

## 算法设计与分析实验报告

实验名称：货郎担问题(分支限界法)

### 一、问题陈述，相关背景、应用及研究现状的综述分析

#### 1.问题陈述：

有一推销员，欲到  $n$  ( $n \leq 10$ ) 个城市推销产品。为了节省旅行费用，在出发前他查清了任意两个城市间的旅行费用，想找到一条旅行路线，仅经过每个城市一次，且使旅行费用最少。本问题已知城市  $n$ ，和表示任意两个城市间费用的花费矩阵。试求最短路径及其费用；

## 二、模型拟制、算法设计和正确性证明

该问题的解空间树为排列树，在搜索解空间树时，采用最小耗费优先的方式搜索，在每一个活结点处，计算其子树对应的路径费用的下限，从当前活结点中选择费用下限最小的那一个成为扩展节点，然后搜索其儿子节点，重复选择最有利的结点成为扩展节点，使搜索朝着解空间树上最优解的分支推进。

利用小根堆构造优先队列，其中的结点排序的依据是子树的路径费用下限，每一个结点还存有当前的费用，到该结点的路径等信息。搜索解空间树，直到产生一个最优解或不再有活结点。

```
//构建结点类
class HeapNode
{
public:
    float cost_lower_bound, current_cost, rest_cost;
    int s;
    int x[vertex_number];
};
```

### 三、时间和空间复杂性分析

#### 解空间树为排列树

时间复杂度：  $O(n!)$

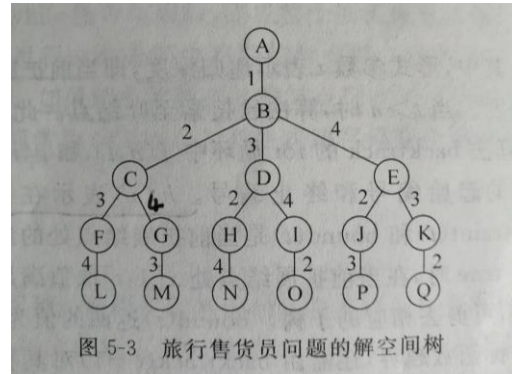
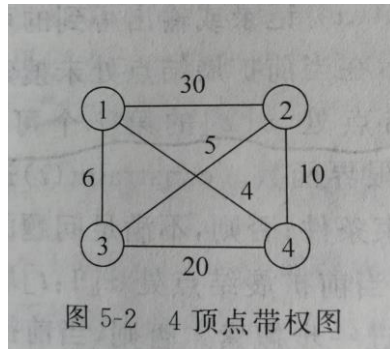
空间复杂度：

需要二维数组存储数据，空间复杂度为  $O(n^2)$ ；

## 四、程序实现和实验测试过程

源程序见 Traversal\_salesman.cpp

测试过程：



```
C:\Users\xx\Desktop\traversal_salesman\bin\Debug\traversal_salesman.exe
1 4 2 3 1
best cost is 25
Process returned 0 (0x0)   execution time : 0.079 s
Press any key to continue.
```

## 五、总结

和回溯法不同，回溯法是深度优先搜索，分支限界法是广度优先搜索，但是两者都有剪枝函数以加快搜索速度。