

华东师范大学数据科学与工程学院上机实践报告

课程名称：算法设计与分析

年级：21 级

上机实践成绩：

指导教师：金澈清

姓名：杨茜雅

上机实践名称：并查集

学号：

上机实践日期：6.2

10215501435

上机实践编号：No.14

一、目的

1. 了解 NP 问题的现实应用
2. 尝试解决 NP 问题

二、内容与设计思想

1. 解决 OJ 上的练习题，习题描述如下所述。
2. 尝试使用动态规划解决进行解决。
3. 总结算法的实现思路以及时间复杂度。

题目：

黄同学准备在暑假游历 A 国的所有城市。

A 国共有 n 座城市，黄同学从 1 号城市出发，分别去往所有城市且每个城市只去一次，然后再回到 1 号城市。 $(1 \leq n \leq 10)$

不同城市之间都有火车直达，票价在 1-100 之间。

请你帮黄同学计算一下这趟旅行的最小花销。黄同学感激不尽。

输入：

第一行为城市个数 n 。 $(1 \leq n \leq 10)$

第 2 到 $n+1$ 行为票价矩阵(n 行 \times n 列)。

输出：

输出一个整数，表示最小花销。回车结束。

三、使用环境

推荐使用 C/C++ 集成编译环境。

不建议随意改变本文档以上部分的结构，以下部分可以按需扩充。但不要缺少“实验过

程”和“总结”。

四、实验过程

```
#include<bits/stdc++.h>
using namespace std;
typedef int LL;
const int N=20,M=1<<N;
LL g[N][N],f[M][N];//f(i,j)表示从0走到j,走过所有点是i的所有路径
int n;
int main()
{
    cin>>n;
    for(int i=0;i<n;i++)
        for(int j=0;j<n;j++)
            cin>>g[i][j];

    memset(f,0x3f,sizeof f);

    f[1][0]=0;
    for(int i=0;i<(1<<n);i++)//集合状态
        for(int j=0;j<n;j++)
            if(i>>j&1)//看j点有没有到
                for(int k=0;k<n;k++)//枚举倒数第二个点
                    if(((i-(1<<j))>>k)&1)//i集合出去j点后
                        k点也要存在
                        f[i][j]=min(f[i][j],f[i-
(1<<j)][k]+g[k][j]);//状态计算最小值

    int res=0x3f3f3f3f;
    for(int i=1;i<n;i++)
    {
        res=min(res,f[(1<<n)-1][i]+g[i][0]);
    }
```

```
cout<<res<<endl;  
  
return 0;  
}
```

实现思路:

这题是很显然地 NPC 问题,暴力枚举的话时间复杂度为 $O(n*n!)$ (对所有节点进行全排序并且计算路径花销),时间复杂度很高

动态规划:

运用二进制和位运算实现状压 dp,二进制表达有无抵达节点 i,总共需要 2^n 次遍历,之后有大小比较,需要 n^2 次,所以时间复杂度为 $O(n^2*2^N)$

状态转移方程为: $f(i, j) = \min(f(i-k, k) + a(k, j))$

点全部达到的标志是在 $f[s(n), n-1]$ 中 $s(n) = (1 \ll n) - 1$

其中关键是到达 j 后,枚举倒数第二个节点,并且用它进行分类

五、总结

通过动态规划的方式迭代寻求子问题最优解,对 NPC 问题和其时间复杂度有了初步的认识。