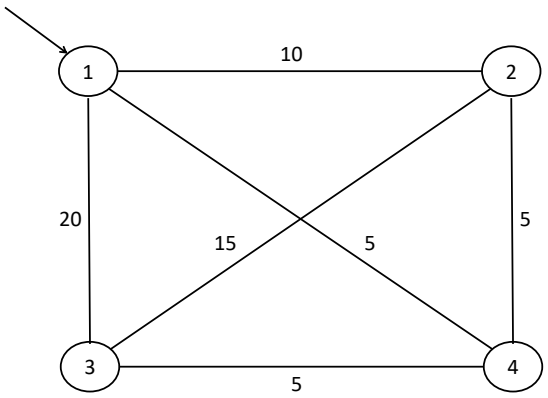
实验六 旅行售货员 TSP 问题的回溯法求解探索

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 (u∈ V  ∧  u∉ x[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 (u∈ V  ∧  u∉ x[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，

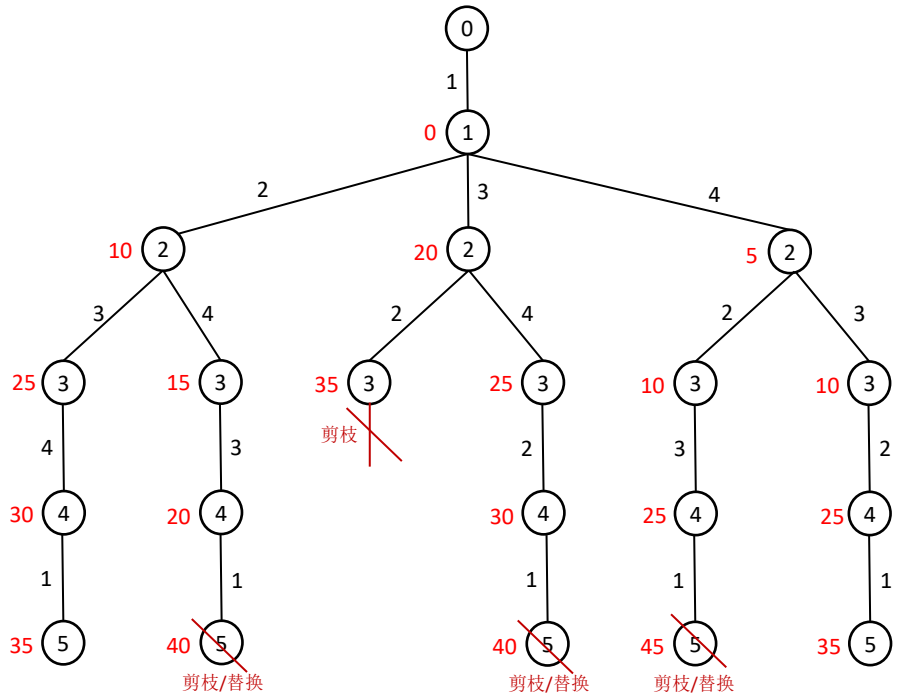
cv + w[x[i-1], x[i]] <= v_best

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：1→2→3→4→1 （1→ 4 →3→2→1），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】