数学建模模型算法精讲课——

图论

—— 江北老师

当蜘蛛网无情地查封了我的炉台，当灰烬的余烟叹息着贫困的悲哀

我依然固执地铺平失望的灰烬，用美丽的雪花写下：相信未来

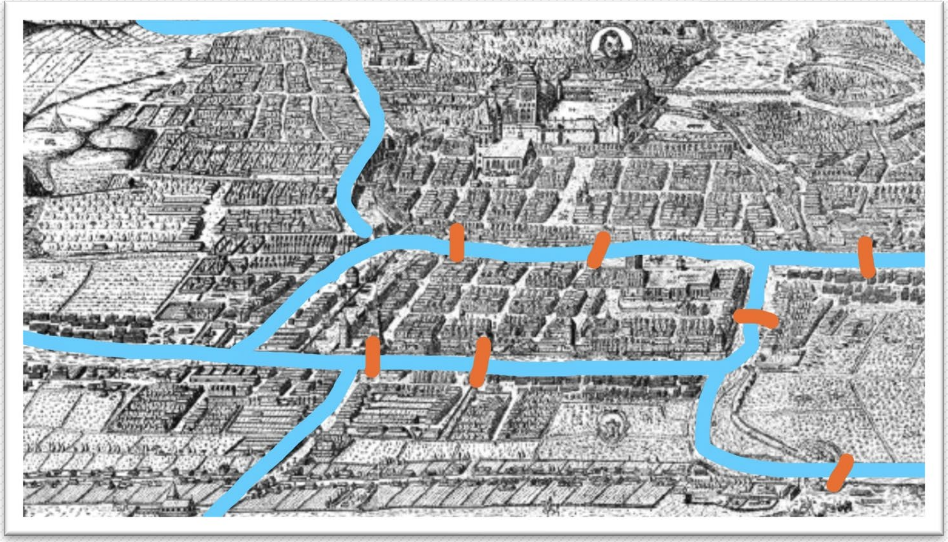
图论

- 概念引出
- 基本概念
- 代码讲解

七桥问题

十八世纪，在今天俄罗斯加里宁格勒市还被称为柯尼斯堡的年代。像其他许多大城市一样，一条大河(普列戈利亚河)穿城而过。柯尼斯堡除了被一分为二以外，还包含河中的两个岛屿，人们建有七座桥梁连接着不同的陆地。

- 当时有一个著名的游戏谜题，就是在所有桥都**只能走一遍**的前提下，怎样才能把这片区域所有的桥都走遍？
- 这个谜题成为当地人一项热衷的消遣运动，许多人声称已经找到了这样一条路径，但当被要求按照规则再走一遍时，却发现还是没有人能够做到。

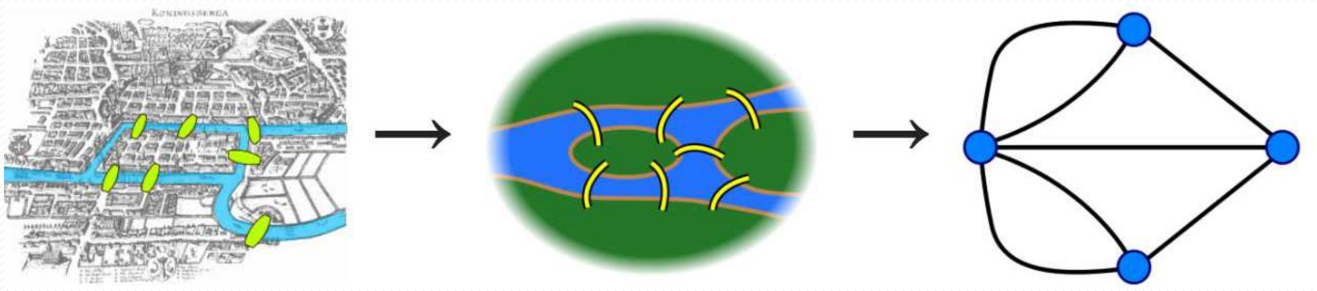


图论——概念引出

➢ 欧拉与柯尼斯堡的七桥问题：图论的诞生

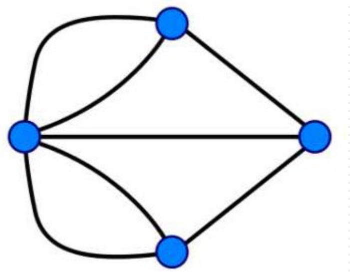
直到 1736 年，瑞士数学家莱昂哈德·欧拉给出了答案，他在论文《柯尼斯堡的七桥》中解释了其中的原因，证明这样的步行方式**并不存在**，并在文中提出和解决了一笔画问题。

- 欧拉的解决方案出乎意料地简单--只要我们以正确的方式看待这个问题就容易得到答案。
- 诀窍是要剔去所有不必要的信息。比如，在不同的陆地上走哪条路并不重要。不要管陆地是什么形状，河流是什么形状，桥梁是什么形状，通通这些都不重要。
- 因此，不妨用一个点来表示一块陆地，用一条边来表示一座桥。根本不需要做到地理上的精确表示：只要不破坏点的连通性，两点间不要连接错误，就能以**任何方式表示地理信息而不改变问题本身**。



图论——概念引出

➢ 欧拉与柯尼斯堡的七桥问题：图论的诞生

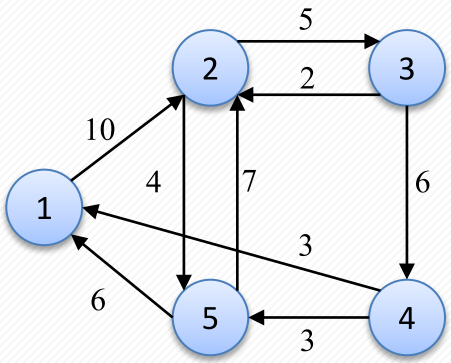


- 当通过一条边到达一个点时（通过桥进入某块陆地），那么除非它是行走的**最后一个点**，否则要再次离开它时，就得通过**不同的边**，因为这是游戏的规则。也就是说，任何不是起点和终点的点都需要有**偶数条边从它那里出来**：每进入一条线，就必须有一条边能离开。
- 为了使每条边都能准确穿过一次，那么最多只能有**两个点有奇数个边出来**。事实上，要么有两个奇数的点，要么根本没有。在前一种情况下，这两个点对应于步行的**起点和终点**，而在后一种情况下，起点和终点是**同一点**。然而，在柯尼斯堡问题中，4 个点都有奇数条边，所以穿过每座桥的步行**是不可能的。**
- 用现在图论的术语来说，柯尼斯堡七桥问题属于一笔画问题：如何判断这个图是否是一个能够遍历完所有的边而没有重复。如果存在这样方法，该图则称为**欧拉图**。这时遍历的路径称作欧拉路径（一个环或者一条链），如果路径闭合（一个圈），则称为**欧拉回路**。
- 欧拉的这个结论**标志着图论的诞生**，即研究由线连接的点组成的网络。他还能够证明，如果一个图满足上述条件，图中奇顶点（连接的边数量为奇数的顶点）的数目等于 0 或者 2。

➢ 图论

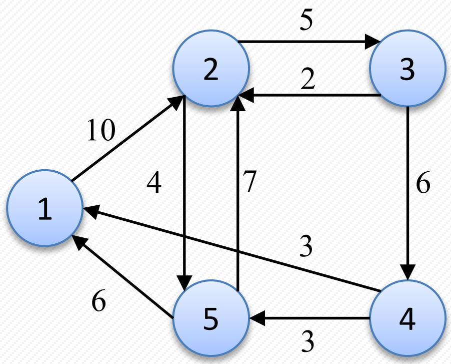
图论（ Graph Theory ）是数学的一个分支。它以图为研究对象。图论中的图是由若干给定的点及连接两点的线所构成的图形，这种图形通常用来**描述某些事物之间的某种特定关系**，用点代表事物，用连接两点的线表示相应两个事物间**具有这种关系。**

- 数学建模中的图，是根据**实际问题简化而来的模型**，对于平面上的n个点，把其中一些点用直线和曲线连接起来，不考虑所画点的位置和**所画直线或曲线的长度**，这样形成的关系称为“图”



- 图论中，**每个节点称为顶点**，连线称为边，边上标出的数字称为边的权重（权重和线段画出来的长度无关）
- 例如左图，点表示$v_1,v_2,\cdots,v_5$5个城市，点与点之间的连线表示城市间道路，有的是单向通行的，有的是双向通行的
- 左图边上标的数字，实际意义表示城市间的通行距离
- 图是实际问题中抽象出来的顶点和边的关系，画出来后，顶点位置是随意的，**边的形状和长短也是随意的**，比如说城市A到B图上画的是直线，实际上的道路可能山路十八弯

图论的基本概念



- **顶点**：用点集V表示，$V=\{v_1,v_2,\cdots,v_5\}$
- **边**：用边集E表示，$E=\{e_1,e_2,\cdots,e_n\}$
- **权重**：点i和点j之间的边的权重用w_{ij}
- **图**：顶点和边的集合，$G=(V,E)$

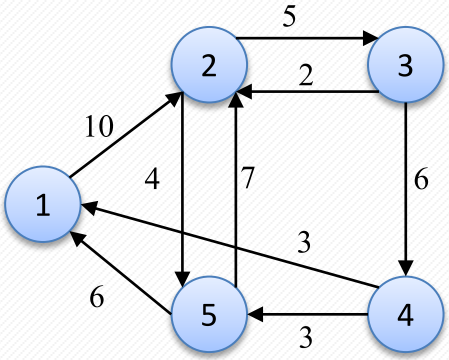
图论的基本概念

- 图中的边如果是单向箭头，表示有方向，构成的图称为**有向图**，如示例中的城市路线，是有方向的
- 如果图里的边没有箭头，只有线，表示双向，构成的图称为**无向图**，比如通信光缆之类
- 权重不仅可以表示长度之类的，也可以表示时间、费用等概念

➢ 图的表达形式

- 邻接矩阵
  - ✓ 邻接矩阵是表示**顶点间关系的矩阵**
  - ✓ 邻接矩阵W中第i行第j列的元素w_{ij}：

$$w_{ij}=\begin{cases}\text{权值，} v_i\text{和}v_j\text{之间有边时}\\ 0\text{，对角线上的元素}\\ \infty\text{，} v_i\text{和}v_j\text{之间无边时}\end{cases}$$



$$\Rightarrow\quad\begin{bmatrix}0 & 10 & \infty & \infty & \infty\\ \infty & 0 & 5 & \infty & 4\\ \infty & 2 & 0 & 6 & \infty\\ 3 & \infty & \infty & 0 & 3\\ 6 & 7 & \infty & \infty & 0\end{bmatrix}$$

题目 → 图 → 邻接矩阵 → 参与运算

Matlab作无向图

```
% （1）无权重（每条边的权重默认为1）
% 函数graph(s,t)：可在 s 和 t 中的对应节点之间创建边，并生成一个图
% s 和 t 都必须具有相同的元素数；这些节点必须都是从1开始的正整数，或都是字符串元胞数组。
% 注意哦，编号最好是从1开始连续编号，不要自己随便定义编号
s1 = [1,2,3,4];
t1 = [2,3,1,1];
G1 = graph(s1, t1);
plot(G1)
% 下面的命令是在画图后不显示坐标
set( gca, 'XTick', [], 'YTick', [] );

% 注意字符串元胞数组是用大括号包起来的哦
s2 = {'学校','电影院','网吧','酒店'};
t2 = {'电影院','酒店','酒店','KTV'};
G2 = graph(s2, t2);
plot(G2, 'linewidth', 2) % 设置线的宽度
% 下面的命令是在画图后不显示坐标
set( gca, 'XTick', [], 'YTick', [] );
```

Matlab作无向图

```
% （2）有权重
% 函数graph(s,t,w): 可在 s 和 t 中的对应节点之间以w的权重创建边，并生成一个图
s = [1,2,3,4];
t = [2,3,1,1];
w = [3,8,9,2];
G = graph(s, t, w);
% G.Edges.Weight 表示图形对象 G 中所有边的权重值，'EdgeLabel' 表示在图形上显示这些权重值
plot(G, 'EdgeLabel', G.Edges.Weight, 'linewidth', 2)
set( gca, 'XTick', [], 'YTick', [] );

% 函数graph(a): a为邻接矩阵，0表示没有边
a=[0 3 9 2;
3 0 8 0;
9 8 0 0;
2 0 0 0];
G = graph(a);
% a=[0 3 9 2;
% 0 0 8 0;
% 0 0 0 0;
% 0 0 0 0];
% G = graph(a,'upper'); %'upper'仅使用上三角矩阵来构造图
% G.Edges.Weight 表示图形对象 G 中所有边的权重值，'EdgeLabel' 表示在图形上显示这些权重值
plot(G, 'EdgeLabel', G.Edges.Weight, 'linewidth', 2)
set( gca, 'XTick', [], 'YTick', [] );
```

MATH 数模加油站

Matlab作有向图

```
%% Matlab作有向图
% 无权图 digraph(s,t)
s = [1,2,3,4,1];
t = [2,3,1,1,4];
G = digraph(s, t);
plot(G)
set( gca, 'XTick', [], 'YTick', [] );

% 有权图 digraph(s,t,w)
s = [1,2,3,4];
t = [2,3,1,1];
w = [3,8,9,2];
G = digraph(s, t, w);
plot(G, 'EdgeLabel', G.Edges.Weight, 'linewidth', 2)
set( gca, 'XTick', [], 'YTick', [] );
%digraph(a) a为邻接矩阵
a=[0 3 0 0;
0 0 8 0;
9 0 0 0;
2 0 0 0];
G = digraph(a);
% s=cellstr(strcat('顶点',int2str([1:4]')));
% % 给每个顶点标注名称, int2str([1:4]')将整数1到4六个数转化为字符
% % strcat把字符串水平串联起来
% % strcat('顶点',int2str([1:4]'))把顶点和字符1到4拼起来
% G = digraph(a,s);
plot(G, 'EdgeLabel', G.Edges.Weight, 'linewidth', 2)
set( gca, 'XTick', [], 'YTick', [] );
```

欢迎关注数模加油站

THANKS