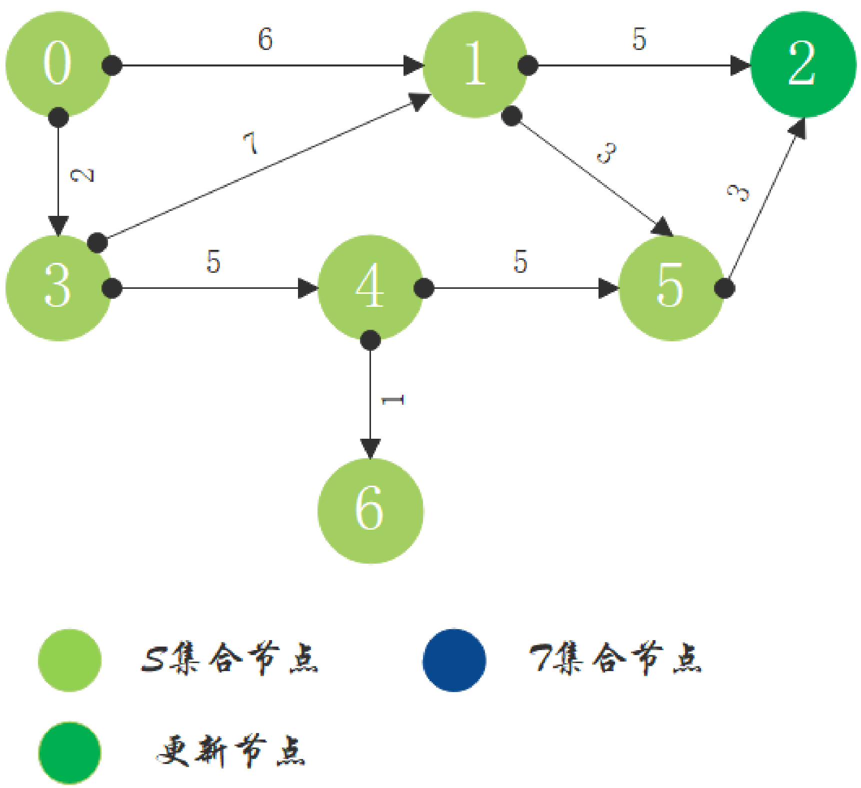
节点	shortest	visited
0	0	true
1	6	true
2	0	false
3	2	true
4	7	true
5	0	false
6	8	true

6、S中现有节点0、1、3、4、6，源节点可达T中的节点2、5，0->2距离为11，0->5距离为9，按距离从小到大排序，因此选择将节点5加入S中。更新源点将节点5作为中间节点到达其它节点的距离。



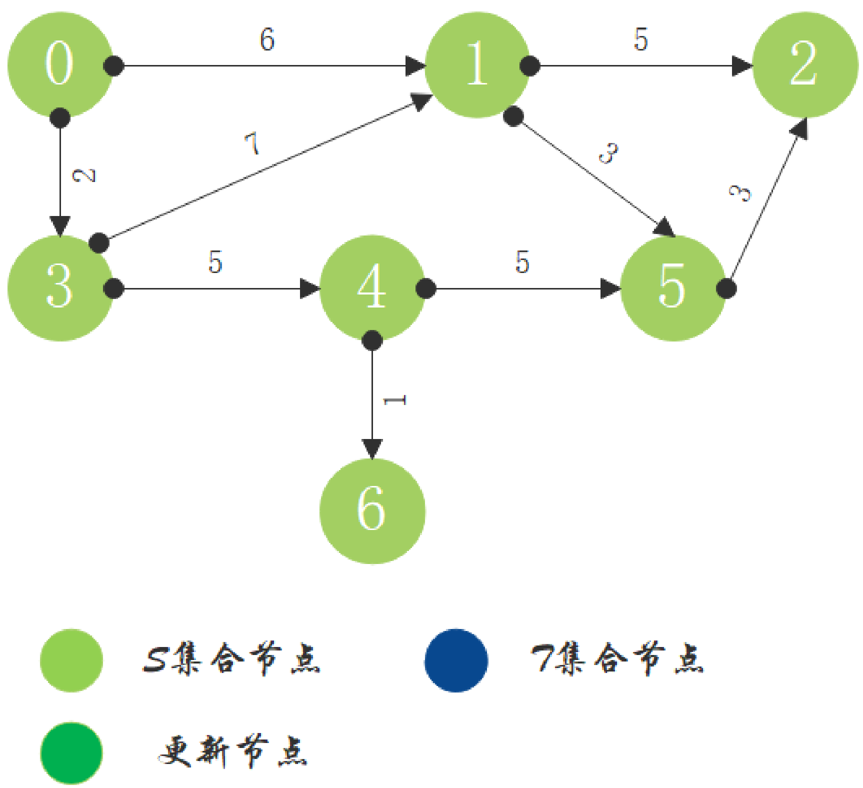
节点	shortest	visited
0	0	true
1	6	true
2	0	false
3	2	true
4	7	true
5	9	true
6	8	true

7、T中只剩下节点2，0->2距离为11，将节点2加入S中。



节点	shortest	visited
0	0	true
1	6	true
2	11	true
3	2	true
4	7	true
5	9	true
6	8	true

8、算法结束，源点到其它节点的最短路径都已依次求出。



节点	shortest	visited
0	0	true
1	6	true
2	11	true
3	2	true
4	7	true
5	9	true
6	8	true

(3)最短路径算法扩展

- 放松最短路条件
  - ✓任意值，即可能存在负数，可能有圈
  - ✓任意两点之间的最短路？
- 其他算法
  - ✓任意权值、单源：**Bellman-Ford**
  - ✓任意权值、任意两点：**Folyd-Warshall**