|  |
| --- |
|  |

|  |
| --- |
| ACM/ICPC at Wuhan University |
| Xioumu STL(文字) |
| Created by xioumu, for OpenShield |

目录

[Graph 2](#_Toc337890409)

[混合图欧拉路径 2](#_Toc337890410)

[生成树的计数（matrix-Tree） 2](#_Toc337890411)

[关于中国邮递员问题和欧拉图应用 3](#_Toc337890412)

[DataStructure 5](#_Toc337890413)

[Computational Geometry 5](#_Toc337890414)

[数论 5](#_Toc337890415)

[欧拉公式 5](#_Toc337890416)

[欧拉函数 5](#_Toc337890417)

[Pick公式 5](#_Toc337890418)

[卡特兰数 6](#_Toc337890419)

[二维旋转 6](#_Toc337890420)

[逆元 6](#_Toc337890421)

[求和公式 7](#_Toc337890422)

[DP 7](#_Toc337890423)

[概率DP 7](#_Toc337890424)

# Graph

### 混合图欧拉路径

**当一个有向图所有的度为0时，这个有向图为欧拉图。**

**所以混合图的判断欧拉图只要判断原图的无向边往哪个方向即可，所以可以先以任意形态连边，然后有流量的为正向边，无流量的为反向边。**

**入度 - 出度 > 0 的，和源点连一条(入度 - 出度) / 2 的边， < 0的， 和汇点连边。**

**最后如果最大流等于所有与源点连的流量，即存在一种方法使原图变为欧拉图。**

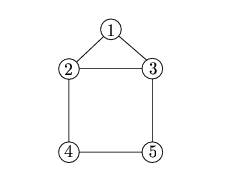
### 生成树的计数（matrix-Tree）

***Matrix-Tree*定理是解决生成树计数问题最有力的武器之一。它首先于1847年被Kirchhoff证明。在介绍定理之前，我们首先明确几个概念：**

1. ***G*的度数矩阵*D*[*G*]是一个*n*\**n*的矩阵，并且满足：当*i*≠*j*时,*dij*=0；当*i*=*j*时，*dij*等于*vi*的度数。**
2. ***G*的邻接矩阵*A*[*G*]也是一个*n*\**n*的矩阵， 并且满足：如果*vi*、*vj*之间有边直接相连，则*aij*=1，否则为0。**

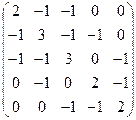
**我们定义*G*的Kirchhoff矩阵(也称为拉普拉斯算子)*C*[*G*]为C[*G*]=*D*[*G*]-*A*[*G*]，则*Matrix-Tree*定理可以描述为：*G*的所有不同的生成树的个数等于其Kirchhoff矩阵*C*[*G*]任何一个*n*-1阶主子式的行列式的绝对值。所谓*n*-1阶主子式，就是对于*r*(1≤*r*≤*n*)，将*C*[*G*]的第*r*行、第*r*列同时去掉后得到的新矩阵，用*Cr*[*G*]表示。**

**下面我们举一个例子来解释*Matrix-Tree*定理。如图*a*所示，*G*是一个由5个点组成的无向图。**

****

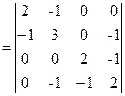
**图*a***

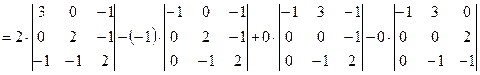
**根据定义，它的Kirchhoff矩阵*C*[*G*]为**

****

**我们取*r*=2，则有：**

**clip_image003**

****

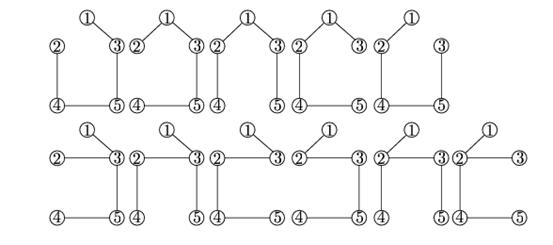
****

**clip_image006**

**clip_image007**

**clip_image008**

**这11棵生成树如图*b*所示。**

****

**图*b***

### 关于中国邮递员问题和欧拉图应用

**中国邮递员问题：**

**1962年有管梅谷先生提出中国邮递员问题（简称CPP）。一个邮递员从邮局出发，要走完他所管辖的每一条街道，可重复走一条街道，然后返回邮局。任何选择一条尽可能短的路线。**

**这个问题可以转化为：给定一个具有非负权的赋权图G，**

**（1）用添加重复边的方法求G的一个Euler赋权母图G\*，使得尽可能小。**

**（2）求G\*的Euler 环游。**

**人们也开始关注另一类似问题，旅行商问题（简称TSP）。TSP是点路优化问题，它是NPC的。而CPP是弧路优化问题，该问题有几种变形，与加权图奇点的最小完全匹配或网络流等价，有多项式算法。**[**[1]**](http://writeblog.csdn.net/Editor/FCKeditor/editor/fckeditor.html?InstanceName=ctl00_ContentPlaceHolder1_EntryEditor1_richTextEditor_richTextEditor&Toolbar=Default#_ftn1)

**欧拉图：**

**图G中经过每条边一次并且仅一次的回路称作欧拉回路。存在欧拉回路的图称为欧拉图。**

**无向图欧拉图判定：**

**无向图G为欧拉图，当且仅当G为连通图且所有顶点的度为偶数。**

**有向图欧拉图判定：**

**有向图G为欧拉图，当且仅当G的基图**[**[2]**](http://writeblog.csdn.net/Editor/FCKeditor/editor/fckeditor.html?InstanceName=ctl00_ContentPlaceHolder1_EntryEditor1_richTextEditor_richTextEditor&Toolbar=Default#_ftn2)**连通，且所有顶点的入度等于出度。**

**欧拉回路性质：**

**性质1设C是欧拉图G中的一个简单回路，将C中的边从图G中删去得到一个新的图G’，则G’的每一个极大连通子图都有一条欧拉回路。**

**性质2设C1、C2是图G的两个没有公共边，但有至少一个公共顶点的简单回路，我们可以将它们合并成一个新的简单回路C’。**

**欧拉回路算法：**

**1             在图G中任意找一个回路C；**

**2             将图G中属于回路C的边删除；**

**3             在残留图的各极大连通子图中分别寻找欧拉回路；**

**4             将各极大连通子图的欧拉回路合并到C中得到图G的欧拉回路。**

**由于该算法执行过程中每条边最多访问两次，因此该算法的时间复杂度为O(|E|)。**

**如果使用递归形式，得注意|E|的问题。使用非递归形式防止栈溢出。**

**如果图 是有向图，我们仍然可以使用以上算法。**

[**http://acm.hdu.edu.cn/showproblem.php?pid=1116**](http://acm.hdu.edu.cn/showproblem.php?pid=1116)**有向图欧拉图和半欧拉图判定**

[**http://acm.pku.edu.cn/JudgeOnline/problem?id=2337**](http://acm.pku.edu.cn/JudgeOnline/problem?id=2337)**输出路径**

**中国邮递员问题①：**

**一个邮递员从邮局出发，要走完他所管辖的每一条街道，可重复走一条街道，然后返回邮局。所有街道都是双向通行的，且每条街道都有一个长度值。任何选择一条尽可能短的路线。**

**分析：**

**双向连通，即给定无向图G。**

**如果G不连通，则无解。**

**如果G是欧拉图，则显然欧拉回路就是最优路线。**

**如果G连通，但不是欧拉图，说明图中有奇点**[**[3]**](http://writeblog.csdn.net/Editor/FCKeditor/editor/fckeditor.html?InstanceName=ctl00_ContentPlaceHolder1_EntryEditor1_richTextEditor_richTextEditor&Toolbar=Default#_ftn3)**。奇点都是成对出现的，证明从略。**

**对于最简单情况，即2个奇点，设（u，v）。我们可以在G中对（u，v）求最短路径R，构造出新图G’ = G ∪ R。此时G’就是欧拉图。**

**证明：u和v加上了一条边，度加一，改变了奇偶性。而R中其他点度加二，奇偶性不变。**

**由此可知，加一次R，能够减少两个奇点。推广到k个奇点的情况，加k/2个R就能使度全为偶数。**

**接下的问题是求一个k个奇点的配对方案，使得k/2个路径总长度最小。**

**这个就是无向完全图最小权匹配问题。有一种Edmonds算法，时间复杂度O（N^3）。**[**[4]**](http://writeblog.csdn.net/Editor/FCKeditor/editor/fckeditor.html?InstanceName=ctl00_ContentPlaceHolder1_EntryEditor1_richTextEditor_richTextEditor&Toolbar=Default#_ftn4)

**也可转换为二分图，用松弛优化的KM算法，时间复杂度也是O（N^3）。**

**完整的算法流程如下：**

**1         如果G是连通图，转2，否则返回无解并结束；**

**2         检查G中的奇点，构成图H的顶点集；**

**3         求出G中每对奇点之间的最短路径长度，作为图H对应顶点间的边权；**

**4         对H进行最小权匹配；**

**5         把最小权匹配里的每一条匹配边代表的路径，加入到图G中得到图G’；**

**6         在G’中求欧拉回路，即所求的最优路线。**

**中国邮递员问题②：**

**和①相似，只是所有街道都是单向通行的。**

**分析：**

**单向连通，即给定有向图G。**

**和①的分析一样，我们来讨论如何从G转换为欧拉图G’。**

**首先计算每个顶点v的入度与出度之差 d’（v）。如果G中所有的v都有d’（v）=0，那么G中已经存在欧拉回路。**

**d’（v）>0 说明得加上出度。d’（v）<0说明得加上入度。**

**而当d’（v）=0，则不能做任何新增路径的端点。**

**可以看出这个模型很像网络流模型。**

**顶点d’（v）>0对应于网络流模型中的源点，它发出d’（v）个单位的流；顶点d’（v）<0对应于网络流模型中的汇点，它接收-d’（v）个单位的流；而d’（v）=0的顶点，则对应于网络流模型中的中间结点，它接收的流量等于发出的流量。在原问题中还要求增加的路径总长度最小，我们可以给网络中每条边的费用值 设为图 中对应边的长度。这样，在网络中求最小费用最大流，即可使总费用最小。**

**这样构造网络N：**

**1         其顶点集为图G的所有顶点，以及附加的超级源 和超级汇 ；**

**2         对于图G中每一条边(u,v)，在N中连边(u,v)，容量为∞，费用为该边的长度；**

**3         从源点 向所有d’(v)>0的顶点v连边(s,v)，容量为d’(v)，费用为0；**

**4         从所有d’(v)<0的顶点 向汇点t连边(u,t)，容量为-d’(v)，费用为0。**

**完整的算法流程如下：**

**1         如果G的基图连通且所有顶点的入、出度均不为0，转2，否则返回无解并结束；**

**2         计算所有顶点v的d’(v)值；**

**3         构造网络N；**

**4         在网络N中求最小费用最大流；**

**5         对N中每一条流量f(u,v)的边(u,v)，在图G中增加f(u,v)次得到G’；**

**6         在G’中求欧拉回路，即为所求的最优路线。**

**NPC问题：**

**如果部分街道能够双向通行，部分街道只能单向通行。这个问题已被证明是NPC的。**[**[5]**](http://writeblog.csdn.net/Editor/FCKeditor/editor/fckeditor.html?InstanceName=ctl00_ContentPlaceHolder1_EntryEditor1_richTextEditor_richTextEditor&Toolbar=Default#_ftn5)

# DataStructure

# Computational Geometry

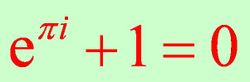
# 数论

### 欧拉公式

**（欧拉公式）如果一个连通的平面图有*n*个点，*m*条边和*f*个面，那么*f*=*m*-*n*+2**

### 欧拉函数

**e^ix=cosx+isinx**

**[](http://baike.baidu.com/albums/398/398/0/0.html#0$a6c7d717a2b9ad2dc93d6d89)**

**自然对数的底e，圆周率π，两个单位：虚数单位i和自然数的单位1，以及被称为人类伟大发现之一的0。**[**数学家**](http://baike.baidu.com/view/66878.htm)**们评价它是“上帝创造的公式”**

**计算机生成了可选文字: 御；",
沪（,,7)
1,;2，杯“，川
－……，
耳‘足耐l{J所有互异索因数，那么
k
,,7fl(I-
1
)
i=l**

**欧拉φ函数：φ(n)是所有小于n的正整数里，和n互素的整数的个数。n是一个正整数。**

|  |  |
| --- | --- |
| **欧拉证明了下面这个式子：** |  |

**如果n的标准素因子分解式是p1^a1\*p2^a2\*……\*pm^am，其中众pj(j=1,2,……,m)都是素数，而且两两不等。则有**

**φ(n)=n(1-1/p1)(1-1/p2)……(1-1/pm)**

**利用容斥原理可以证明它。**

### Pick公式

**定理一：Pick 定理，1899年**

**设 Γ 为平面上以格子点为顶点之单纯多边形，则其面积为**

**\begin{displaymath}
A=\frac{b}{2}+i-1\eqno{(8)}
\end{displaymath}**

**其中 *b* 为边界上的格子点数，*i* 为内部的格子点数。(8)式叫做 Pick 公式。**

**定理二：（棱线定理，Edge Theorem）**

**对于任意连通的三角形化的平面图枝，其棱线的个数恒为**

**\begin{displaymath}E=2b+3i-3\eqno{(13)}\end{displaymath}**

**其中 *b* 与 *i* 分别表示图枝边界上的顶点数与内部的顶点数。**

### 卡特兰数

**卡特兰数 ： 1, 1, 2, 5, 14, 42, 132, 429, 1430, 4862, 16796, 58786, 208012, 742900, 2674440, 9694845, 35357670, 129644790, 477638700, 1767263190, 6564120420, 24466267020, 91482563640, 343059613650, 1289904147324, 4861946401452,**

**令h(1)=1,h(0)=1，catalan数满足递归式：**

**h(n)= h(0)\*h(n-1)+h(1)\*h(n-2) + ... + h(n-1)h(0) (其中n>=2)**

**另类递归式：**

**h(n)=h(n-1)\*(4\*n-2)/(n+1);**

**该递推关系的解为：**

**h(n)=C(2n,n)/(n+1) (n=1,2,3,...)**

### 二维旋转

**看下面的矩阵**

**[clip_image002[4]](http://gaoyunxiang.com/wp-content/uploads/2010/02/clip_image0024.gif)**

**这个是二维向量的旋转矩阵，它可以将一个向量逆时针旋转一个角度。**

**将其变形，变会得到二维向量的顺时针旋转的形式：**

**[clip_image002[16]](http://gaoyunxiang.com/wp-content/uploads/2010/02/clip_image00216.gif)**

**这里要注意一种特殊的情况，就是当角度为90度的时候，sin和cos的结果只有1, 0, -1三种可能性。所以这个矩阵可以改写成特殊的形式，其意义在于用这样的旋转操作，不会产生精度问题。sin和cos的运算的精度是比较低的，能少用则尽量少用。在计算几何中的向量旋转操作，大部分都可以通过变形，只用到旋转90度，从而避免精度问题。**

**===3维====**

**围绕轴u = (ux, uy, uz)来旋转的矩阵代码如下：**

1. **double a[3][3] = {**
2. **{SQR(ux) + (1 – SQR(ux)) \* c, ux \* uy \* (1 – c) – uz \* s, ux \* uz \* (1 – c) + uy \* s},**
3. **{ux \* uy \* (1 – c) + uz \* s, SQR(uy) + (1 – SQR(uy)) \* c, uy \* uz \* (1 – c) – ux \* s},**
4. **{ux \* uz \* (1 – c) – uy \* s, uy \* uz \* (1 – c) + ux \* s, SQR(uz) + (1 – SQR(uz)) \* c}**
5. **};**

**要注意这个ux \* ux + uy \* uy + uz \* uz = 1**

**有一个比较有意思的问题就是，知道旋转矩阵之后，怎么来确定向量u呢？**

**我们做如下变形之后，可以得到：**

**Ru = Iu –> (R – I)u = 0**

**也就是说我们要找到一个非0的向量，使得他和一个矩阵乘起来得到一个空矩阵。这个问题可以用高斯消元来解决。实际上我们就是要解一个线性方程组。**

### 逆元

**定义 如果*ab*≡1(mod *m*), 则称*b*是*a*的模*m*逆,记作*a*的模*m*逆是方程*ax*≡1(mod *m*)的解.**

**即gcd(a, m) == 1, 即(a, m)互质**

**例：求5的模7逆**

**做辗转相除法, 求得整数*b,k*使得 5*b*+7*k*=1, 则*b*是5的模7逆.**

**计算如下:**

**7=5+2,  5=2×2+1.**

**回代    1=5-2×2=5-2×(7-5)= 3×5-2×7,**

**得 5 -1≡3(mod7).**

**例：求21的模73逆**

**做辗转相除法, 求得整数*b,k*使得 21*b*+73*k*=1, 则*b*是21的模73逆.**

**计算如下:**

**73=21\*3+10**

**21=10\*2+1**

**回代    1=21-10\*2**

**1=21-（73-21\*3）\*2**

**=21-73\*2+6\*21**

**=7\*21-73\*2**

**得 21 -1≡7(mod73).**

### 求和公式

1^4+2^4+3^4+4^4+……+n^4 = n(n+1)(2n+1)(3n^2+3n-1)/30

1^3+2^3+3^3+4^3+……+n^3=[n(n+1)/2]^2

1^2+2^2+3^2+4^4+……+n^2=n(n+1)(2n+1)/6

1+2+3+4+……+n=n(n+1)/2

# DP

### 概率DP

**要倒着DP，有环可用高斯消元，或者设X直接把X给化简到一边。**