

# 中山大学数据科学与计算机学院本科生实验报告

## (2016 学年春季学期)

课程名称: Algorithm design

任课教师: 张子臻

年级	1501	专业 (方向)	移动信息工程
学号	15352015	姓名	曹广杰
电话	13727022190	Email	<a href="mailto:1553118845@qq.com">1553118845@qq.com</a>
开始日期	2017/3/25	完成日期	2017/3/25

### 1. 实验题目

### 2. 实验目的

有编号 1 到 N 的 N 个城市，问从 1 号城市出发，遍历完所有的城市并最后停留在 N 号城市的最短路径长度；

### 3. 程序设计

#### 划归子问题

#### 无序点最短路径

#### 动态规划

#### 整体解决问题

1. 需要计算从第一个点到第 N 个点的最短距离，首先排除所有线路可能性的遍历，如果暴力法不能解决该问题。就需要设计一些算法以适应该问题。鉴于该题目有一些似曾相似的特点，比如最短、无向等，可以考虑将该问题划分成几个子问题；
2. 因素一：起讫点确定。所以区别于最小生成树问题，区别于寻找最小距离问题，所以不能使用 djikstra 算法或者弗洛伊德算法；
3. 因素二：即便是没有起讫点问题的影响，我们在考虑最短的遍历路径问题的时候，依然需要考虑最短的遍历路径应该是一个连线，也就意味着每一个路径必然只走一遍，绝无回头路之说，因为在欧几里得几何中三角形两边和大于第三边的规律。即路径是唯一的；
4. 以上分析可以归纳出，在不考虑起讫点的前提下，其余的无序的点的最小遍历是一个典型的最小路径问题，而对于无序的最小路径问题，是存在有效的算法的。更多的，即便已知起始点，其余点都是无序的，这种情况下，最小路径问题的解法依然是存在并且高效的；
5. 无序点的最小路径问题。无序点的最小路径问题，从几何的角度来讲，圆环式的放射型结构可以在一定程度上简化该问题（主要依据还是不走回头路）。当然，在二进制计算机中计算这种放射型问题需要大量的浮点数运算。此处参照他人的代码，意识到对于所有的数据进行遍历也可以高效（数据量为 20 的条件下），就是使用动态规划算法；
6. 这里的状态有几个要素，一则已经走过的路，二则将要走的城。于是对这两个元素进行状态的描述继而寻找状态之间的关系，依然是多维数组的数据结构，依然是循环嵌套中的状态转移方程；

7. 在实现了以上点的链接之后，对最后一个点进行一次遍历尝试，就遍历了所有的点；

## 4. 程序运行与测试

```
ude <cmath>
ne in C:\Users\Phelip\Desktop\surpurlink\dailycode\Alg\Graph\Travelingsale
name 1
4
0 0
[25], 0 1
e dst 1 0
dist( 1 1
3.41
ouble
st[b]
-----
Process exited after 8.533 seconds with return value 0
请按任意键继续. . .
```

## 5. 实验总结与心得

1. 需要承认一点，在计算无序的点之间的最短路径问题时候，个人认为动态规划并不是最有效的算法，但是是有现有模型可以使用的，也因此编程时比较高效吧，毕竟机器目前已经如此；
2. 最大的启发就是这个问题的划归思想，真正让我感觉到各种元素之间巧妙的融合，故而即便该题目不是练习题目之一，依然任性滴写下报告；
3. 动态规划的状态分析其实非常考究，这是一个建模的过程，一个巧妙的模型可以简化很多计算过程。另外，动态规划的递推过程实在很精妙；

## 附录、提交文件清单

15352015-caogj-week7;  
TSP.cpp;