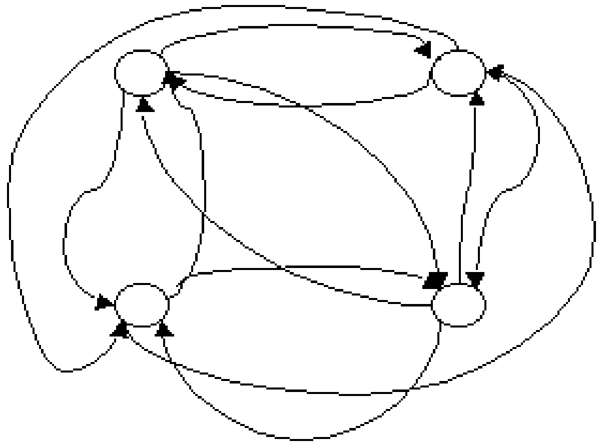
# 货郎担问题

# 问题定义

$$C=\begin{bmatrix}\infty & 2 & 1 & 3\\ 1 & \infty & 6 & 1\\ 3 & 4 & \infty & 4\\ 7 & 1 & 5 & \infty\end{bmatrix}$$



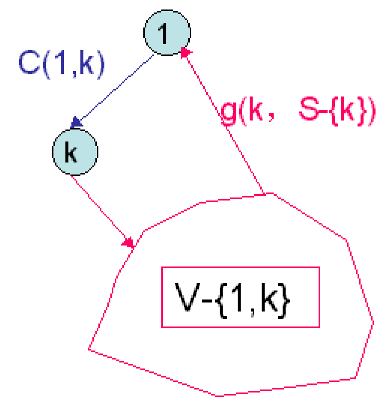
货郎担问题要从图G的所有周游路线中求取具有最小成本的周游路线，而由始点出发的周游路线一共有(n-1)!条，即等于除始结点外的n-1个结点的排列数，因此货郎担问题是一个排列问题。

n个物体有n!种排列，

通过枚举(n-1)!条周游路线，从中找出一条具有最小成本的周游路线的算法，其计算时间显然为O(n!) (n!>O($2^n$))。

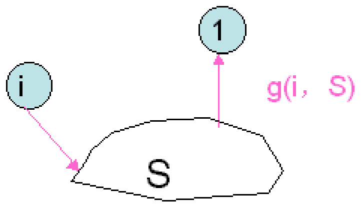
**周游路线最优值计算递推式：**

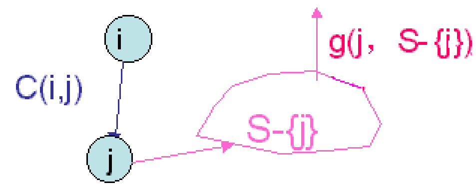
不失一般性，假设周游路线是开始于结点1并终止于结点1的一条简单路径。每一条周游路线都由一条边<1，k>和一条由结点k到结点1的路径所组成，其中k∈V－{1}；而这条由结点k到结点1的路径通过V－{1，k}的每个结点各一次。容易看出，如果这条周游路线是最优的，那么这条由k到1的路径必定是通过V－{1，k}中所有结点的由k到l的最短路径，因此最优性原理成立。




从1到k，再经V-{1,k}的顶点各一次回到1的周游路线

- 全解 = 1→k + min (k → 1)
- 枚举k，算出k → 1的最小，则全解就完成了。




设g(i，S)是由结点i开始，通过S中的所有结点，在结点l终止的一条最短路径长度。

g(1，V—{1})是一条最优的周游路线长度。于是可以得出:

$$g(1, V-\{1\}) = \min\{\, c(1,k) + g(k, V - \{1,k\}) \,\}$$

$$g(i, S) = \min\{\, c(i, j) + g(j, S - \{j\}) \,\}$$

**周游路线最优值计算实例：** G由矩阵C给出：

g(2，{ })=C(2,l)=5

g(3，{ })＝C(3,l)=6

g(4，{ })＝c(4,1)=8

$$C=\begin{bmatrix}\infty & 10 & 15 & 20\\ 5 & \infty & 9 & 10\\ 6 & 13 & \infty & 12\\ 8 & 8 & 9 & \infty\end{bmatrix}$$

g(2，{3})=c23+g(3， )=15　　g(2，{4})＝18

g(3，{2})=18　　g(3，{4})=20

g(4，{2})＝13　　g(4，{3})=15

g(2，{3，4})

＝ min{c(2,3)+g(3，{4})，c(2,4)+g(4，{3})}＝25

g(3，{2，4})

＝min{c(3,2)+g(2，{4})，c(3,4)+g(4，{2})}＝25

g(4，{2，3})

＝ min{c(4,2)+g(2，{3})，c(4,3)+g(3，{2})}＝23

最后得

g(1，{2，3，4})

=min{c(1,2)+g(2，{3，4})，c(1,3)+g(3，{2，4}，c(l,4)+g(4，{2，3}))

=min{35，40，43}

=35

# 作业

- 利用动态规划算法解决0-1背包问题:
- 要求:
  - 算出最优解，给出最优组合。
  - 实现基本算法，调试成功;
  - 实现改进算法，调试成功。