深圳北理莫斯科大学
УНИВЕРСИТЕТ МГУ-ППИ В ШЭНЬЧЖЭНЕ
SHENZHEN MSU-BIT UNIVERSITY

Математическое моделирование и исследование моделей с помощью математических программ

数学建模及数学软件的使用

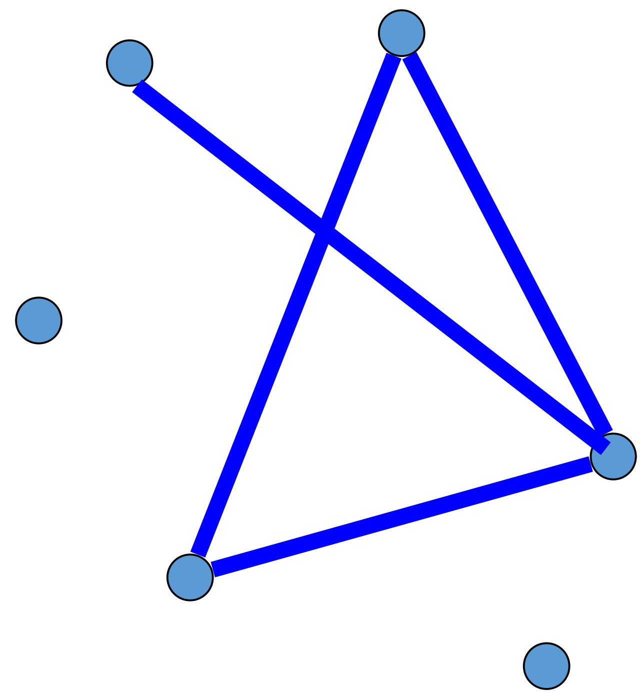
Лекция № 6 (图论&现代数学)

张晔

ye.zhang@smbu.edu.cn

什麼是图?

图论中所谓的图是由一些点(vertices (nodes)/ вершины (узлы))，和一些连接两点的边(edges/ ребра)所形成的。



- 关于图的第一篇论文是瑞士数学家欧拉（E. Euler）在1736年发表的解决“哥尼斯堡”七桥难题的论文；
- 德国的哥尼斯堡城有一条普雷格尔河，河中有两个岛屿，河的两岸和岛屿之间有七座桥相互连接，当地的居民热衷于这样一个问题，一个漫步者如何能够走过这七座桥，并且每座桥只能走过一次，最终回到原出发地。尽管试验者很多，但是都没有成功。为了寻找答案，1736年欧拉将这个问题抽象成图所示图形的一笔画问题。

前苏联

东德

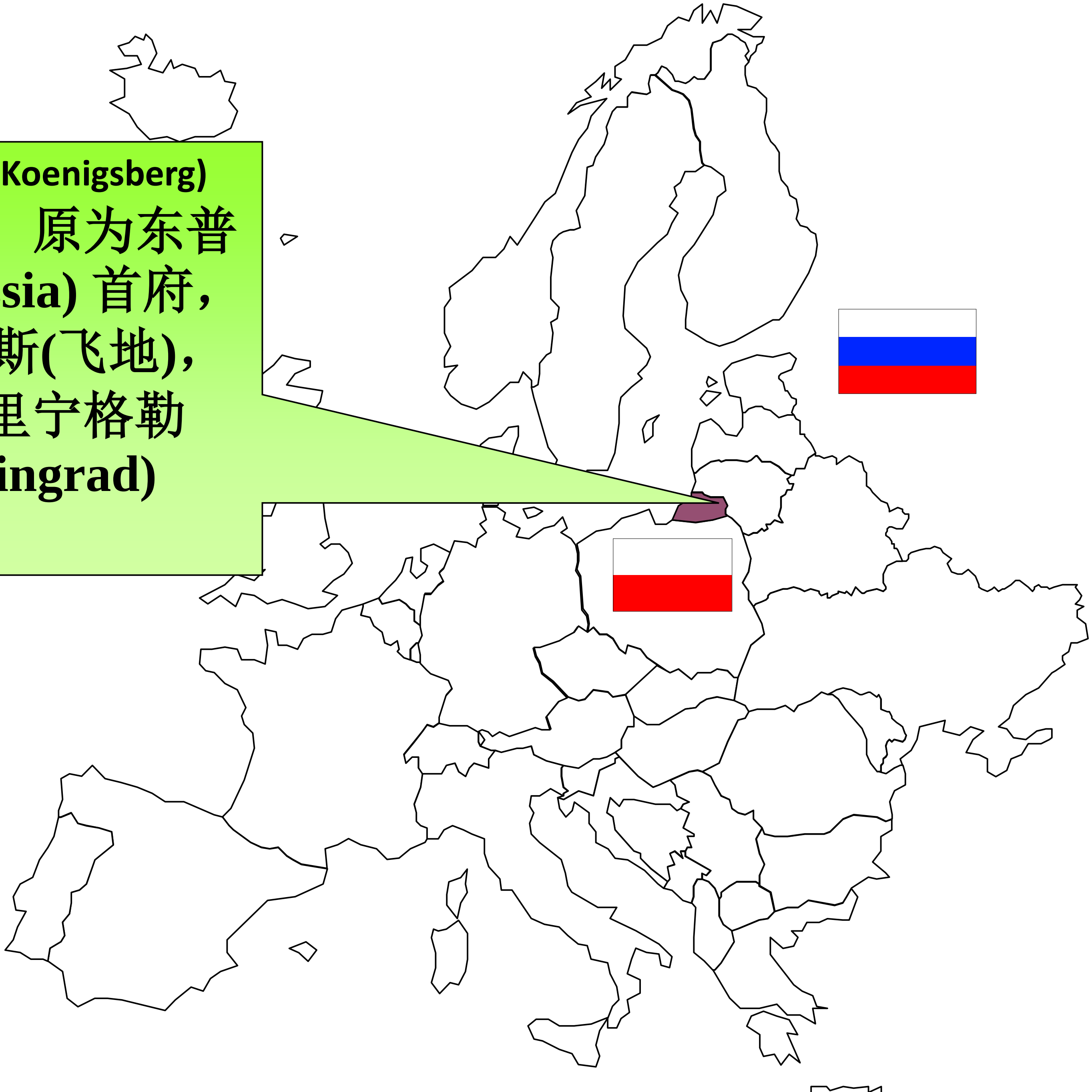
俄罗斯
欧拉学派

Эйлер (1707, Швейцария - 1783, Санкт-Петербург)

圣彼得堡学派：
Чебышёв (1821-1894)

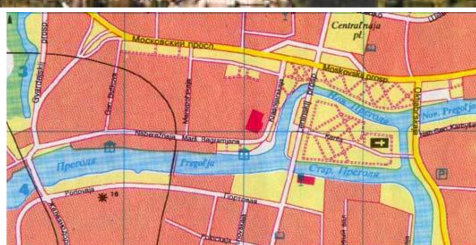
Königsberg (Koenigsberg)

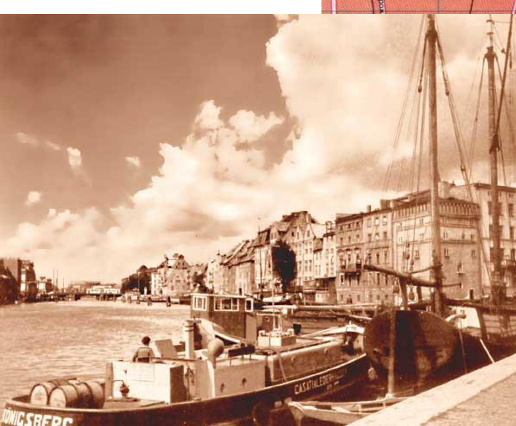
哥尼斯堡，原为东普鲁士 (Prussia) 首府，现属俄罗斯(飞地)，名为加里宁格勒 (Kaliningrad)



现德国拜仁的哥尼斯堡

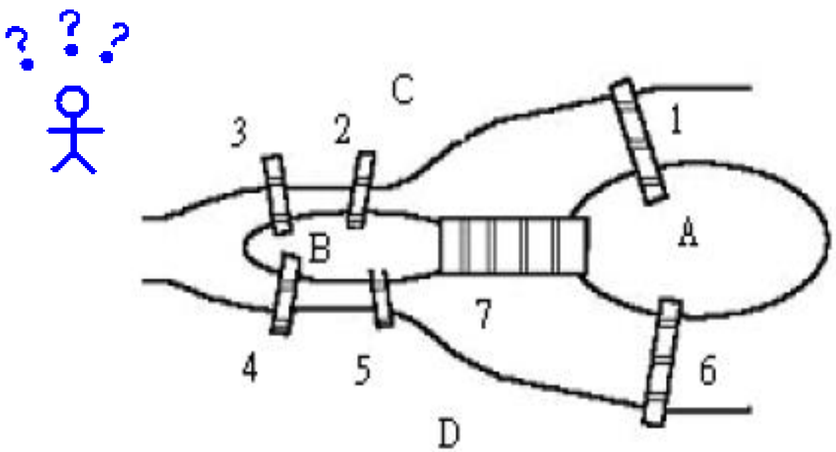
Kolding
Kiel
Schwerin
Hamburg
Szczecin (Stettin)
Bremen
Berlin
Amsterdam
Poznan (Posen)
Hannover
Potsdam
Magdeburg
Dortmund
Kassel
Leipzig
Wroclaw (Breslau)
Düsseldorf
Brüssel
Köln
Erfurt
Dresden
Frankfurt
Königsberg i.Bay.
Wiesbaden
Mainz
Prag
Luxemburg
Würzburg
Saarbrücken
Nürnberg
Mannheim
Metz
Stuttgart
Ulm
München
Linz/ Graz
Freiburg
Belfort
Salzburg
Basel
Zürich
Innsbruck

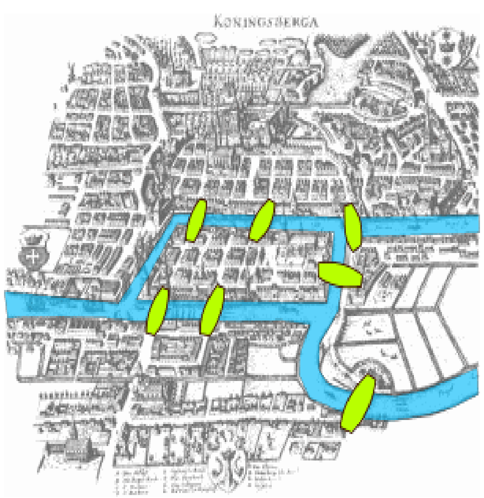


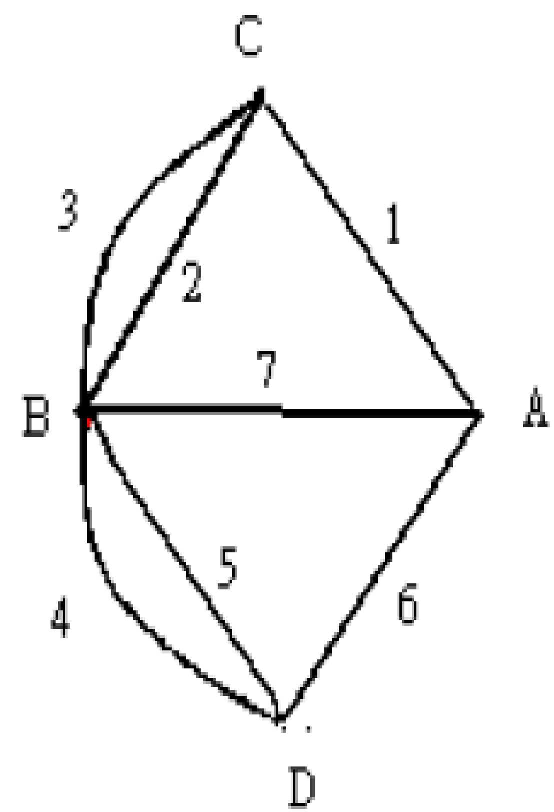


哥尼斯堡七桥问题: 要如何走过每座桥恰一次，再返回出发点?

- ✓ **Euler于1736年严格地证明了上述哥尼斯堡七桥问题无解，并且由此开创了图论的典型思维方式及论证方式。**







Euler的一笔画原理是：

1. 一笔画必须是连通的(图形的各部分之间连接在一起)；
2. 没有奇点的连通图形是一笔画，画时可以以任一偶点为起点，最后仍回到这点；
3. 只有两个奇点的连通图形是一笔画，画时必须以一个奇点为起点，以另一个奇点为终点；
4. 奇点个数超过两个的图形不是一笔画。

Замечание：

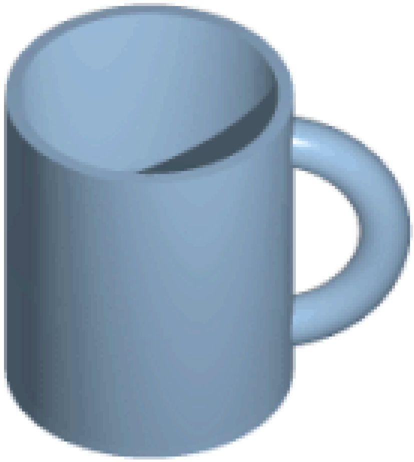
1. 一个图形的奇点数目一定是偶数。
2. 有K个奇点的图形要K÷2笔才能画成。

欧拉定理： 如果一个凸多面体的顶点数是v、棱数是e、面数是f，那么它们总有这样的关系：f+v-e=2。

- 只存在**五**种正多面体。它们是正四面体、正六面体、正八面体、正十二面体、正二十面体。

拓扑学（Топология, topology）：主要研究空间内，在连续变化（如拉伸或弯曲，但不包括撕开或黏合）下维持不变的性质。

- 代数拓扑学运用同调与同伦群等代数结构量测连通性的程度。
- 微分拓扑学研究在微分流形上的可微函数，与微分几何密切相关。
- 几何拓扑学主要研究流形与其对其他流形的嵌入。 包括“纽结理论”



在拓扑学中，

- **一个杯子和一个面包圈（实心环面）是相同的，**
- **一头母牛和一个球面也是相同的。**

集合 vs 空间：空间是有结构(运算法则)的集合

X 为一集合，且 τ 为 X 的子集族；则 τ 称之为“X 上的拓扑”，若：

1. 空集合与 X 均为 τ 的元素
2. τ 内元素间的任何并集均为 τ 的元素
3. τ 内有限多个元素间的任何交集均为 τ 的元素

若 τ 为 X 上的拓扑，则二元对 (X, τ) 称之为“拓扑空间”。

- 拓扑空间之间的函数被称为“连续”的，若任一开集合的原像均为开集合。

- “流形”（многообразие，manifold）是一个拓扑空间，在每一点附近都类似欧氏空间。
- 二维流形亦称之为曲面

定义 1.1.1 (C^r 流形). 设 M 是具有 A_2, T_2 性质的拓扑空间. 如果存在 M 的开覆盖 $\{U_\alpha\}_{\alpha\in\Gamma}$ 以及相应的连续映射族 $\varphi_\alpha : U_\alpha \to \mathbb{R}^n$, 使得

(1) $\varphi_\alpha : U_\alpha \to \varphi_\alpha(U_\alpha) \subset \mathbb{R}^n$ 为从 U_α 到欧氏空间开集 $\varphi_\alpha(U_\alpha)$ 上的同胚;

(2) 当 $U_\alpha \cap U_\beta \neq \varnothing$ 时, 如下的转换映射

$$\varphi_\beta \circ \varphi_\alpha^{-1} : \varphi_\alpha(U_\alpha \cap U_\beta) \to \varphi_\beta(U_\alpha \cap U_\beta)$$

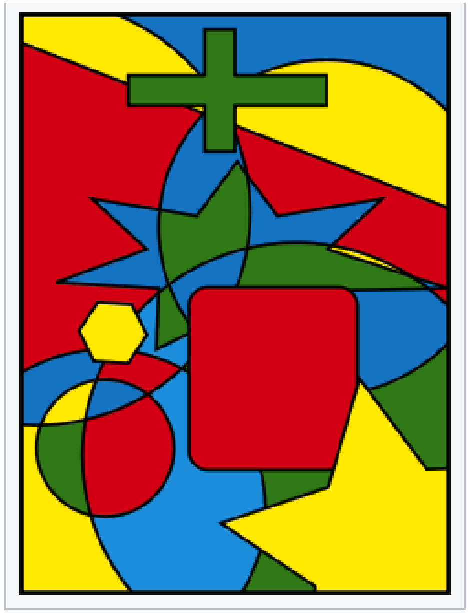
为 C^r $(r \geqslant 1)$ 映射, 则称 M 为 C^r 流形.

- 在点集拓扑中, 具有可数拓扑基的拓扑空间称为A2 的, 具有Hausdorff 性质的拓扑空间称为T2 的.

四色猜想（four color theorem, Теорема о четырёх красках）：

在一个平面或球面上的任何地图能够只用四种颜色来着色，使得没有两个相邻的国家有相同的颜色。每个国家必须由一个单连通域构成，而两个国家相邻是指它们有一段公共的边界，而不仅仅只有一个公共点。

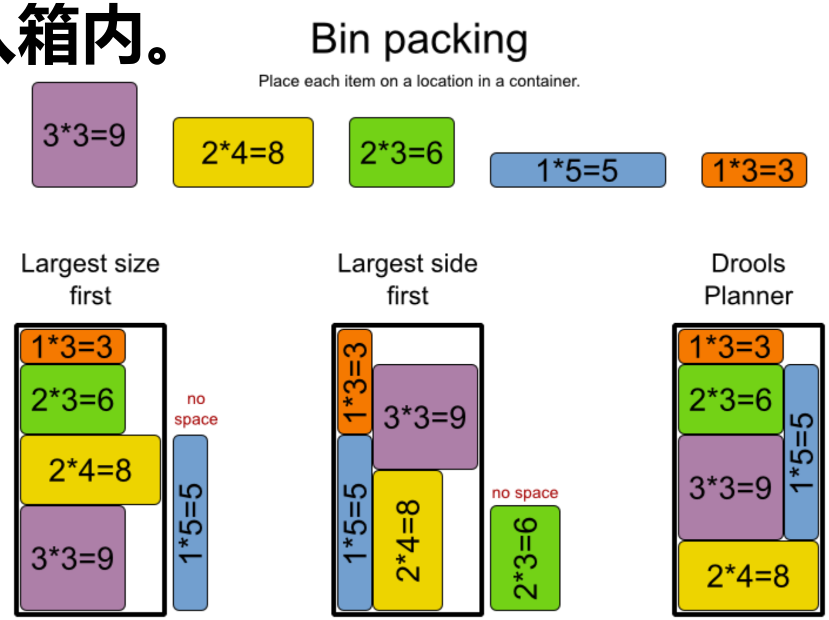
- **1976年，美国数学家阿佩尔与哈肯在美国伊利诺斯大学的两台不同的电子计算机上，用了1200个小时，作了100亿判断，终于完成了四色定理的证明。**
- **目前没有数学证明**



四色地图的一个例子

Задача об упаковке в контейнеры/ Bin packing problem/装箱问题

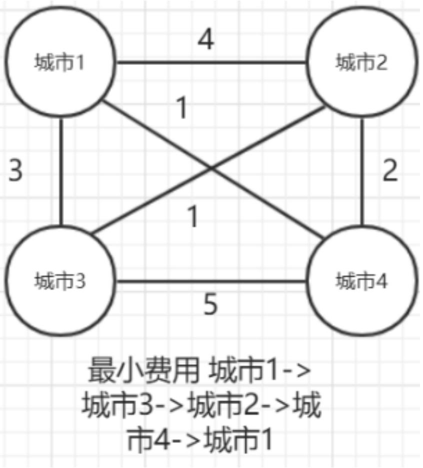
- 设有许多具有同样结构和负荷的箱子B1，B2，…，其数量足够供所达到目的之用。每个箱子的负荷（可为长度、重量等等.）为 C，今有 n 个负荷为 wj，0 < wj < C，j = 1，2，…，n 的物品 J1，J2，…，Jn 需要装入箱内。装箱问题就是指寻找一种方法，使得能以最小数量的箱子数将J1，J2，…，Jn 全部装入箱内。



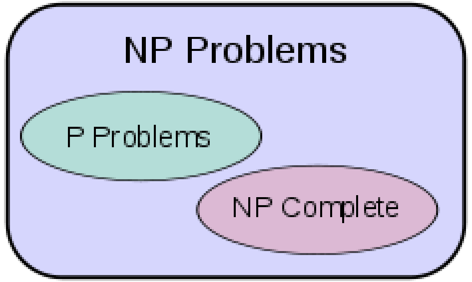
启发式算法 (heuristic algorithm)

Задача коммивояжёра/Traveling Salesman Problem/旅行商问题

- 一个售货员必须访问n个城市，这n个城市是一个完全图，售货员需要恰好访问所有城市的一次，并且回到最终的城市。 城市于城市之间有一个旅行费用，售货员希望旅行费用之和最少。



- 完全图：完全图是一个简单的无向图，其中每对不同的顶点之间都恰连有一条边相连。



- **NP完全问题**

- 复杂度类P即为所有可以由一个确定型图灵机在多项式表达的时间内解决的问题；

- 类NP由所有可以在多项式时间内验证它的解是否正确的决定问题组成，或者说，可在非确定型图灵机上以多项式时间找出解的问题的集合。

克雷数学研究所七个千禧年大奖难题

- 每解破一题可获奖金100万美元

✓ **庞加莱猜想**（*Poincaré conjecture*）：**Григо́рий Я́ковлевич Перельма́н (格里戈里·雅柯夫列维奇·佩雷尔曼)**

1. **P/NP问题**（*P versus NP problem*）
2. **霍奇猜想**（*Hodge conjecture*）
3. **黎曼猜想**（*Riemann hypothesis*）
4. **杨-米尔斯存在性与质量间隙**（*Yang–Mills existence and mass gap*）
5. **纳维-斯托克斯存在性与光滑性**（*Navier–Stokes existence and smoothness*）
6. **贝赫和斯维讷通-戴尔猜想**（*Birch and Swinnerton-Dyer conjecture*）

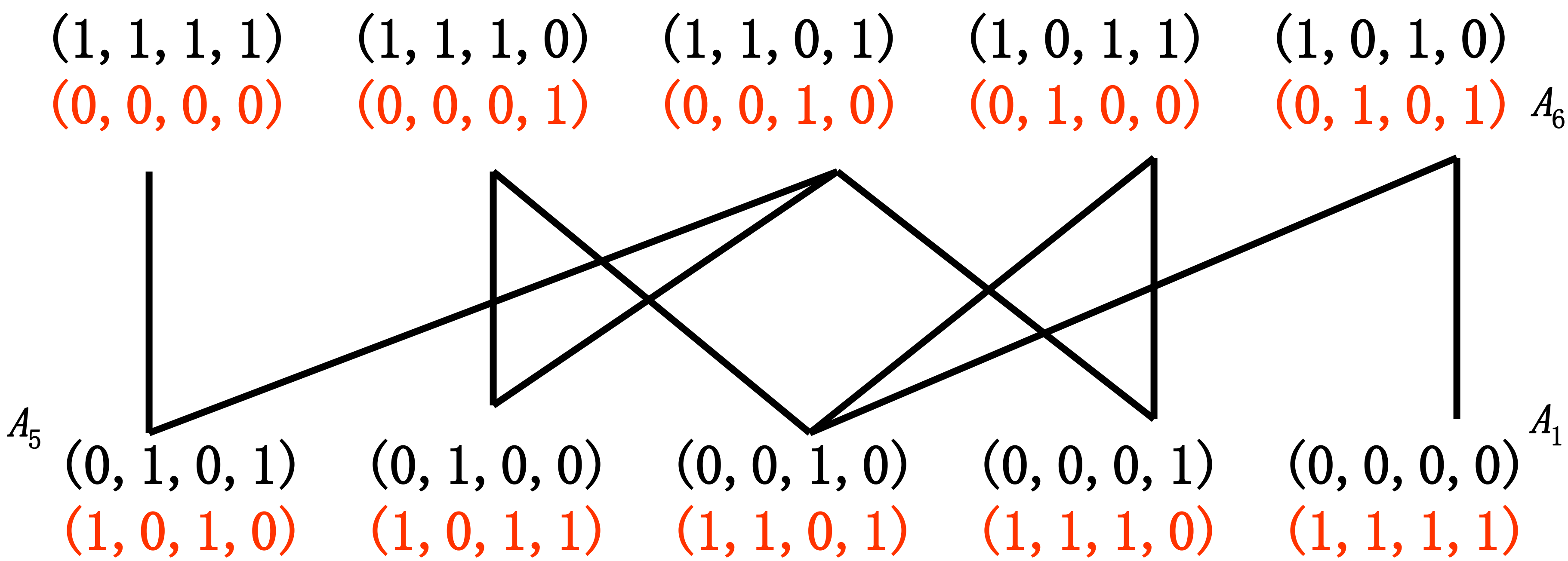
例：一摆渡人欲将一只狼,一头羊,一篮菜从河东渡过河到河西. 由于船小,一次只能带一物过河，并且狼与羊,羊与菜不能独处. 给出渡河方法.

解： 用四维0-1向量表示(人,狼,羊,菜)在河西岸的状态(在河西岸则分量取1,否则取0),共有$2^4=16$ 种状态.在河东岸的状态类似记作.

由题设,状态(0,1,1,0),(0,0,1,1),(0,1,1,1)是不允许的,从而对应状态(1,0,0,1), (1,1,0,0), (1,0,0,0)也是不允许的.

以可允许的10个状态向量作为顶点,将可能互相转移的状态用线段连接起来构成一个图.

根据此图便可找到渡河方法.



河西=(人, 狼, 羊, 菜)　河东=(人, 狼, 羊, 菜)

将10个顶点分别记为A_1，A_2，…，A_{10}，则渡河问题化为在该图中求一条从A_1到A_{10}的路.

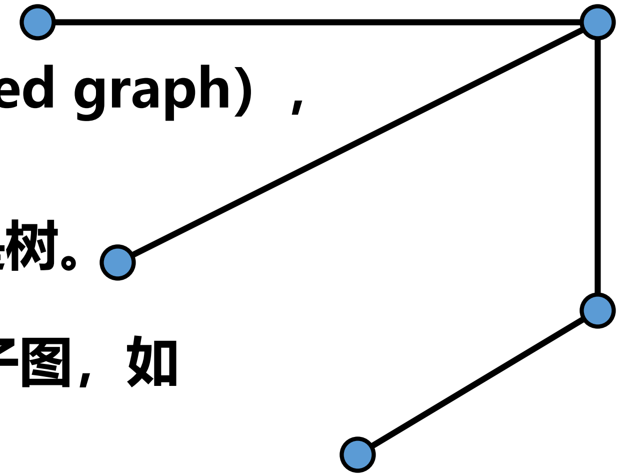
从图中易得到两条路：

最优方法：权重

$$A_1\ A_6\ A_3\ A_7\ A_2\ A_8\ A_5\ A_{10};$$

$$A_1\ A_6\ A_3\ A_9\ A_4\ A_8\ A_5\ A_{10}.$$

树



Определение：树是一种无向图（undirected graph），其中任意两个顶点间存在唯一一条路径。

- **只要没有回路的连通图（无圈的连通图）就是树。**

Опр. 设图T=(V, E’) 是图G=(V, E)的一个支撑子图，如果T是一个树，则称T是G的一个支撑树。

- **定理5：图G有支撑树的充要条件是图G是连通的。**

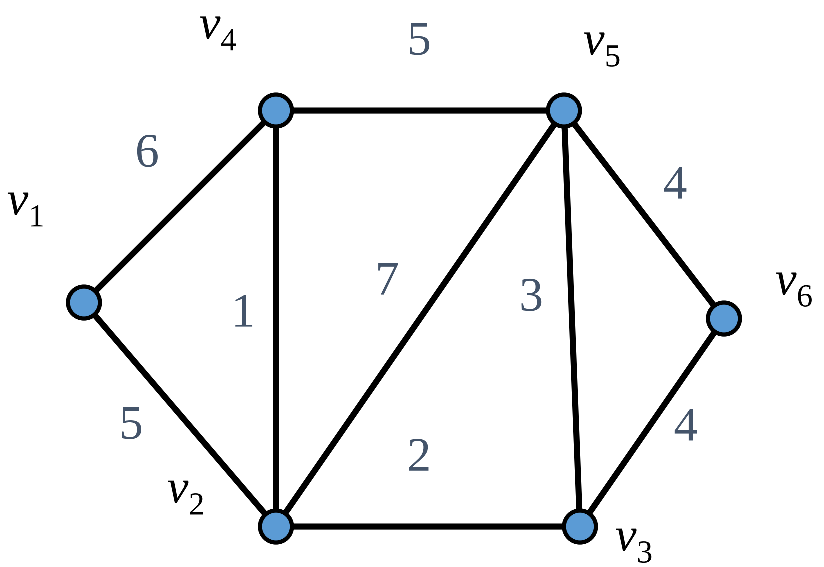
Определение:

- 给定图 $G=(V, E)$，对 G 中的每一条边 $[v_i, v_j]$，相应地有一个数 w_{ij}，则称这样的图为赋权图。
- w_{ij} 称为边 $[v_i, v_j]$上的权。
- 权是与边有关的数量指标，可以是距离、时间、费用等。
- 如果$T=(V, E')$，是 G 的一个支撑树，称E'中所有边的权之和为支撑树 T 的权，记为 $w(T)$，即：

$$\boxed{w(T) = \sum_{[v_i, v_j]\in T} w_{ij}}$$

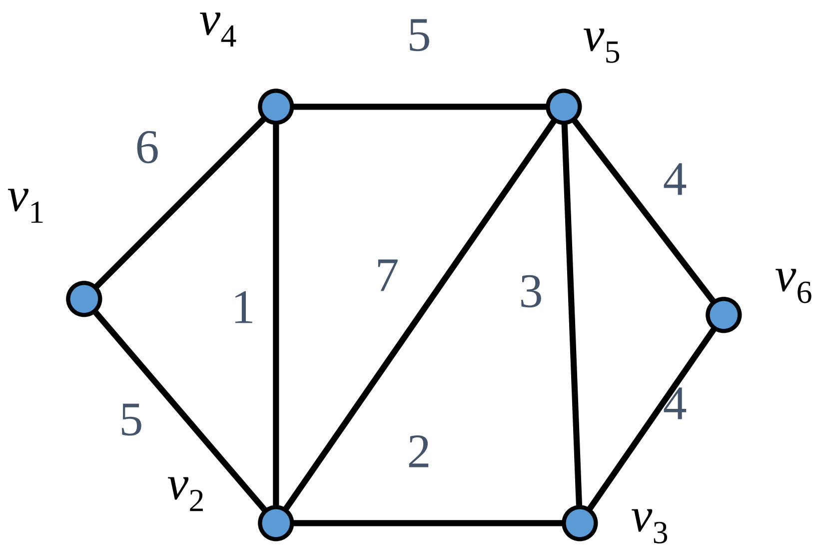
- 最小支撑树（最小树）：$\boxed{w(T^*) = \min_T w(T)}$

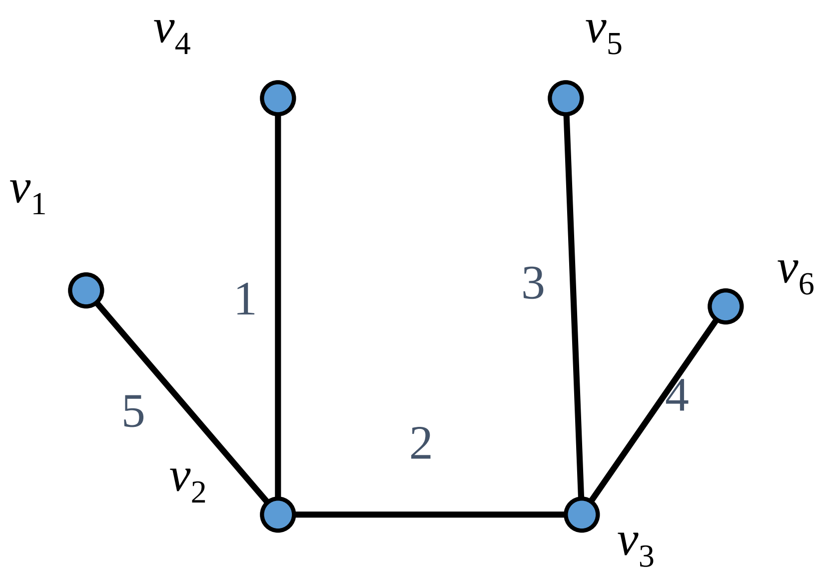
例： **某厂内联结六个车间的道路网如下所示，已知每条路的长，要求沿道路架设联结六个车间的电话线网，使电话线的总长最小。**



避圈法求最小支撑树

- i=1, E_0=Φ。从E中选最小权边。依次选最小权边，并使之不构成圈。共选5条边

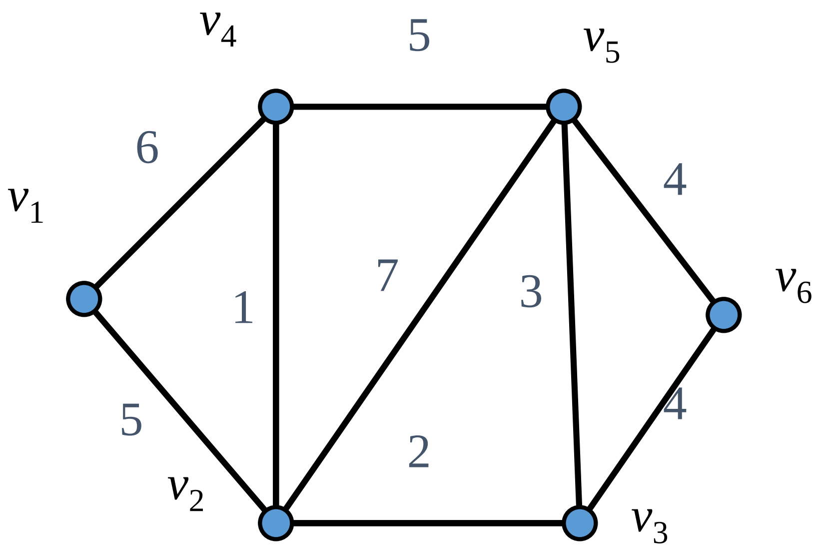




- 最后，电话线总长1+2+3+4+5=15

破圈法求最小支撑树

- 任取一个圈，从中去掉一条权最大的边。依次重复，直到不含圈为止。



- 最后，电话线总长1+2+3+4+5=15

矩阵计算方法

$$
\begin{array}{c}
 \\ \mathrm{T}
\end{array}
\quad
\begin{array}{c|cccccc}
 & v_1 & v_2 & v_3 & v_4 & v_5 & v_6 \\
\hline
v_1 & 0 & 5 & \infty & 6 & \infty & \infty \\
v_2 & 5 & 0 & 2 & 1 & 7 & \infty \\
v_3 & \infty & 2 & 0 & \infty & 3 & 4 \\
v_4 & 6 & 1 & \infty & 0 & 5 & \infty \\
v_5 & \infty & 7 & 3 & 5 & 0 & 4 \\
v_6 & \infty & \infty & 4 & \infty & 4 & 0
\end{array}
$$

矩阵计算方法

$$
\begin{array}{cc}
 & \begin{array}{cccccc} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 \end{array} \\
\begin{array}{c} \mathrm{T}\quad v_1 \\ \mathrm{T}\quad v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{array} &
\begin{bmatrix}
0 & \boxed{5} & \infty & 6 & \infty & \infty \\
5 & 0 & 2 & 1 & 7 & \infty \\
\infty & 2 & 0 & \infty & 3 & 4 \\
6 & 1 & \infty & 0 & 5 & \infty \\
\infty & 7 & 3 & 5 & 0 & 4 \\
\infty & \infty & 4 & \infty & 4 & 0
\end{bmatrix}
\end{array}
$$

矩阵计算方法

$$
\begin{array}{cc}
 & \begin{array}{cccccc} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 \end{array} \\
\begin{array}{cc} \mathrm{T} & v_1 \\ \mathrm{T} & v_2 \\ & v_3 \\ \mathrm{T} & v_4 \\ & v_5 \\ & v_6 \end{array} &
\begin{bmatrix}
0 & \boxed{5} & \infty & 6 & \infty & \infty \\
5 & 0 & 2 & \boxed{1} & 7 & \infty \\
\infty & 2 & 0 & \infty & 3 & 4 \\
6 & 1 & \infty & 0 & 5 & \infty \\
\infty & 7 & 3 & 5 & 0 & 4 \\
\infty & \infty & 4 & \infty & 4 & 0
\end{bmatrix}
\end{array}
$$

矩阵计算方法

$$
\begin{array}{cc|c}
 & & \begin{matrix} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 \end{matrix} \\
\begin{matrix} \mathrm{T} \\ \mathrm{T} \\ \mathrm{T} \\ \mathrm{T} \\ \\ \\ \end{matrix} & \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{matrix} & \begin{bmatrix} 0 & \boxed{5} & \infty & 6 & \infty & \infty \\ 5 & 0 & \boxed{2} & \boxed{1} & 7 & \infty \\ \infty & 2 & 0 & \infty & 3 & 4 \\ 6 & 1 & \infty & 0 & 5 & \infty \\ \infty & 7 & 3 & 5 & 0 & 4 \\ \infty & \infty & 4 & \infty & 4 & 0 \end{bmatrix}
\end{array}
$$

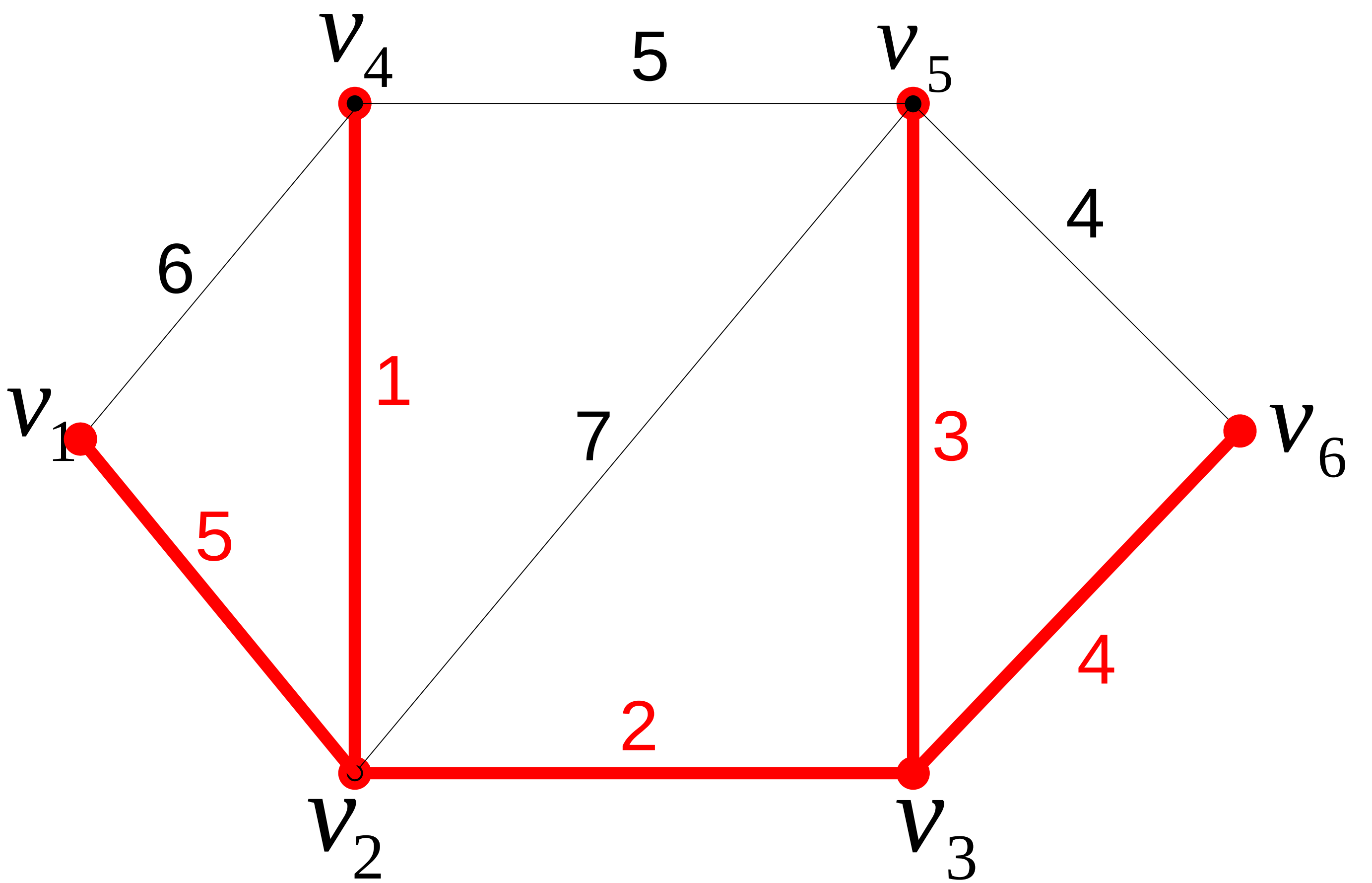
矩阵计算方法

$$
\begin{array}{cc}
 & \begin{array}{cccccc} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 \end{array} \\
\begin{array}{c} \mathrm{T}\ v_1 \\ \mathrm{T}\ v_2 \\ \mathrm{T}\ v_3 \\ \mathrm{T}\ v_4 \\ \mathrm{T}\ v_5 \\ v_6 \end{array} &
\begin{bmatrix}
0 & \boxed{5} & \infty & 6 & \infty & \infty \\
5 & 0 & \boxed{2} & \boxed{1} & 7 & \infty \\
\infty & 2 & 0 & \infty & \boxed{3} & 4 \\
6 & 1 & \infty & 0 & 5 & \infty \\
\infty & 7 & 3 & 5 & 0 & 4 \\
\infty & \infty & 4 & \infty & 4 & 0
\end{bmatrix}
\end{array}
$$

矩阵计算方法

$$
\begin{array}{ccc}
 & & \begin{matrix} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 \end{matrix} \\
\begin{matrix} \mathrm{T} \\ \mathrm{T} \\ \mathrm{T} \\ \mathrm{T} \\ \mathrm{T} \\ \mathrm{T} \end{matrix} &
\begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{matrix} &
\begin{bmatrix}
0 & \boxed{5} & \infty & 6 & \infty & \infty \\
5 & 0 & \boxed{2} & \boxed{1} & 7 & \infty \\
\infty & 2 & 0 & \infty & \boxed{3} & \boxed{4} \\
6 & 1 & \infty & 0 & 5 & \infty \\
\infty & 7 & 3 & 5 & 0 & 4 \\
\infty & \infty & 4 & \infty & 4 & 0
\end{bmatrix}
\end{array}
$$

矩阵计算结果



Задача о кратчайшем пути/ Shortest path problem/最短路问题

- **确定起点的最短路径问题** - 即已知起始结点，求最短路径的问题。适合使用Dijkstra算法。
- **确定终点的最短路径问题** - 与确定起点的问题相反，该问题是已知终结结点，求最短路径的问题。在无向图中该问题与确定起点的问题完全等同，在有向图中该问题等同于把所有路径方向反转的确定起点的问题。
- **确定起点终点的最短路径问题** - 即已知起点和终点，求两结点之间的最短路径。通常可以用广度优先搜索（BFS）、深度优先搜索（DFS）等方式来实现，时间复杂度是**O(|V|)**。

Задача о кратчайшем пути/ Shortest path problem/最短路问题

- **全局最短路径问题** - 求图中所有的最短路径。适合使用Floyd-Warshall算法，算法时间复杂度为**O(|V|^3)**。对于稀疏图，还可采用Johnson算法，其采用Bellman-ford和Dijkstra作为其子函数，时间复杂度为**O(VE logV)**。二者都可计算含负权路径的图，但不可含有负环。

无向图:

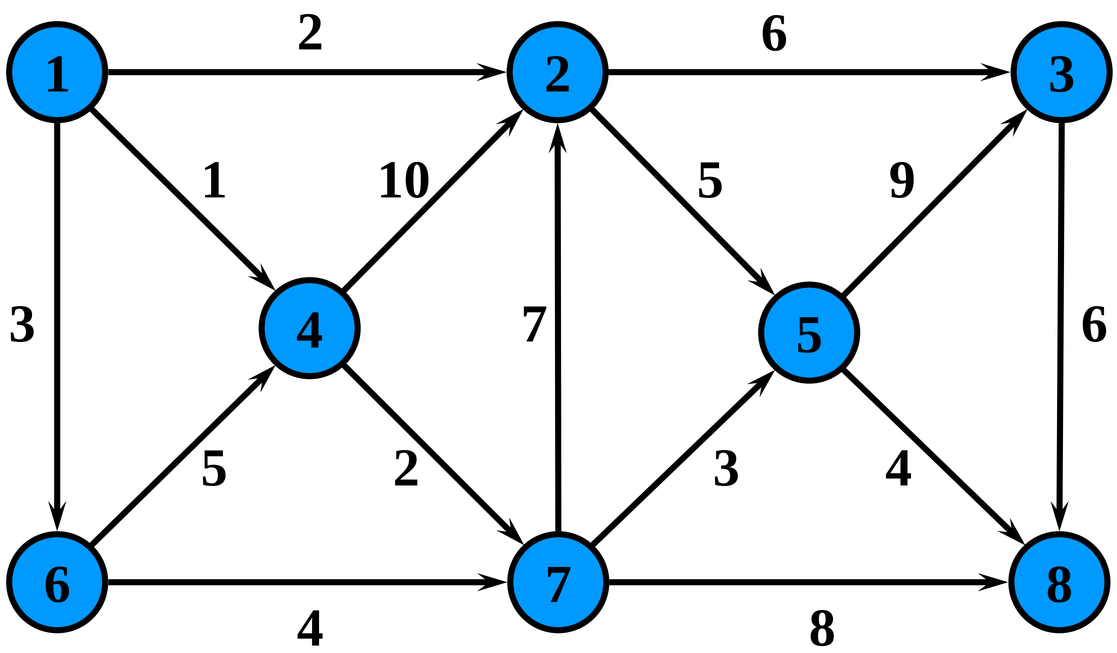
权值要求	时间复杂度	作者
$\mathbb{R}_+$	$O(V^2)$	Dijkstra 1959
$\mathbb{R}_+$	$O((E+V)logV)$	Johnson 1977 (二叉堆)
$\mathbb{R}_+$	$O(E+VlogV)$	Fredman & Tarjan 1984 (斐波那契堆)
$\mathbb{N}$	$O(E)$	Thorup 1999 (要求常数时间复杂度的乘法)。

Dijkstra法

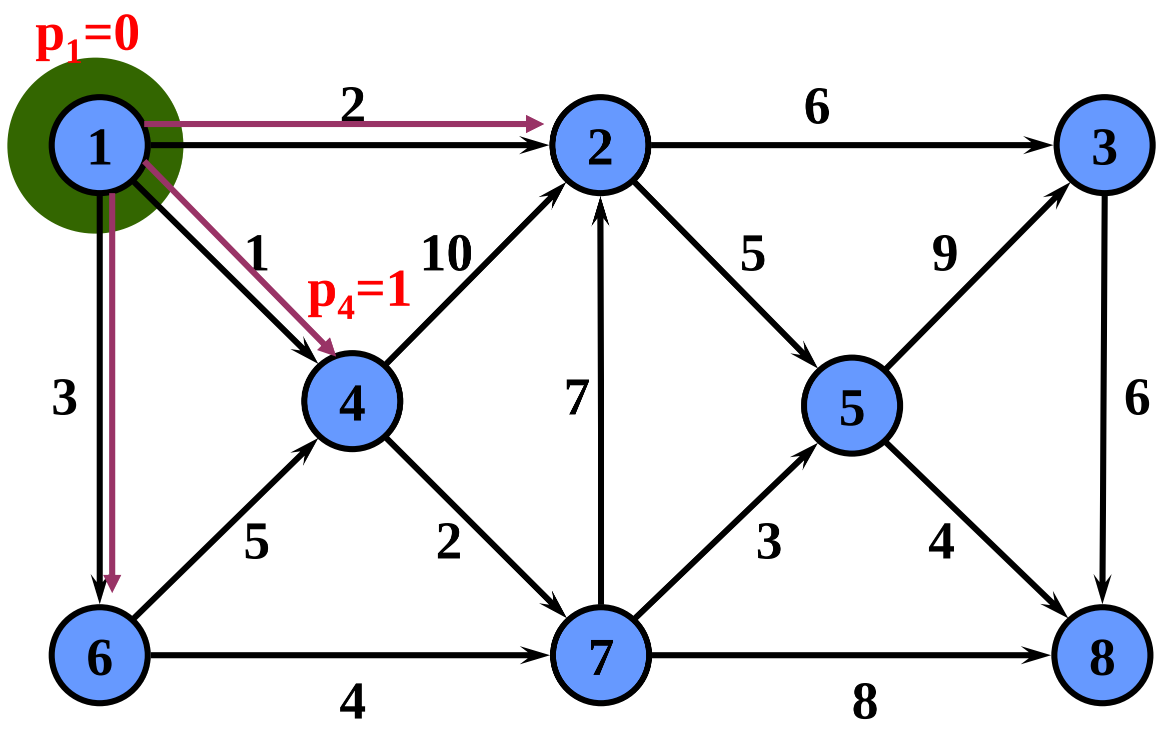
基本思路：从v_s出发，逐步地向外探寻最短路。

- 执行过程中，与每个点对应，记录下一个数 (称为此点的标号)，
  - 它或者表示从v_s到该点的最短路的权 (称为P标号)，
  - 或者是从v_s到该点的最短路的权的上界 (称为T标号)
- 方法的每一步是修改T标号，并且把某一个具T标号的点改变为具P标号的点，从而使D中具P标号的点多一个，
- 如此经过 p-1 步，就可以求出从v_s到各点的最短路。

求从1到8的最短路径



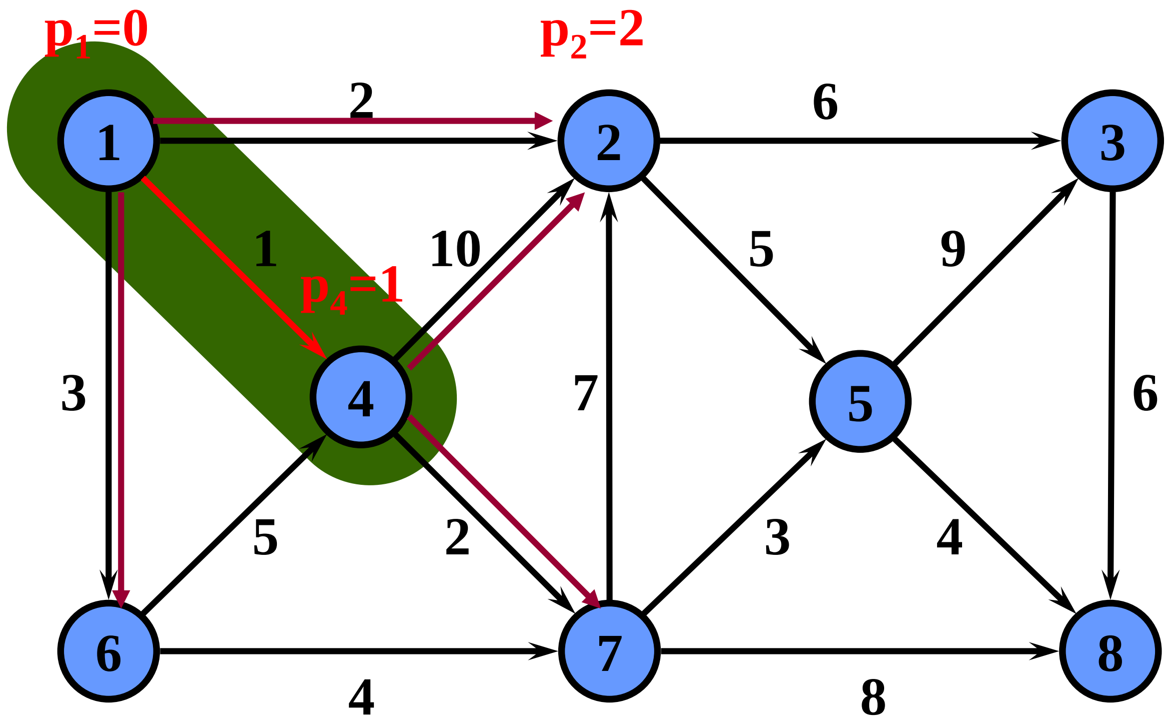
$X=\{1\}, w_1=0$



$\min\{c_{12},c_{14},c_{16}\}=\min\{0+2,0+1,0+3\}=\min\{2,1,3\}=1$

$X=\{1,4\}, p_4=1$

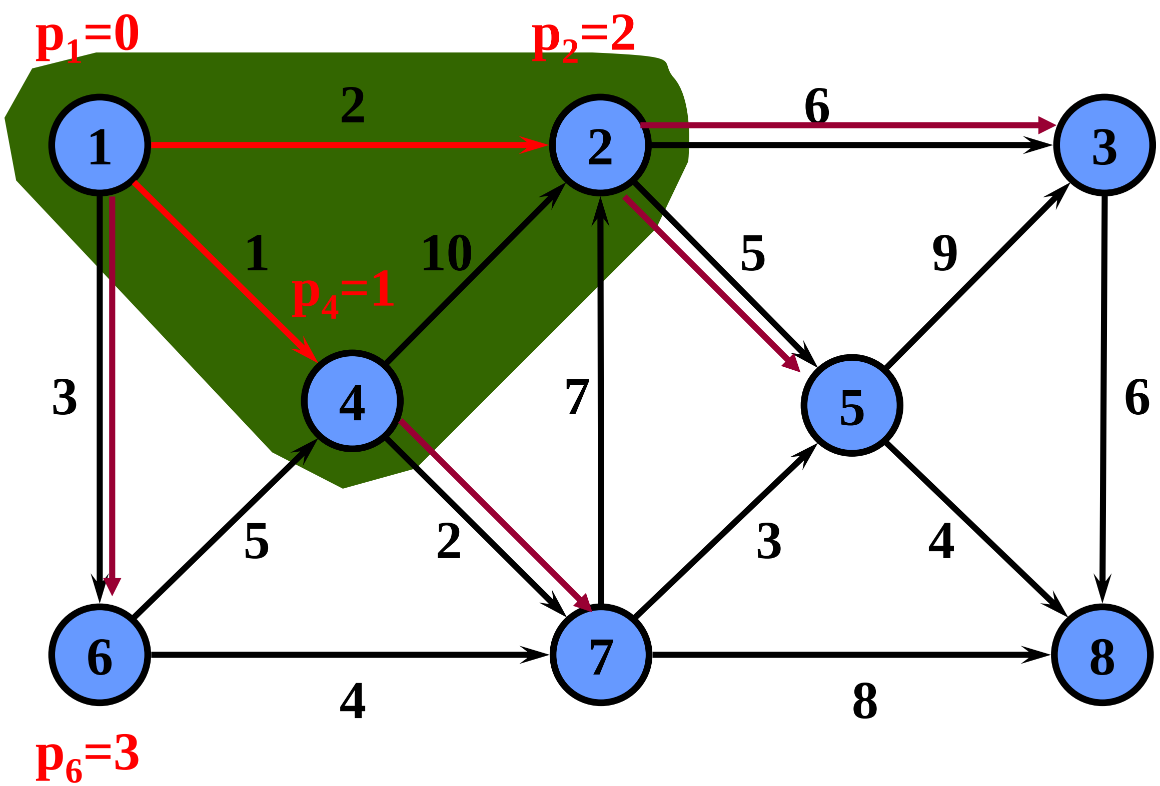
$X=\{1,4\}$



$\min\{c_{12},c_{16},c_{42},c_{47}\}=\min\{0+2,0+3,1+10,1+2\}=\min\{2,3,11,3\}=2$

$X=\{1,2,4\}$, $p_2=2$

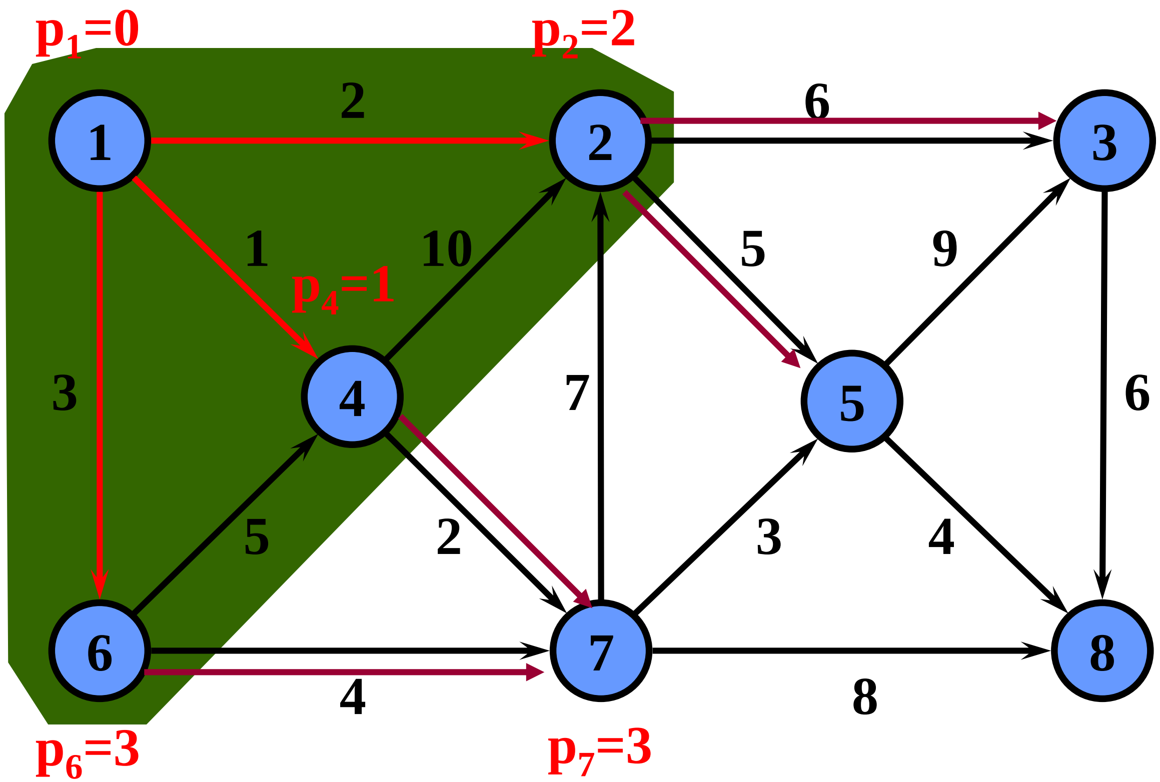
$X=\{1,2,4\}$



$\min\{c_{16},c_{23},c_{25},c_{47}\}=\min\{0+3,2+6,2+5,1+2\}=\min\{3,8,7,3\}=3$

$X=\{1,2,4,6\}$, $p_6=3$

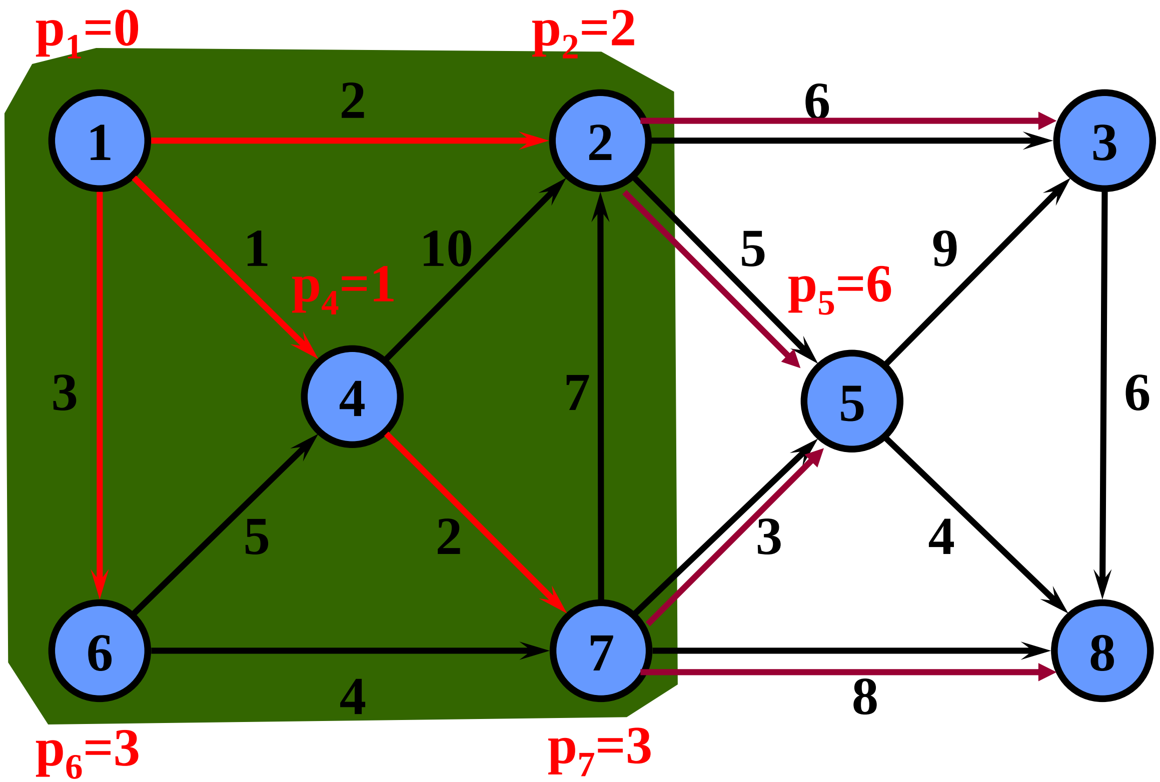
$X=\{1,2,4,6\}$



$\min\{c_{23},c_{25},c_{47},c_{67}\}=\min\{2+6,2+5,1+2,3+4\}=\min\{8,7,3,7\}=3$

$X=\{1,2,4,6,7\}$, $p_7=3$

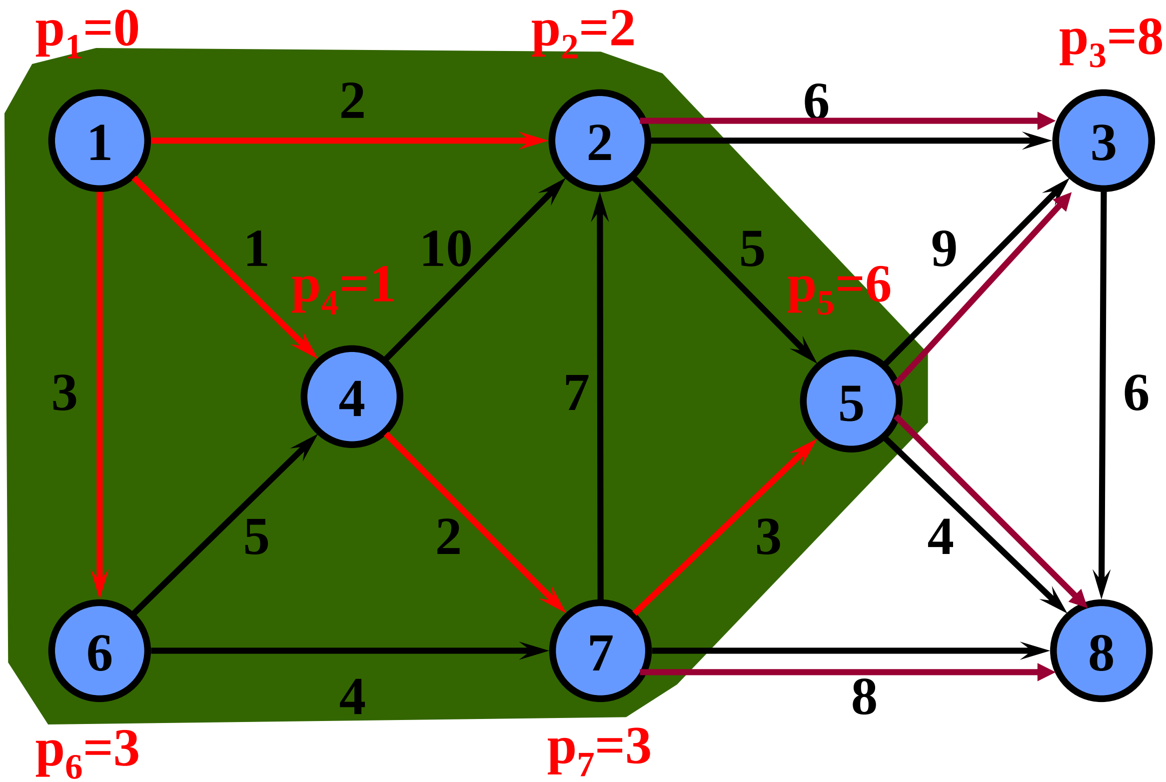
$X=\{1,2,4,6,7\}$



$\min\{c_{23},c_{25},c_{75},c_{78}\}=\min\{2+6,2+5,3+3,3+8\}=\min\{8,7,6,11\}=6$

$X=\{1,2,4,5,6,7\}$, $p_5=6$

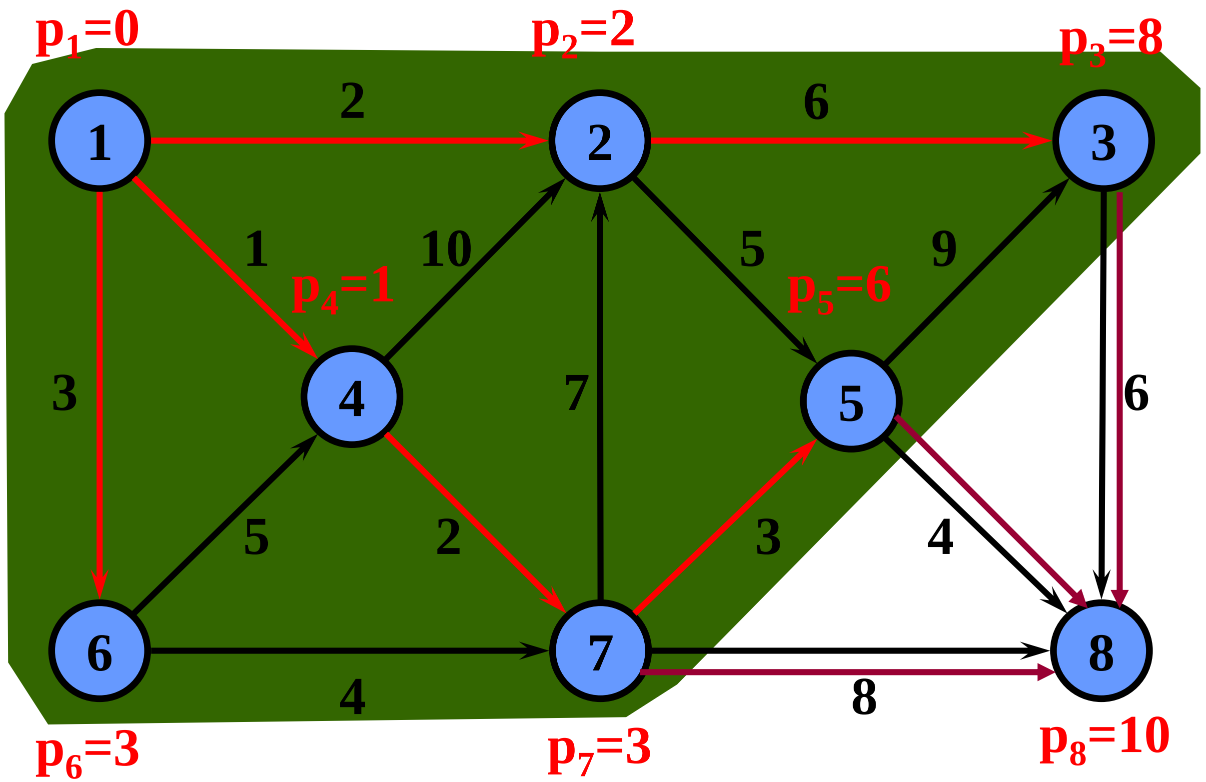
$X=\{1,2,4,6,7\}$



$\min\{c_{23}, c_{53}, c_{58}, c_{78}\} = \min\{2+6, 6+9, 6+4, 3+8\} = \min\{8, 15, 10, 11\} = 8$

$X=\{1,2,3,4,5,6,7\}$, $p_3=8$

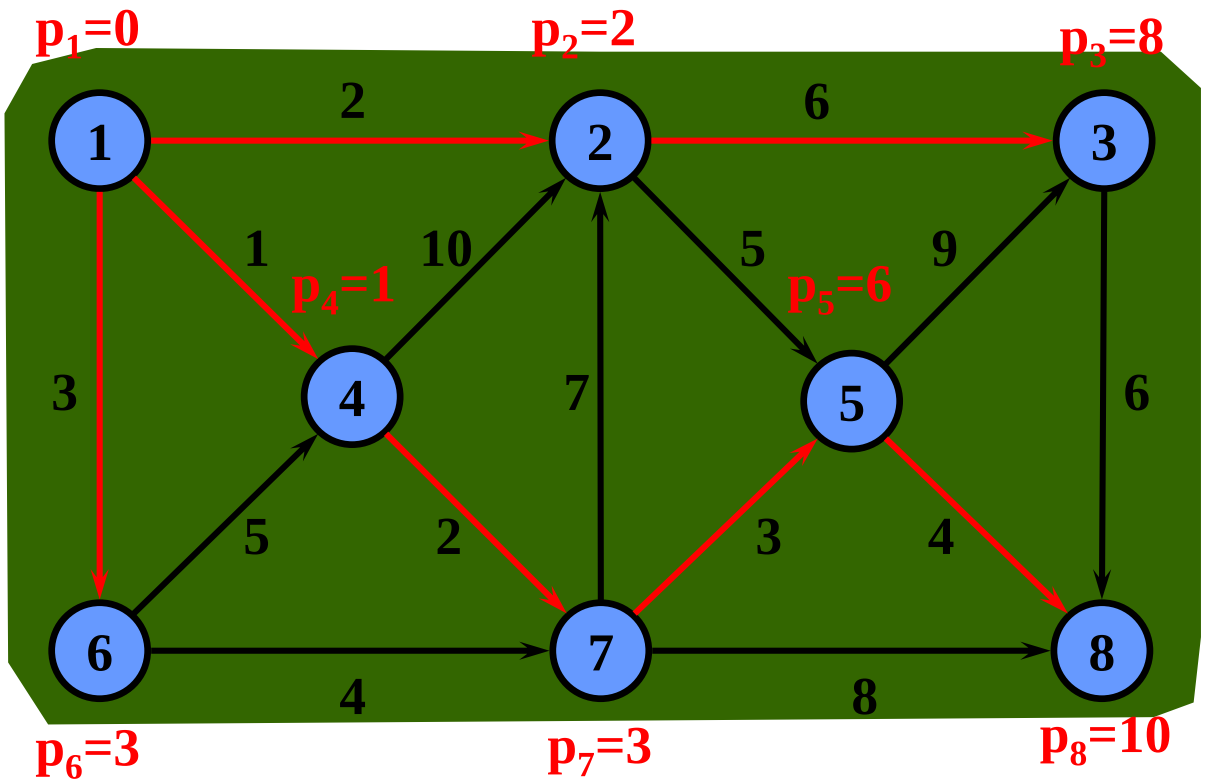
$X=\{1,2,3,4,6,7\}$



$\min\{c_{38},c_{58},c_{78}\}=\min\{8+6,6+4,3+7\}=\min\{14,10,11\}=10$

$X=\{1,2,3,4,5,6,7,8\}$, $p_8=10$

X={1,2,3,4,6,7,8}



1到8的最短路径为{1，4，7，5，8}，长度为10。