实验四 旅行售货员TSP问题的回溯法求解探索

**一、实验内容**

如图1所示，节点代表城市，节点之间的边代表城市之间的路径。每个城市都有一条进入路径和离开路径，不同的路径将耗费不同的旅费。旅行售货员选择城市1作为出发城市，途经其他每个城市，要求每个城市必须经过一次，并且只能经过一次，求一条具有最小耗费的路径，该路径从城市1出发，经过其余5个城市后，最后返回城市1。

采用C/C++/Java/Python语言，采用回溯法，使用递归或非递归的方法，求解旅行售货员问题。要求完成以下内容：

1. 设计采用的界限函数（剪枝函数）；

2. 给出回溯过程中对结点采用的剪枝策略。

3. 编写基于回溯法的算法代码，求出一条最短回路及其长度；

4. 画出回溯搜索过程中生成的解空间树，说明发生剪枝的结点，以及树中各个叶结点、非叶结点对应的路径长度。



**二、参考代码**

（1）代码如下

V：顶点集合

n：顶点个数

w[n][n]：权重矩阵，表示无边，否则表示边上的权重

x[n]：路径序列

x\_best[n]：最短路径序列

cv：当前最短路径长度

v\_best：最短路径长度

**递归回溯**

void TspDFS(i)

begin

if (i = n) then // 结束

cv  cv + w[x[n-1], 0]

if (cv < v\_best) then // 小于最短路径

(x\_best, v\_best)  (x, cv) // 更新最优解

end if

end if

foreach (uV  ux[0,1..i-1] ) do // 剩余可选城市

if (cv + w[ x[i-1], u] < v\_best) then // 加入后小于最短路径

x[i] = u // 加入

cv  cv + w[ x[i-1], u] // 更新当前解

TspDFS(i+1) // 搜索下一层

x[i] = 0 // 回溯

cv  cv - w[ x[i-1], u]

end if

end foreach

end

**非递归回溯**

void TspDFS()

begin

i = 0

while ( i >= 0) do

foreach (uV  ux[0,1..i-1]) do // 剩余可选城市

x[i] = u // 加入

cv  cv + w[ x[i-1], u] // 更新当前解

if (cv < v\_best) then // 加入后小于最短路径

if (i = n-1) then // 结束

cv  cv + w[ x[n-1], 0]

if (cv < v\_best) then // 小于最短路径

(x\_best, v\_best)  (x, cv) // 更新最优解

end if

else

i  i + 1 // 搜索下一层

end if

else

i  i - 1 // 回溯

cv  cv - w[ x[i-1], u]

end if

end foreach

end while

end

**三、界限函数（剪枝函数）及搜索树（参考）**

界限函数cv：到目前为止所走过城市对应的部分路径的总长度cv

剪枝条件一：当i<n时，如果x[i]添加到当前路径已经不小于当前最优路径，则不再继续搜索，

cv + w[x[i-1], x[i]] <= v\_best

条件二：当i<n时，如果i与其子结点之间没有边，即路径长度w[x[i-1]，x[i]]为正无穷时，剪枝。

（2）

最短回路：12341 （1321），长度为：35

搜素树如下：



被剪枝的节点（2分）：第3层第3个节点，

【或者：第5层第2、3、4个节点比较后被舍弃】