湖南大学
HUNAN UNIVERSITY

图的遍历问题

图的遍历是指从图中的任意一点出发，对图中所有顶点访问一次且只访问一次。图的遍历操作是其他遍历操作的基础

图的遍历分类

01. 一笔画问题（欧拉路径）

02. 哈密尔顿问题

03. 中国邮递员问题

04. 旅行推销员问题

- 问题概述
- 求解算法思想
- 求解过程的说明（实例）
- 算法具体步骤
- 性能分析

一. 一笔画问题（欧拉路径）

问题概述

什么是欧拉路径？欧拉路径就是一条能够不重不漏地经过图上的每一条边的路径，即一笔画问题。而若这条路径的起点和终点相同，即从起点出发后可以再次回到终点，则将这条路径称为欧拉回路。

无向图G中存在欧拉回路的充要条件是图中所有结点的度数均是偶数

求解的算法思想

如何判断一个图是否有欧拉路径呢？与一笔画问题相同，一个图有欧拉路径需要以下几个条件：

- 一个连通图
- 若是无向图，则这个图的度数为奇数的点的个数必须是0或2；若是有向图，则要么所有点的入度和出度相等，要么有且只有两个点的入度分别比出度大1和少1

求解的算法思想

根据欧拉路径有几个特点:

- 入度为0的点不超过1个
- 出度为0的点不超过1个
- 度为奇数点的个数等于0或者2(等于0说明可以回来,不等于0说明不能回来)

可以通过判断这些条件,从任意一个点开始深度优先搜索,如果能遍历到图中所有的点,则图为连通图,那么可以判断此图可以构成欧拉路径

求解过程的说明

描述

zyc从小就比较喜欢玩一些小游戏，其中就包括画一笔画，他想请你帮他写一个程序，判断一个图是否能够用一笔画下来。

规定，所有的边都只能画一次，不能重复画。

输入

第一行只有一个正整数N(N<=10)表示测试数据的组数。
每组测试数据的第一行有两个正整数P,Q(P<=1000,Q<=2000)，分别表示这个画中有多少个顶点和多少条连线。（点的编号从1到P）
随后的Q行，每行有两个正整数A,B(0<A,B<P)，表示编号为A和B的两点之间有连线。

输出

如果存在符合条件的连线，则输出"Yes"，
如果不存在符合条件的连线，输出"No"。

样例输入

```
2
4 3
1 2
1 3
1 4
4 5
1 2
2 3
1 3
1 4
3 4
```

样例输出

```
No
Yes
```

分析：

- 这个题目设计到处理顶点之间多对多的关系，考虑到用图来求解。这是一个图的欧拉路径的问题。
- 首先想到的就是遍历，从图中一个顶点出发，如果能满足欧拉图的特点，并且遍历完所有结点，则说明这个图能一笔画出。

求解过程的说明

我们采用邻接矩阵的方式来存储图

- 对于第一个样例，找到第一个结点1，遍历它的第一个点2，判断出度和入度，再接着遍历3,和4，发现出度为0的点存在三个，不满足条件，故这个图不能一笔画成。
- 对于第二个样例，找到第一个结点1，遍历它的第一个结点2，每次遍历一个结点都进行一次出度和入度的判断，不满足条件即返回，表明不能一笔画成。2的度为2，遍历2的下一个结点3,3的度数为2，遍历3的下一个结点2，而2已被标记，再遍历4,4的度为2，满足。遍历完所有结点后均满足条件，故该图可以一笔画成。

求解的算法步骤

算法流程：

- 对于无向图，判断度数为奇数的点的个数，若为0，则设任意一点为起点，若为2，则从这2个点中任取一个作为起点；对于有向图，判断入度和出度不同的点的个数，若为0，则设任意一点为起点，若为2，则设入度比出度小1的点为起点，另一点为终点。具体起点的选择要视题目要求而定。
- 从起点开始进行递归：对于当前节点x，扫描与x相连的所有边，当扫描到一条(x,y)时，删除该边，并递归y。扫描完所有边后，将x加入队列。如果遍历完所有点则说明这个图存在欧拉路径。

二. 哈密尔顿问题

问题概述

哈密顿问题一种遍历完图中所有点的问题。
如果图中存在一条经过所有结点一次且仅有一次的路，则称为哈密顿路，如果回到起点则称为哈密顿回路。存在哈密顿回路的图称为哈密顿图。

一些对比：
和欧拉路径进行对比，欧拉路径问题是一次不重复的遍历完所有边，而哈密顿问题则是一次不重复遍历完所有点

求解的算法思想

- 哈密尔顿问题判定的充分条件

设G是n阶无向简单图，若对于G中任意不相邻的顶点u、v，均有d(u)+d(v)>=n-1，则说明G中存在哈密顿通路。需要注意的是满足该条件一定存在哈密顿通路，但不满足该条件不一定不存在哈密顿通路。

- 判断一个图是否存在哈密顿路和哈密顿回路

每一对节点的度数和不小于n-1/每一对节点的度数和不小于n.

- 求出每个点的度数，根据度数来判断是否存在哈密顿回路。

求解过程的说明

1.遇到一个图，可以先将其装换为用邻接矩阵来存储，计算出每个点的度数

```
void calDegree(int A[][N],int n,int D[])
{
    //求每个点的度数
    int i=0,j=0;
    for (i=0;i<n;i++){D[i]=0;}
    for (i=0;i<n;i++)
    {
        for (j=0;j<n;j++){D[i]+=(A[i][j]==0?0:1);}
    }
}
```

2.其次，遍历每一个点，看两个点的度数之和是否满足是哈密顿回路的条件，关键步骤如下：

```
for (i=0;i<n;i++)
{
    for (j=0;j<n;j++)
    {
        if(i!=j)
        {
            if (D[i]+D[j]<(n-1)){isHp=0;isHg=0;break;}
            else if (D[i]+D[j]<n){isHg=0;break;}
        }
    }
    if (isHp==0 ){break;}
}
```

求解的算法步骤

1. 将输入信息存储在邻接矩阵中来表示图

2. 计算每一个点的出度和入度之和，只需求对应行或者对应列非0位置的个数即可

3. 根据哈密顿路的判定条件去判断遍历所有结点去看是否每一对结点都满足条件

三. 中国邮递员问题

问题概述

中国邮递员问题是邮递员在某一地区的信件投递路程问题。邮递员每天从邮局出发，走遍该地区所有街道再返回邮局，问题是他应如何安排送信的路线可以使所走的总路程最短。这个问题由中国学者管梅谷在1960年首先提出，并给出了解法——“奇偶点图上作业法”，被国际上统称为“中国邮递员问题”。

用图论的语言描述，给定一个连通图G，每边e有非负权，要求一条回路经过每条边至少一次，且满足总权最小。

求解的算法思想

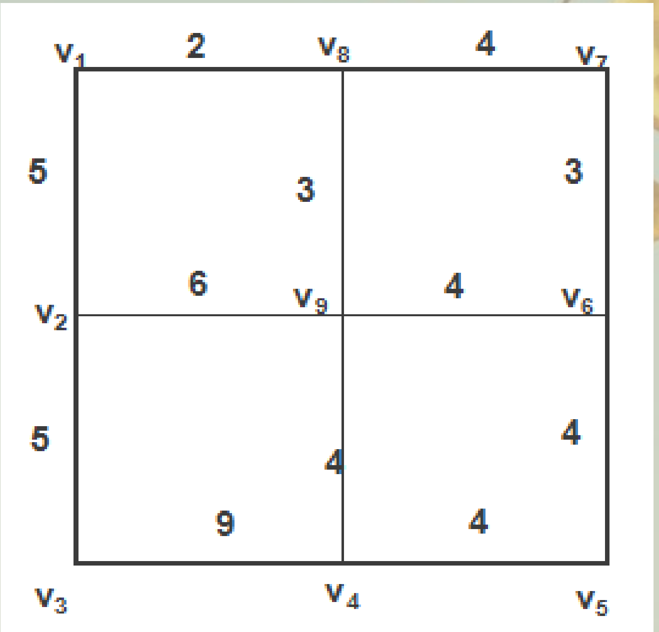
分析:

- 问题中要求我们每条边都要走过一遍并且最后还要能返回到出发点。每一节点都一定会进行后还要出来，所以每个点的出度和入度必须相等，即每个节点的度应为偶数个，即满足欧拉回路。
- 但实际上有的图并非所有点的度数都为偶数个，这时可以根据需要自己构造边去构造出一个欧拉回路。此时最短路径就是原来所有边的长度和新构造出的边数的最小长度和。

求解过程的说明

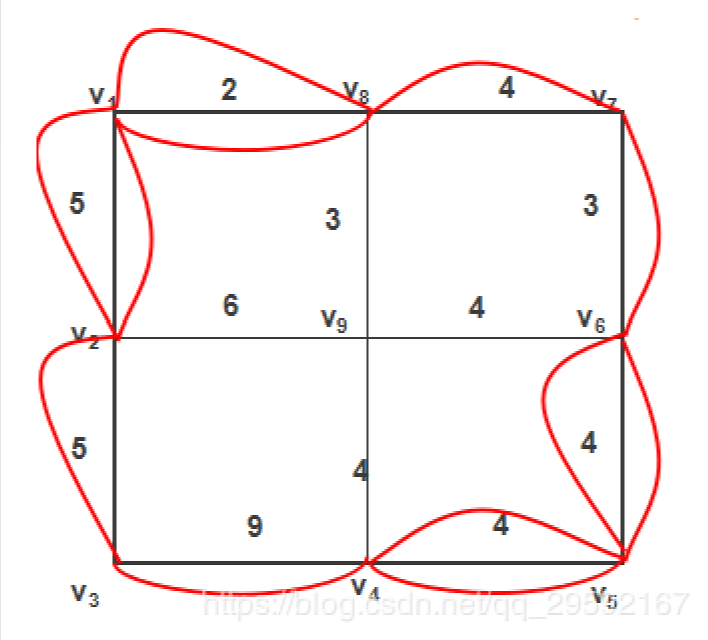
问题描述：给出一个图，这个图中的每一个节点表示村庄，每一条边表示街道，要求走完所有街道，求解走完所有街道的最小路径和

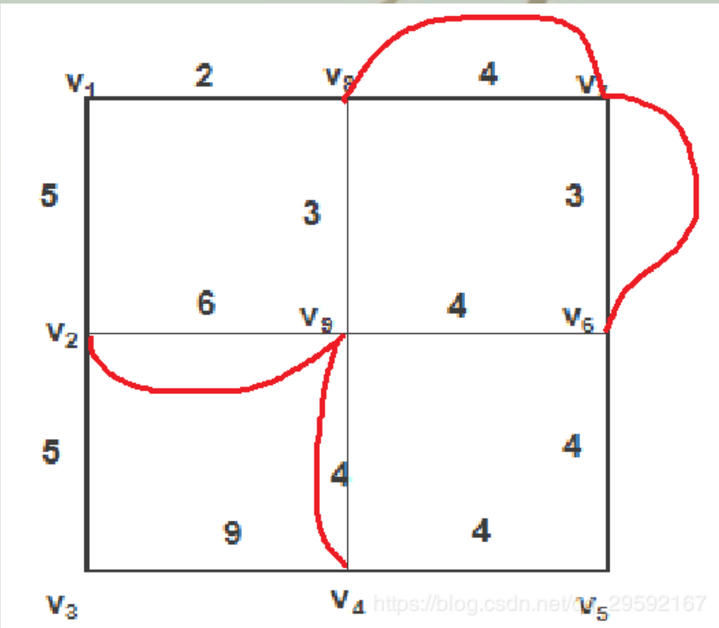
问题的图如右所示

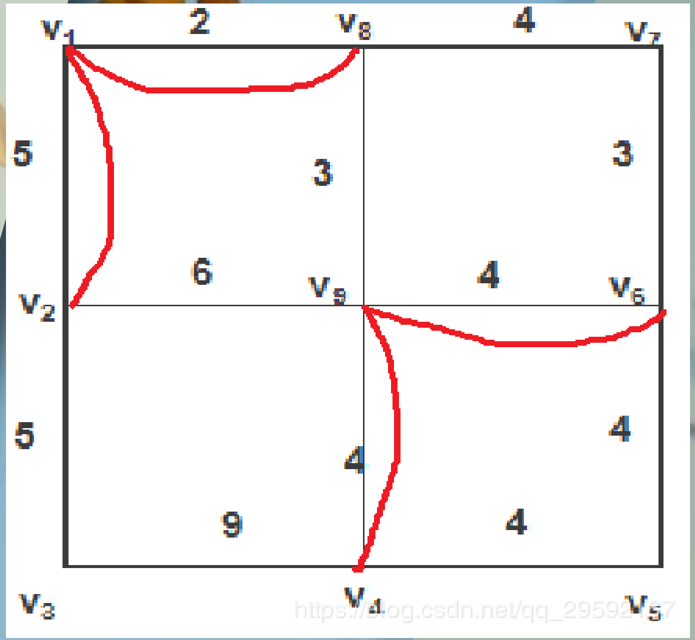


求解过程的说明

第一步根据需要去构造出一些欧拉回路





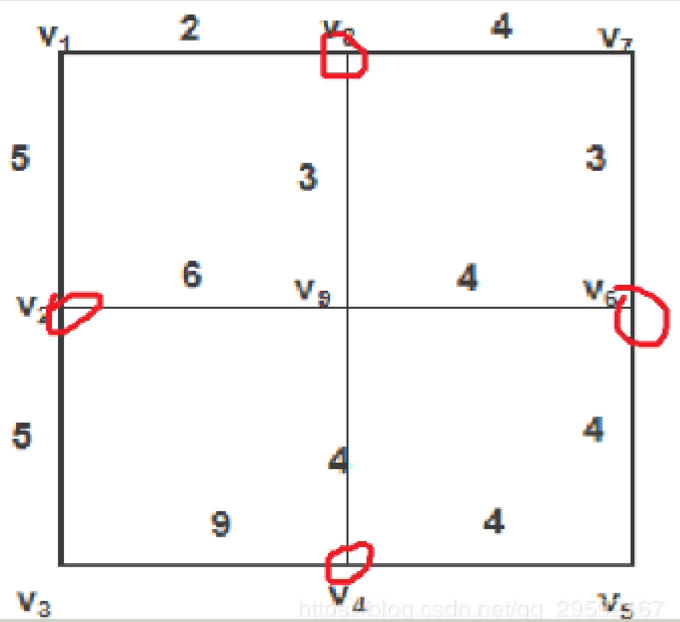


综合对比发现，构造出的所有欧拉回路，只有最后一个是最短路径的，因此问题的关键在于**构造出最优的欧拉回路**

求解过程的说明

构造最优的欧拉回路

1. 找出所有点中度数为奇数的点，并标记



2. 奇数节点一定是有偶数个，因此，最终的距离应该是：两两点之间的最短距离，而两两点之间的最短距离可以用Floyd或者Dijkstra或者bellman-ford算法来得到。这里使用使用Floyd算法，不容易出错。

求解过程的说明

构造最优的欧拉回路

3.遍历所有的组合情况，求出最短的组合方式例如比较 d(2,8)+d(4,6)，d(2,6)+d(4,8)，d(2,4)+d(6,8)，然后取最小值即为最终的优化方案：d(2,8)+d(4,6)。这里可以采用DFS来寻找最短路径的方案，也可以用DFS的思想采用动态规划来实现

4.最终的方案结果是：原图中的所有路径+构造时新添加的路径。

求解的算法步骤

算法步骤：

1、生成邻接矩阵，利用Floyd算法求出每两个点之间的最短距离。（Floyd）

2、判断整个过程是否是欧拉回路，如果不是构造欧拉回路。

3、利用动态规划或者DFS计算构造欧拉回路的最优方案。

4、计算原路径与新构造路径的长度总和即为最终结果。

四．旅行推销员问题

问题概述

- 旅行推销员问题（Travelling salesman problem，TSP）是这样一个问题：给定一系列城市和每对城市之间的距离，求解访问每一座城市一次并回到起始城市的最短回路。它是组合优化中的一个NP难问题，在运筹学和理论计算机科学中非常重要。
- 最早的旅行商问题的数学规划是由Dantzig（1959）等人提出，并且是在最优化领域中进行了深入研究。许多优化方法都用它作为一个测试基准。尽管问题在计算上很困难，但已经有了大量的启发式算法和精确方法来求解数量上万的实例，并且能将误差控制在1%内。

问题概述

- 旅行商问题是图论中最著名的问题之一，即“已给一个n个点的完全图，每条边都有一个长度，求总长度最短的经过每个顶点正好一次的封闭回路”。该问题通常被认为是一个NP完全问题。时间复杂度为O(n!)

求解的算法思想

- TSP问题要求遍历不重复所有顶点，求出最短路径即等价于求解图的最短哈密尔顿回路问题。
- 令G=(V, E)是一个带权重的有向图，顶点集V=(v0, v1, ..., vn-1)。从图中任一顶点vi出发，经图中所有其他顶点一次且只有一次，最后回到同一顶点vi的最短路径，故可以用动态规划来求解此问题

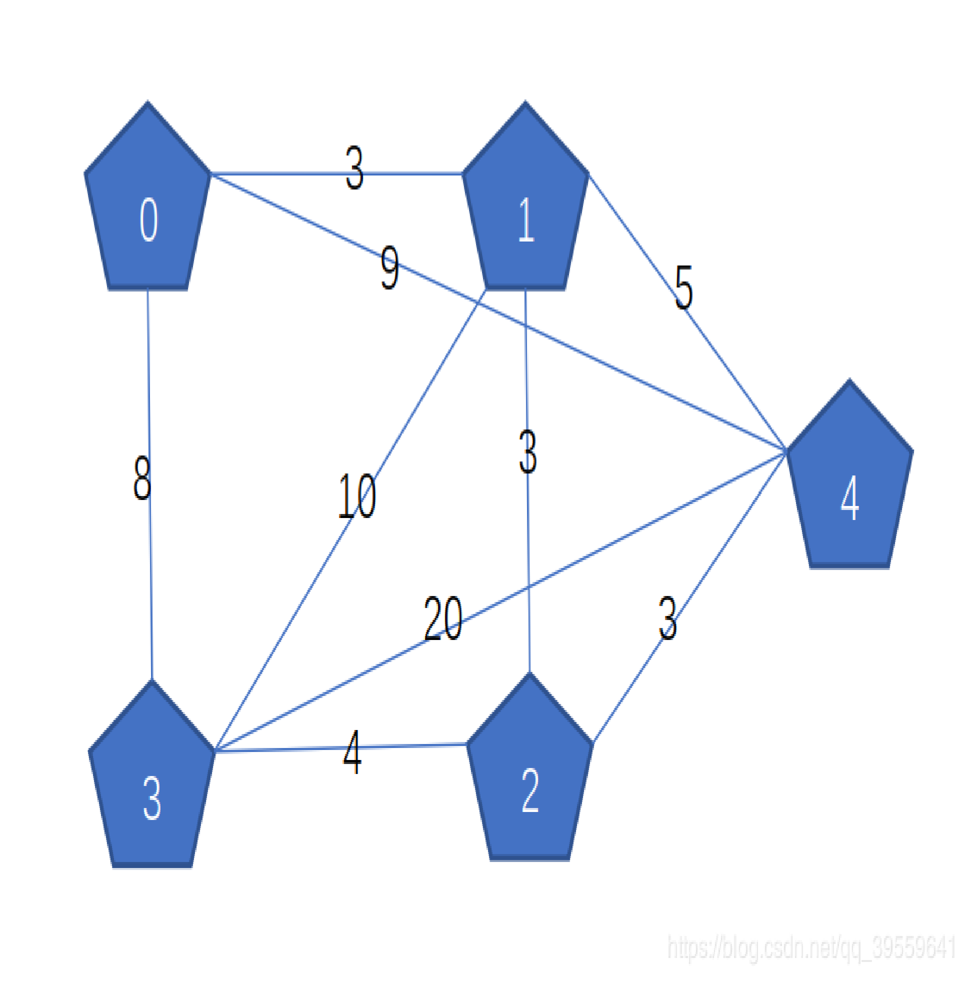
求解过程的说明

问题描述

一个售货员必须访问n个城市，恰好访问每个城市一次，并最终回到出发城市。

售货员从城市i到城市j的旅行费用是一个整数，旅行所需的全部费用是他旅行经过的的各边费用之和，而售货员希望使整个旅行费用最低。

测试用例

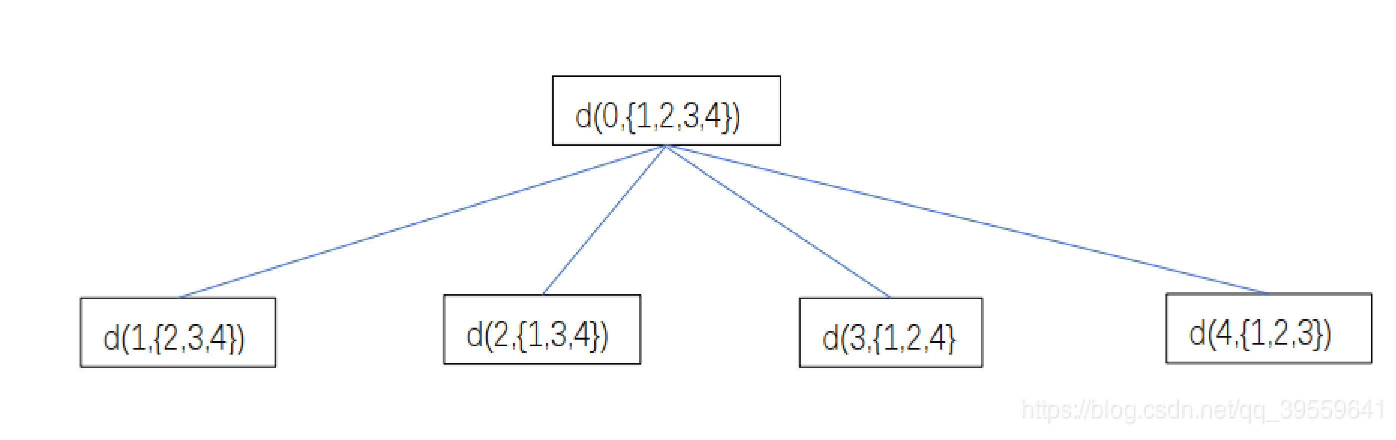


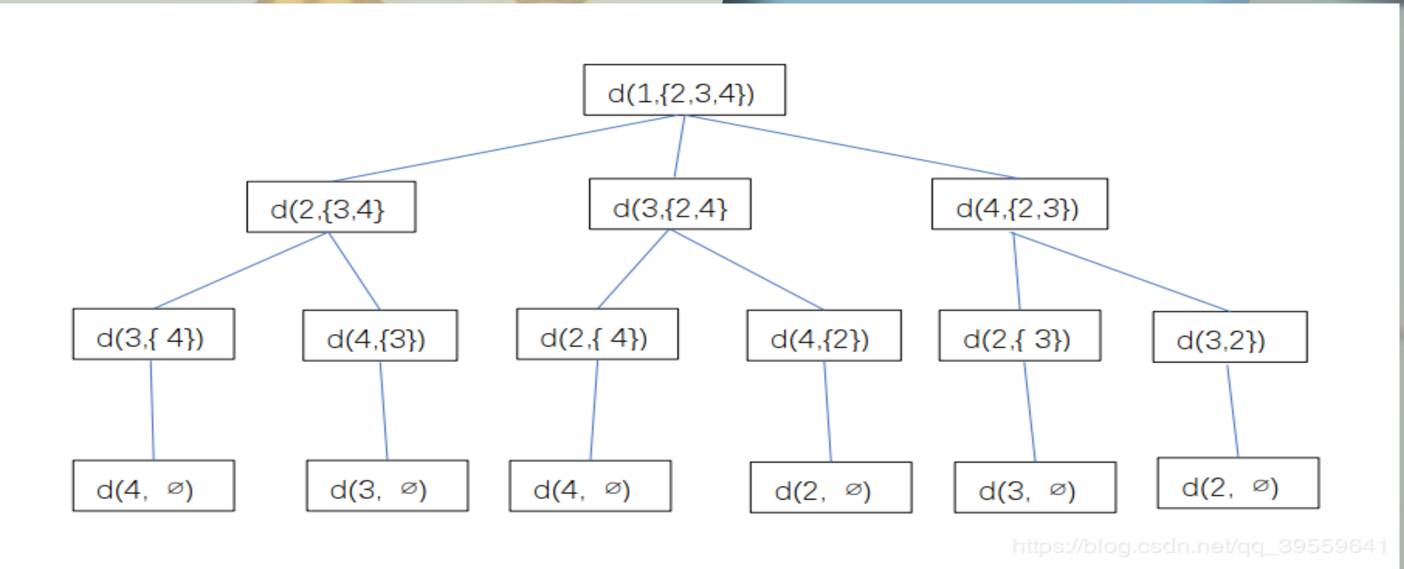
求解过程的说明

将图用邻接矩阵表示出来

$$\begin{vmatrix} \infty & 3 & \infty & 8 & 9 \\ 3 & \infty & 3 & 10 & 5 \\ \infty & 3 & \infty & 4 & 3 \\ 8 & 10 & 4 & \infty & 20 \\ 9 & 5 & 3 & 20 & \infty \end{vmatrix}$$

假设出发城市是 0城市





求解过程的说明

1.我们要求的最终结果是d(0,{1,2,3,4}),它表示，从城市0开始，经过{1,2,3,4}之中的城市并且只有一次，求出最短路径.

2.d(0,{1,2,3,4})是不能一下求出来的，看上图的第二层，第二层表明了d(0,{1,2,3,4})所需依赖的值。那么得出：

$$
d(0,\{1,2,3,4\}) = \min \left\{
\begin{array}{l}
C_{01}+d(1,\{2,3,4\} \\
C_{02}+d\{2,\{1,3,4\}\} \\
C_{03}+d\{3,\{1,2,4\}\} \\
C_{04}+d\{3,\{1,2,3\}\}
\end{array}
\right\}
$$

3. d(2,{1,3,4})，d(3,{1,2,4})，d(4,{1,2,3})同样需要这么求。按照上面的思路，只有最后一层的，当V为空集时，就可以满足d(I,V)=cis的条件，直接求出dp数组部分的值

注意，这里的二维数据dp[N][M]来表示，N表示城市的个数M表示集合的数量，即M=2^(N-1)

求解的算法步骤

算法步骤：
假设从顶点s出发，令d(i, V)表示从顶点i出发经过V(是一个点的集合)中各个顶点一次且仅一次，最后回到出发点s的最短路径长度。
①当V为空集，那么，表示直接从i回到s了，此时
d(i,V)=Cis 且 i!=s
②如果V不为空，那么就是对子问题的最优求解。必须在V这个城市集合中，尝试每一个，并求出最优解。
d(i,V)=min(Cik+d(k,V-(k));
注：Cik表示选择的城市i和城市k的距离，d(k,V-(k)是一个子问题。

动态规划的方程如右

$$d(i,V)=\begin{cases} c_{is}, & V=\emptyset, i\neq s \\ \min\{c_{ik}+d(k,V-\{K\})\}, & k\in V, V\neq\emptyset \end{cases}$$

其中 s 为起点。

四种问题求解的性能分析

- 欧拉路径问题

利用DFS遍历所有结点，每次进行是否是欧拉路径的一个判定，时间复杂度为O（n）

- 中国邮递员问题

这个问题实际上一个欧拉回路的求解，没有欧拉回路则构造一个欧拉回路，使得路径最短。求每两个点之间的最短路用Floyd算法，最后用DFS去遍历，时间复杂度为O（n^3）

四种问题求解的性能分析

- 哈密尔顿问题

验证每一对结点是否满足判断条件的时候，利用二重循环进行判断，根据复杂度计算原则，时间复杂度为O（n^2）.

- 旅行推销员问题问题

TSP问题要求遍历不重复所有顶点，求出最短路径即等价于求解图的最短哈密尔顿回路问题。采用动态规划找出经过所有点的最小路径。时间复杂度为O（2^n*n*n）,空间复杂度为O(n*n)

谢谢观看

参考文献

【1】 杨圣洪 马乐等著 。离散数学【M】。北京：科学出版社，2018.8：197-200.

【2】 Anany Levitin 著 潘彦译 。算法设计与分析基础【M】。北京：清华大学出版社，2015：336-340.

【3】百度百科