

# 一笔画问题

## 一、问题概述

对于一个给定的连通图，怎样判断是否存在一个恰好包含了所有的边，并且没有重复的路径？这就是一笔画问题。用图论的术语来说，就是判断这个图是否是一个能够遍历完所有的边而没有重复。这样的图现称为欧拉图。这时遍历的路径称作欧拉路径（一个圈或者一条链），如果路径闭合（一个圈），则称为欧拉回路。

## 二、求解算法思想

先判断图是否存在欧拉路径。首先可用并查集判断图是否连通。如果图是无向图，则图的度为奇数的顶点的个数是否 0 或 2。如果是有向图，一个连通的有向图可以表示为一条从顶点  $u$  到  $v$  的（不闭合的）欧拉路径的充要条件是： $u$  的出度（从这个顶点发出的有向边的数量）比入度（指向这个顶点的有向边的数量）多 1， $v$  的出度比入度少 1，而其它顶点的出度和入度都相等。

## 三、举例说明

一笔画游戏是一个数学游戏 即平面上由多条线段构成的一个图形能不能一笔画成，使得在每条线段上都不重复？例如汉字‘日’和‘中’字都可以一笔画的，而‘田’和‘目’则不能。

请编程实现一笔画：

首先输入坐标系上的点数个数，然后输入点的坐标，请判断这张图是否可以一笔画出，并输出画线顺序（每条线段必须经过一次，且只能经过一次。每个端点可以经过多次。）当有多种方式可以完成一笔画时，每一步都必须尽可能先画数值最小的端点。

输入：顶点的个数，顶点的坐标

输出：如果可以一笔画则输出画线顺序

样例输入：

```
3
1 2
1 3
2 3
```

样例输出：

true

#### 四、具体步骤

- 1、建立邻接矩阵,  $\text{link}[i,j] = 1$  或  $0$ ;
- 2、求每个顶点的度数, 存入  $\text{degree}[i]$ ;
- 3、统计奇点的个数, 存入  $\text{odt}$ ;
- 4、若无奇点, 则可从任意一结点出发开始一笔画, 一般  $\text{start}=1$ ;
- 5、若有 2 个奇点, 则从其中一个奇点  $i$  出发开始一笔画,  $\text{start}=i$ ;
- 6、若奇点个数超过 2 个, 则不能一笔画出。 如何画呢? 即如何打印序列呢? 细化第 4、5 步:

- 4.1 设  $\text{sum}$  为图的总度数;
- 4.2 从一个奇点 (偶点) 出发, 打印出来, 且  $\text{degree}[i]-1$ ;
- 4.3 从邻接矩阵中找到结点  $i$  的下一个邻接点  $j$ , 且  $\text{degree}[j]-1$ ;
- 4.4  $\text{sum}:=\text{sum}-2$ ;
- 4.5 判断“ $\text{sum}=0$ ”, 成立就结束, 否则转 4.2。

#### 五、性能分析

用邻接矩阵存储图,  $O(n^2)$ .

计算每个顶点的度数, 每条边有两个顶点, 所以  $O(2|E|)$ .

图的一笔画即每条边都会遍历一遍且不重复, 所以  $O(|E|)$ .

## 哈密尔顿问题

#### 一、问题描述

回溯搜索解哈密尔顿问题: 哈密尔顿回路就是指经过图 (有向图或无向图) 中所有顶点一次且仅一次的通路。

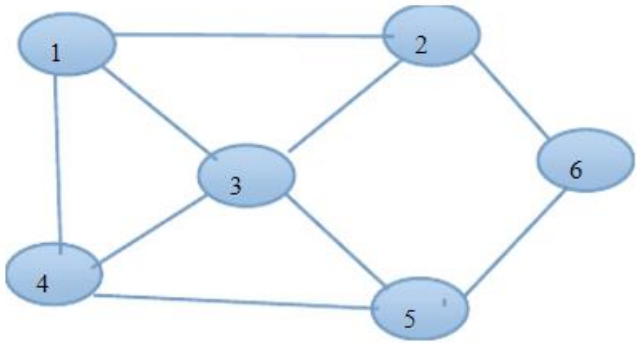
#### 二、求解算法思想

用回溯法求解哈密尔顿回路问题, 首先要画出问题的解空间树, 该解空间树是一棵最大度是  $n$  的树 (其中  $n$  为图中的顶点数), 树中只有第一个结点的度为  $n$ , 其余的结点的度都为  $n-1$  (该结点不用与其自身相连)。在编写算法时, 可以通过判断该边在图的邻接矩阵中

的值来剪枝，如果其值不是 1，则说明该边不存在，则剪枝不用继续搜索。由于在求图的哈密尔顿回路时，走过的顶点不能再重复走，所以要对已经遍历过的顶点做一个标记，如果在搜索时找到的是一个带有标记的顶点，那么该路径也是不可行的，应剪去。

### 三、举例说明

创建一个有 6 个顶点，9 条边的连通图，如下图所示，求解哈密尔顿回路的所有解。



此实例共有 6 种解，分别为：

回路 1: 1 2 6 5 3 4

回路 2: 1 2 6 5 4 3

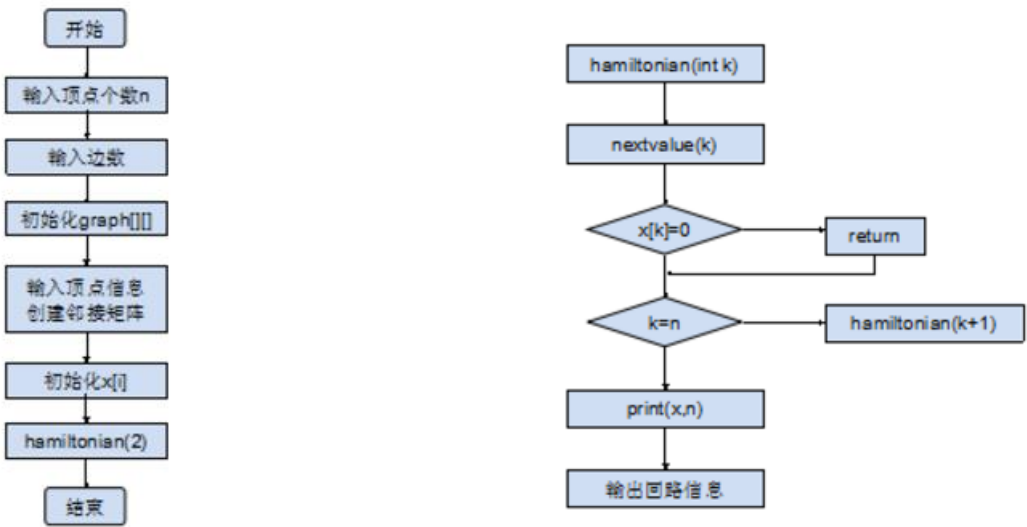
回路 3: 1 3 2 6 5 4

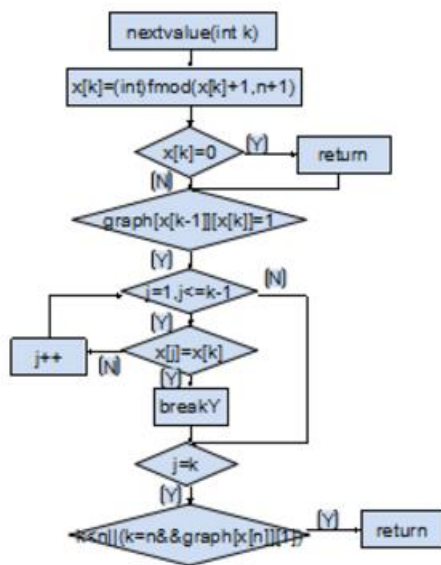
回路 4: 1 3 4 5 6 2

回路 5: 1 4 3 5 6 2

回路 6: 1 4 5 6 2 3

### 四、算法步骤





## 五、性能分析

回溯算法采用深度优先遍历的算法对解空间树进行遍历，所以 $O(|V| + |E|)$

# 中国邮递员问题

## 一、问题描述

中国邮递员问题是我国数学家管梅谷先生在 20 世纪 60 年代提出来的。该问题描述如下：一个邮递员从邮局出发，到所管辖的街道投递邮件，最后返回邮局，若必须走遍所辖各街道中每一条至少一次，则怎样选择投递路线使所走的路程最短？

下面用图论的语言来描述：在一个带权图  $G$  中，能否找到一条回路  $C$ ，使  $C$  包含  $G$  的每条边至少一次且  $C$  的长度最短？该问题求解思路大体包括三个方面 5： (1)若  $G$  没有奇度数结点，则  $G$  是 Euler 图，于是 Euler 回路  $C$  是惟一的最小长度的投递路线。 (2)若  $G$  恰有两个奇度数结点  $iv$  和  $jv$ ，则  $G$  具有 Euler 迹，且邮局位于结点  $iv$ ，则邮递员走遍所有的街道一次到达结点  $jv$ ；从  $jv$  返回  $iv$  可选择其间的一条 最短路径。这样，最短邮问题转化为求  $iv$  到  $jv$  的 Euler 迹和  $jv$  到  $iv$  的最短路径问题。 (3)若  $G$  中奇度数结点的数目多于 2，则回路中必须增加更多的重复边(路径)。这时，怎样使重复边的总长度最小？前面两种情况比较容易求解，第三种情况相对要复杂一些，因为涉及到奇度数结点的配对问题。当奇度数结点数目只有两个时，它存在的配对方案只有 1 种；当奇度数结点数目有四个时，它存在的配对方案有  $3 \times 1 = 3$  种；当奇度数结点增加到六个时，它存在的配对方案增加到  $5 \times 3 \times 1 = 15$

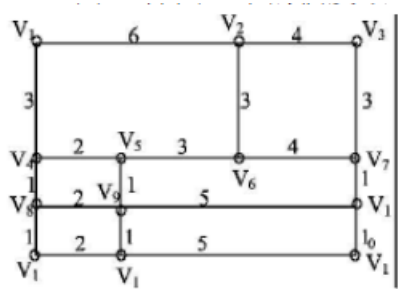
种。依此规律，有下面的结论：定理设连通图  $G$  中有  $n(n \geq 2)$  个奇度数结点，则存在奇度数结点两两配对的方案数共有  $(n-1) \times (n-3) \times (n-5) \times \dots \times 1$  种。

二、求解算法思想

求解中国邮递员问题的传统方法是通过计算奇度数结点之间的最短路径来确定结点配对，添加重复边。但是，从上面定理的结论可以看出，当图中的奇度数结点数目较多时，其计算量将非常大。因此，我们可以考虑，既然核心问题是要解决奇度数结点的配对问题并在奇度数结点之间添加重复边，那么不妨将原始图中的偶度数结点去掉，只考虑奇度数结点，并利用最小生成树的观点来快速地找出奇度数结点的最优配对方案。具体思路如下：

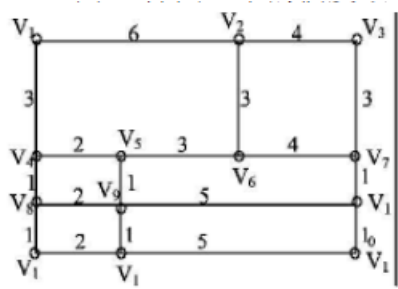
- (1)去掉原始图中的偶度数结点，即得到的新图中只包含原奇度数结点与它们之间的路径；
- (2)求新图的最小生成树；
- (3)由最小生成树确定奇度数结点的配对；
- (4)在原始图中添加重复边。

三、举例说明



四、算法步骤

- (1)如图 1 所示，该中国邮递员问题共有 8 个奇度数结点



- (2)去掉图 1 中的偶度数结点，得到新图 2。

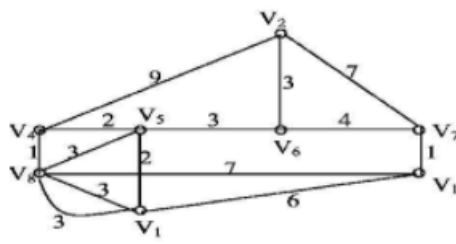


图 2

(3)求图 2 的最小生成树(如图 3 所示)。

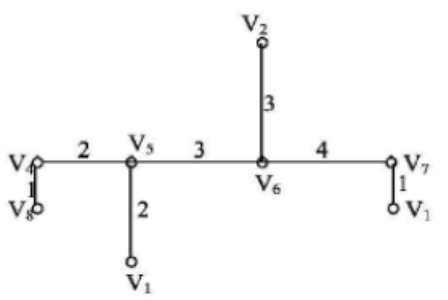


图 3

(4)由图 3 可知 2v 与 6v 配对, 4v 与 8v 配对, 5v 与 12v 配对, 7v 与 10v 配对。

(5)根据奇度数结点的配对在原始图 1 中添加重复边(如图 4 所示), 经检验此方案已是最优。

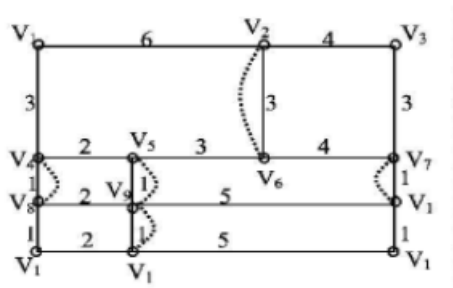


图 4

## 五、性能分析

此方案已是最优。

# 旅行推销员问题

## 一、问题描述

一个旅行推销员在返回他所在的城市之前, 他要访遍上司安排他去的所有城市。我们如

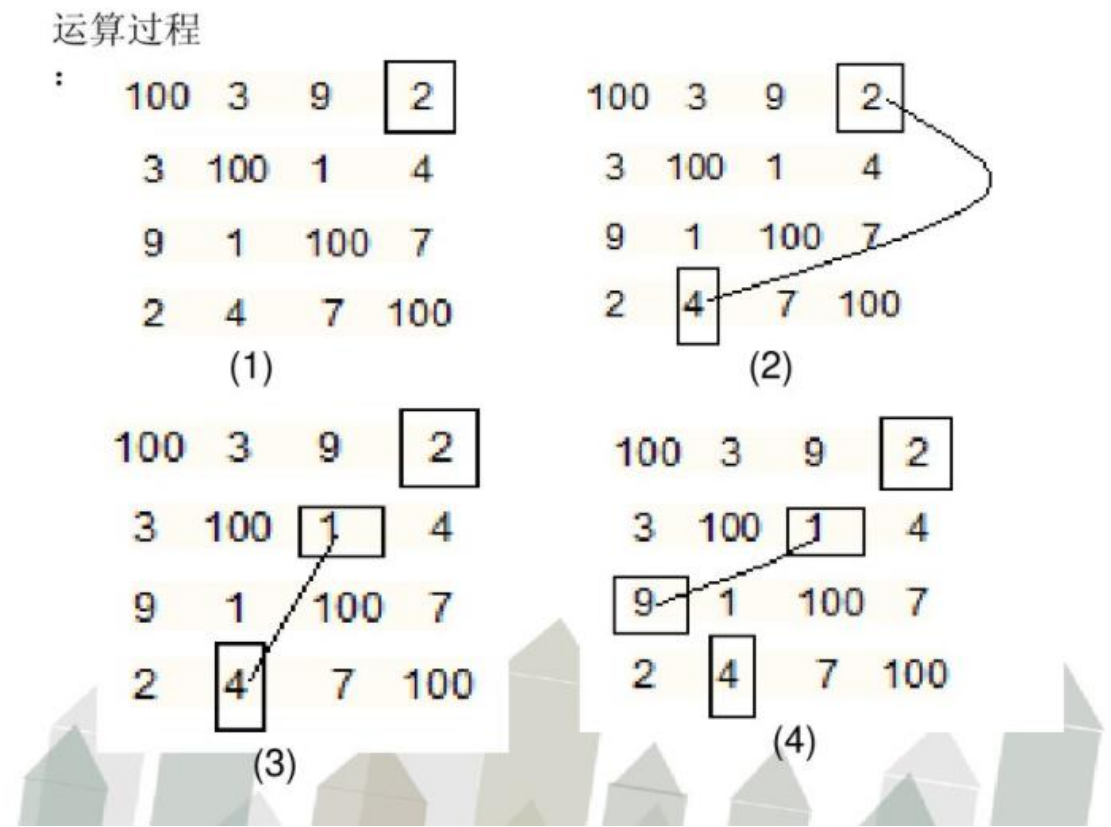
何能找到一条路线，使推销员以最小的总路程（或时间，旅费）访遍这些城市，最后再回到他所在的城市。这就是著名的“旅行推销员问题”。这个问题可以用图论语言说的更广义一些，“旅行推销员问题”就是在给定的连通加权图  $(G, W)$ 。其中  $G$  的顶点视为各个城市，城市间的航线视为边，权视为两个城市间的踞离，也可以视为时间或旅费。

二、求解算法思想

针对 TSP 问题，使用贪心算法的求解的过程为：

- 1.从某一个城市开始，每次选择一个城市，直到所有的城市被走完。
- 2.每次在选择下一个城市的时候，只考虑当前情况，保证迄今为止经过的路径总距离最小。

三、举例说明



以4个节点为例，演示算法运行过程（以100表示无大）：

输入连接矩阵：

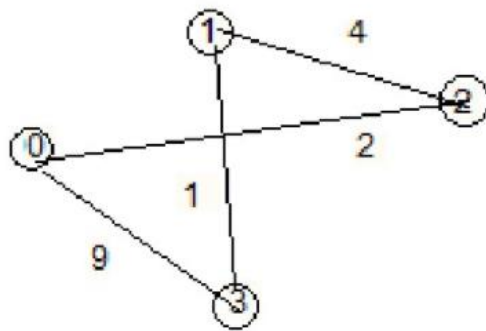
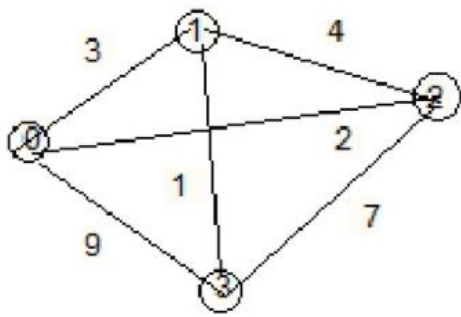
100 3 9 2

3 100 1 4

9 1 100 7

2 4 7 100

对应连线图：



#### 四、算法步骤

城市用数字编号来表示， $1, 2, \dots, N$ ，任何两个城市的距离记录在数组  $D[i, j]$  中，一次访问过的城市编号被记录在  $S[1], S[2], \dots, S[N]$  中，即第  $i$  次访问的城市记录在  $S[i]$  中。从第一个城市开始访问起，将城市编号 1 赋值给  $S[1]$ 。将第  $i$  次访问的城市  $j$ ，其距第  $i-1$  次访问城市的距离最短。

#### 五、性能分析

因为旅行商问题是一个典型的 NP 问题，找不到一个算法能保证在多项式时间内得到最优解。所以无需证明其贪心选择性质，而本算法只要求找到近似解，而在多项式时间内结束。