《计算机算法设计与分析》

# 第七章 分枝—限界法

马丙鹏

2020年11月19日

中国科学院大学
University of Chinese Academy of Sciences

# 第七章 分枝—限界法

- 7.1 一般方法
- 7.2 LC-检索
- 7.3 15-谜问题
- 7.4 LC-检索(续)
- 7.5 分枝-限界算法
- 7.6 0/1背包问题
- 7.7 货郎担问题

# 7.6 0/1背包问题

- 问题描述
  - 假定n个物品的重量$w_i$，效益值$p_i$和背包容量M均为已知的正数，
  - 目标函数(极小化)

$$-\sum_{1\le i\le j} p_i x_i$$

  - 约束条件

$$\sum_{1\le i\le j} w_i x_i \le M$$

$$x_i = 0\text{或}1, p_i > 0, w_i > 0, 1 \le i \le j$$

# 7.6 0/1背包问题

- 分枝限界法求解
  - 对0/1背包问题的解的结构，约束条件，目标函数和状态空间树的分析与回溯法时相同。
  - 求解0/1背包问题的一个关键问题是设计上下界函数。

(1)目标函数: $\text{cost}(X) = -\sum_{i=1}^{n} p_i x_i$ (求最小值)

(2)代价函数:

$$C(X)=\begin{cases} \text{cost}(X) & X\text{是答案结点} \\ \text{正无穷} & X\text{是叶结点但非答案结点} \\ \min\{c(\text{lchild}(X)),\ r(\text{lchild}(X))\} & X\text{代表非叶节点} \end{cases}$$

(3)上下界函数: U, $\hat{c}(X)$

# 7.6 0/1背包问题

- 分枝限界法求解
  - X是状态空间树上的结点，从根到X的部分向量为($x_1$, $x_2$, …, $x_k$)，
  - 背包的剩余载重为cu，以X为根的子树可以看成背包载重为cu，由剩余物品组成物品集的0/1背包的状态空间树：
  - 设Z代表子树X上一般背包问题的最优解($z_{k+1}$, $z_{k+2}$, …, $z_n$)，
  - ans代表0/1背包的最优解($x_{k+1}$, $x_{k+2}$, …, $x_n$)，
  - Y代表0/1背包的任一可行解($y_{k+1}$, $y_{k+2}$, …, $y_n$)，则必有：

$$-\sum_{i=k+1}^{n} p_i z_i \le -\sum_{i=k+1}^{n} p_i x_i \le -\sum_{i=k+1}^{n} p_i y_i$$

# 7.6 0/1背包问题

■ 分枝限界法求解

- 由此，可得0/1背包问题的代价函数和上下界函数:
  - (1)代价函数: $$\text{cost}(X) = -\sum_{i=1}^{k} p_i x_i - \sum_{i=k+1}^{n} p_i x_i$$
  - (2)下界函数: $$\hat{c}(X) = -\sum_{i=1}^{k} p_i x_i - \sum_{i=k+1}^{n} p_i z_i$$
    - ✓这是X子树上最小代价答案结点代价的下界估计值
  - (3)上界函数: $$U(X) = -\sum_{i=1}^{k} p_i x_i - \sum_{i=k+1}^{n} p_i y_i$$
    - ✓这是X子树上最小代价答案结点代价的上界估计值。

# 7.6 0/1背包问题

- 分枝限界法求解
  - 下界的定义
    - 按贪心法定义计算最大装入效益值的Bound函数：
    - $\hat{c}(X)$= Bound($-\sum_{i=1}^{j-1} p_i x_i$, $\sum_{i=1}^{j-1} w_i x_i$, j-1, M)

```
for i=k+1 to n do
      c = c+w(i);
      if c<M then
         b = b - p(i);
      else
         return b - (1+(c-M)/W(i)*p(i));
      endif
end
```

# 7.6 0/1背包问题

- 分枝限界法求解
  - 上界的定义
    - U(X) = LBound($-\sum_{i=1}^{j-1} p_i x_i$ , $\sum_{i=1}^{j-1} w_i x_i$ , j-1, M)
    - 其中j是结点X所在的层级

```
for i=k+1 to n do
    if c+w(i) ≤ M
then
        c = c + w(i);
        b= b - p(i);
    endif
repeat
```

# 7.6 0/1背包问题

■ 求解0/1背包问题的分枝限界法:

① 定义一个上界变量U，记录当前为止最小代价答案结点代价的上界值，

② 生成根结点，计算根结点的上下界值U和$\hat{c}(X)$，

③ 令$U = U + \varepsilon$;

④ 若X是答案结点，且该答案结点的收益prof小于U，则记录该答案结点，并令U= prof,

⑤ 若X不是答案结点，若左儿子结点可行，即(cu>w[k])，则生成左儿子结点($\hat{c}(X)$不变);
否则，计算右孩子的上下界，若右孩子的下界小于U，则生成右孩子结点，若右孩子的上界+ε小于U，则令U=U+ε;

⑥ 当活结点(优先权队列)为空时或当扩展结点的下界≥ U时，结束。

# 7.6 0/1背包问题

- 实例
  - $n=4, M=15$
  - $(p_1, p_2, p_3, p_4) = (10, 10, 12, 18)$
  - $(w_1, w_2, w_3, w_4) = (2, 4, 6, 9)$

# 7.6 0/1背包问题

n=4, M=15

$(p_1, p_2, p_3, p_4) = (10, 10, 12, 18)$

$(w_1, w_2, w_3, w_4) = (2, 4, 6, 9)$

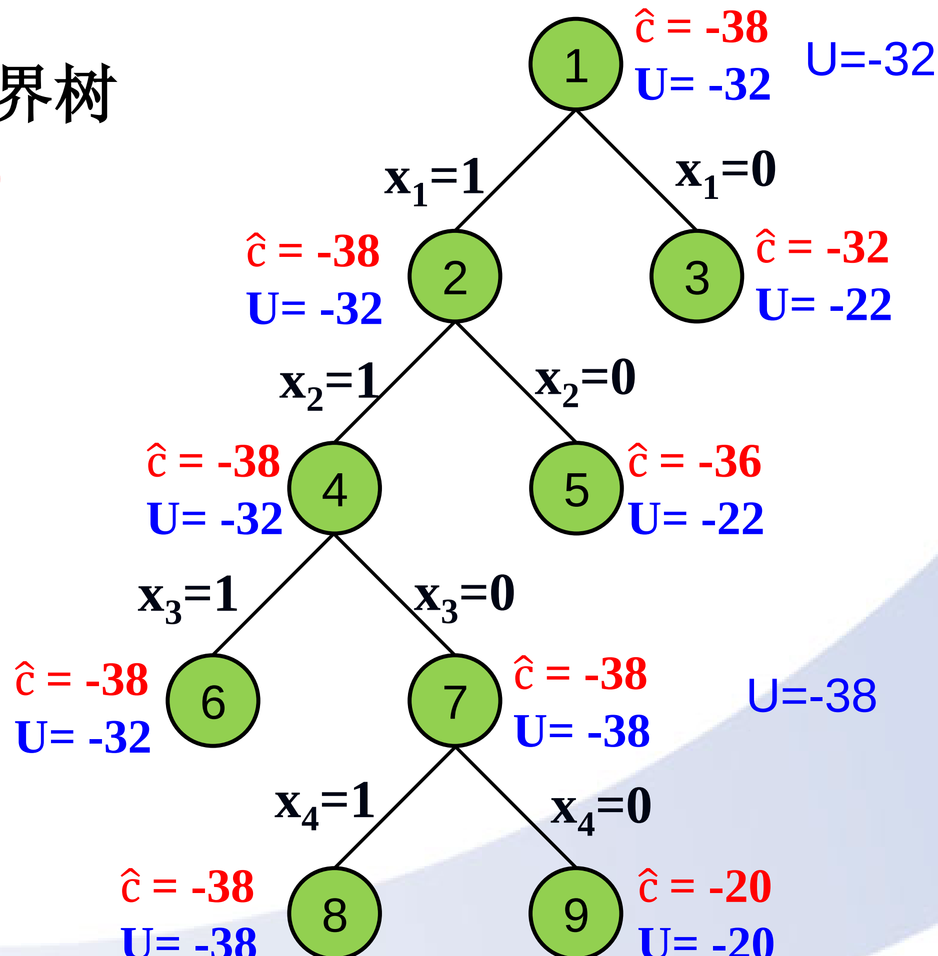
■ 实例

□LC分枝 –限界树

上面的数=ĉ(X)

下面的=U

大小固定元组




中国科学院大学 University of Chinese Academy of Sciences

# 7.6 0/1背包问题

n=4, M=15

$(p_1, p_2, p_3, p_4) = (10, 10, 12, 18)$

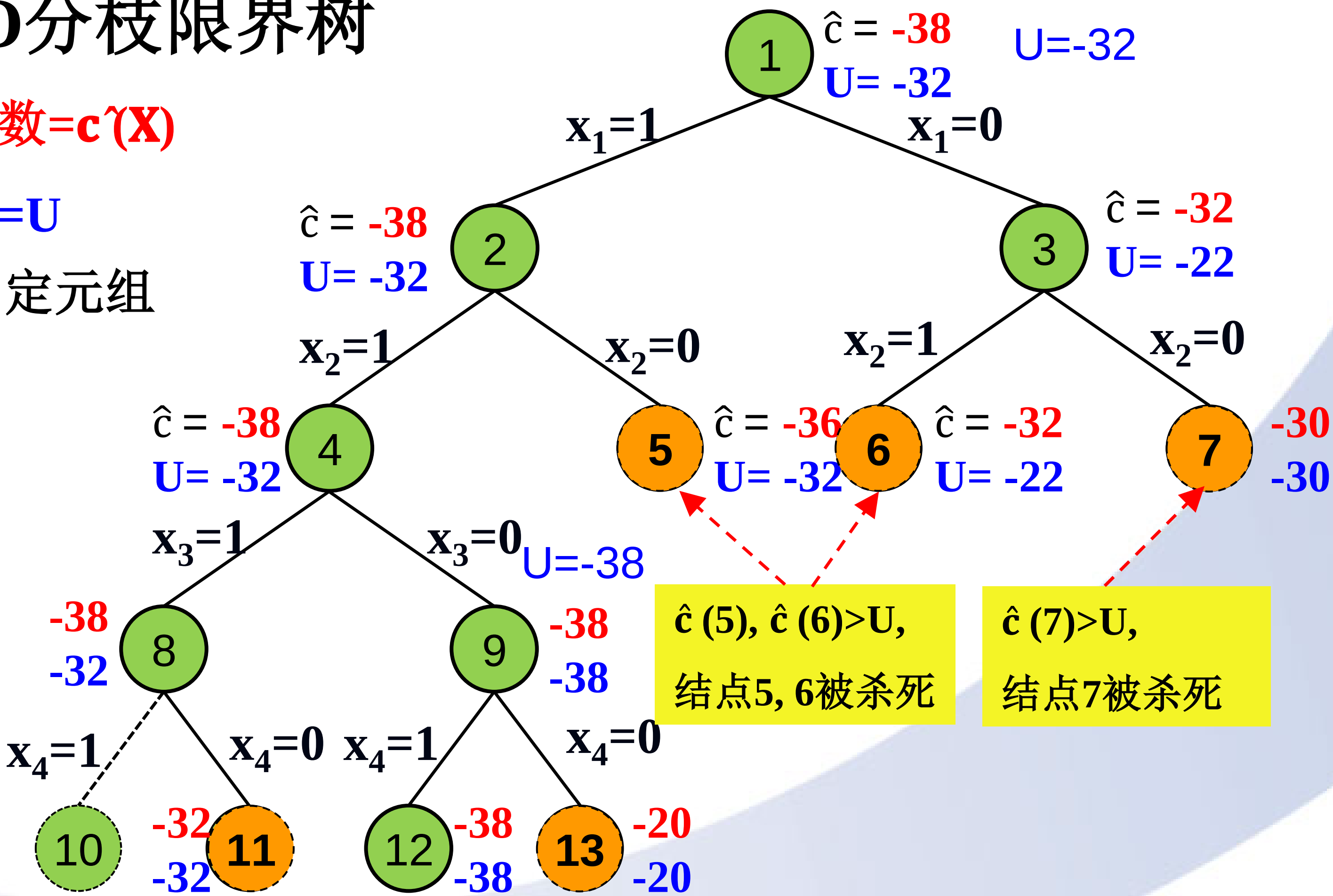
$(w_1, w_2, w_3, w_4) = (2, 4, 6, 9)$

## ■ FIFO分枝限界树

上面的数=ĉ(X)

下面的=U

大小固定元组



# 第七章 分枝—限界法

- 7.1 一般方法
- 7.2 LC-检索
- 7.3 15-谜问题
- 7.4 LC-检索(续)
- 7.5 分枝-限界算法
- 7.6 0/1背包问题
- 7.7 货郎担问题

# 7.7 货郎担问题

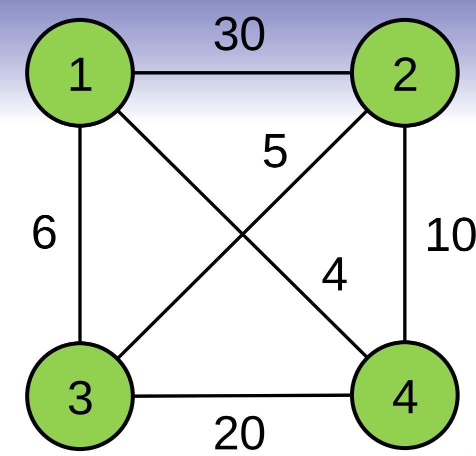

## ■ 问题描述

- □某售货员要到n个城市去推销商品，已知各城市之间的路程(或旅费)。他要选定一条从驻地出发，经过每个城市一次，最后回到驻地的路线，使总的路程(或总旅费)最小。
- □有向图：G=(V, E), |V|=n
- □成本邻接矩阵：

$$C=(c_{ij});\quad <i,j>\in E,\ c_{ij}>0;\quad <i,j>\notin E,\ c_{ij}=\infty.$$

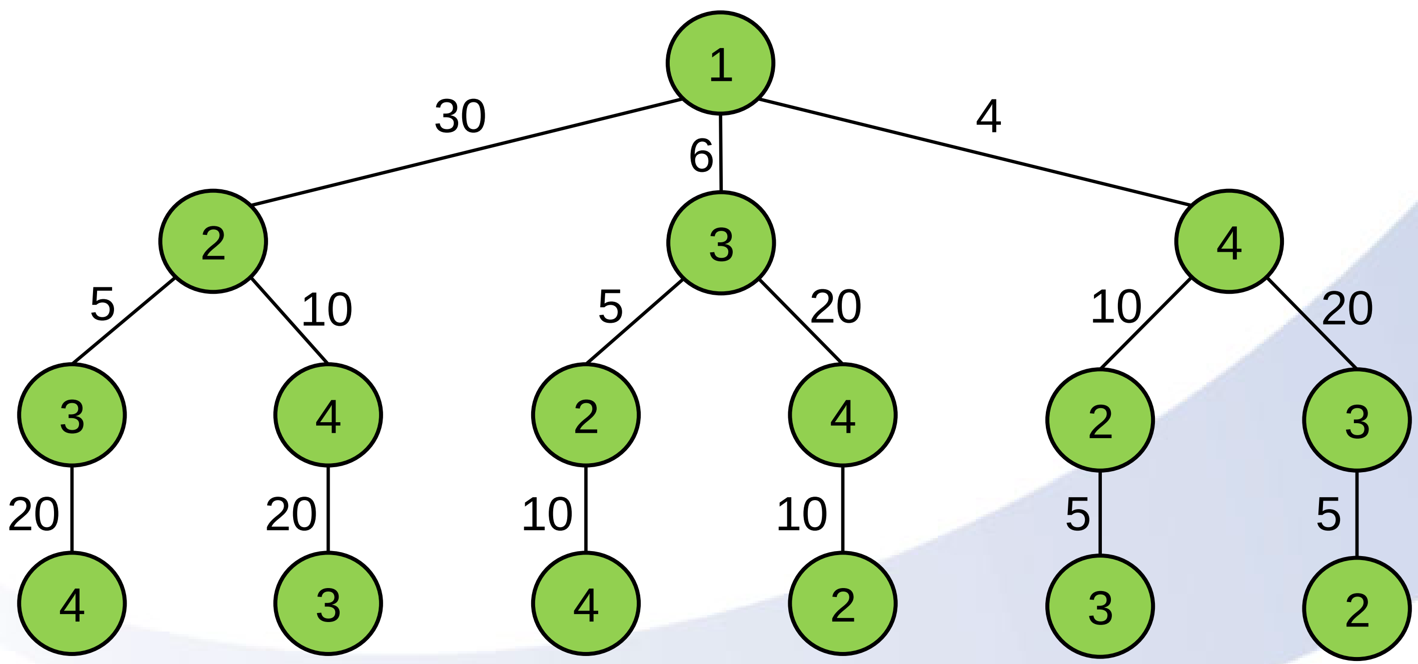
- □G的一条周游路线：包含n个节点的有向环，
- □周游路线成本：此路线上所有边的成本和，
- □求：具有最小成本的周游路线。

# 7.7 货郎担问题




■ 实例分析

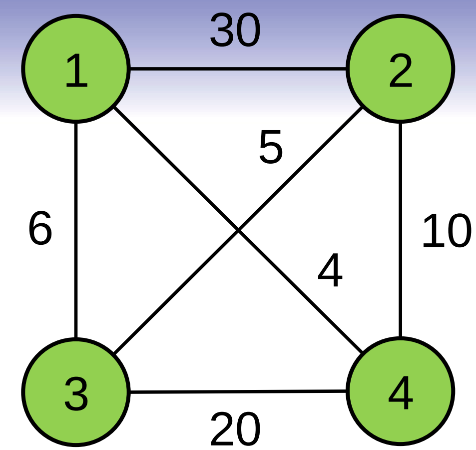
- □有n个村庄，每个村庄必须经过一次，也只能经过一次，求一条走遍全部村庄的最短路。

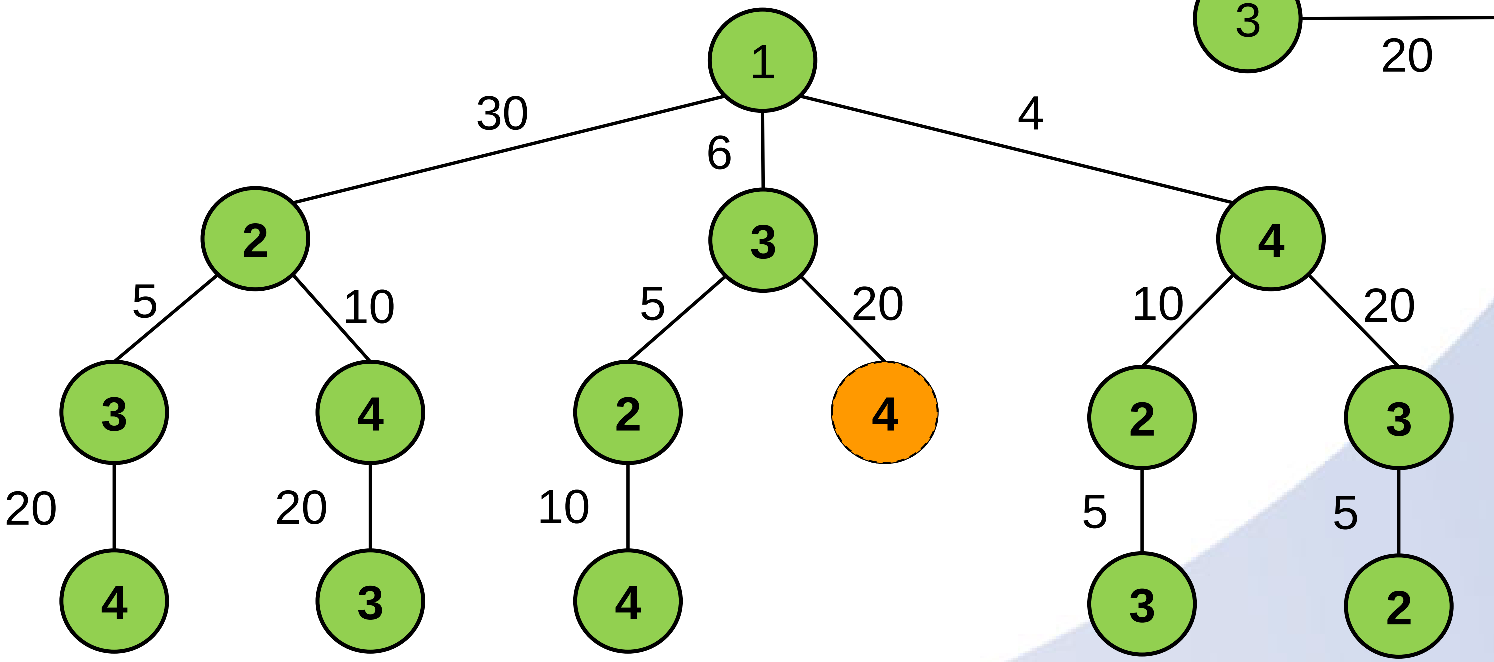


# 7.7 货郎担问题

■ 实例分析

□FIFO分枝限界树




30+5+20 +4=59

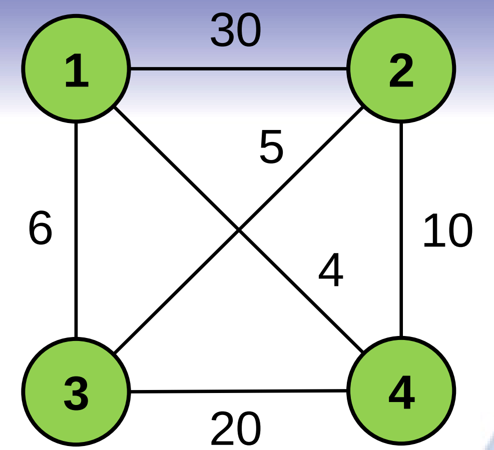
30+10+20 +6=66

6+5+10 +4=25

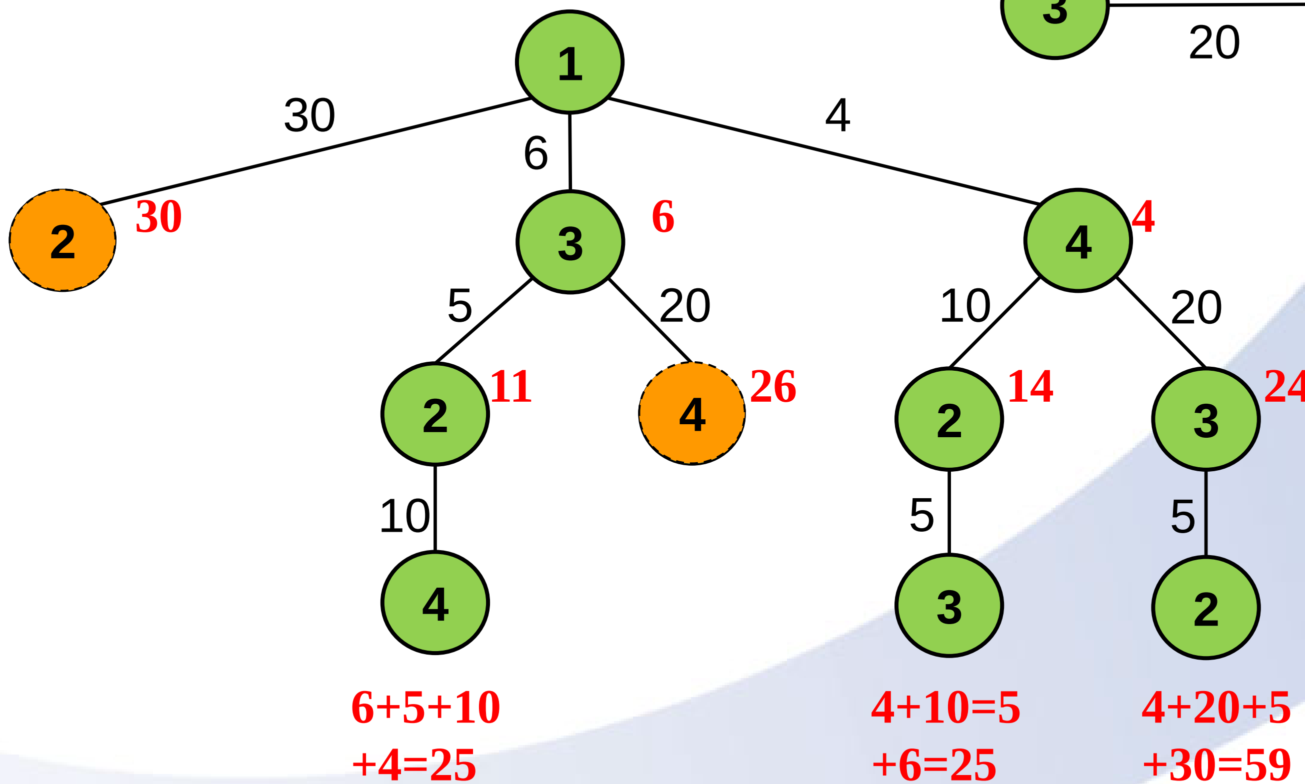
4+10=5 +6=25

4+20+5 +30=59

# 7.7 货郎担问题




- ■ 实例分析
  - □LC分枝 –限界树



# 7.7 货郎担问题

■ 分支限界法求解

□解空间表示

➢不失一般性，以节点1作为起点和终点，解空间为:

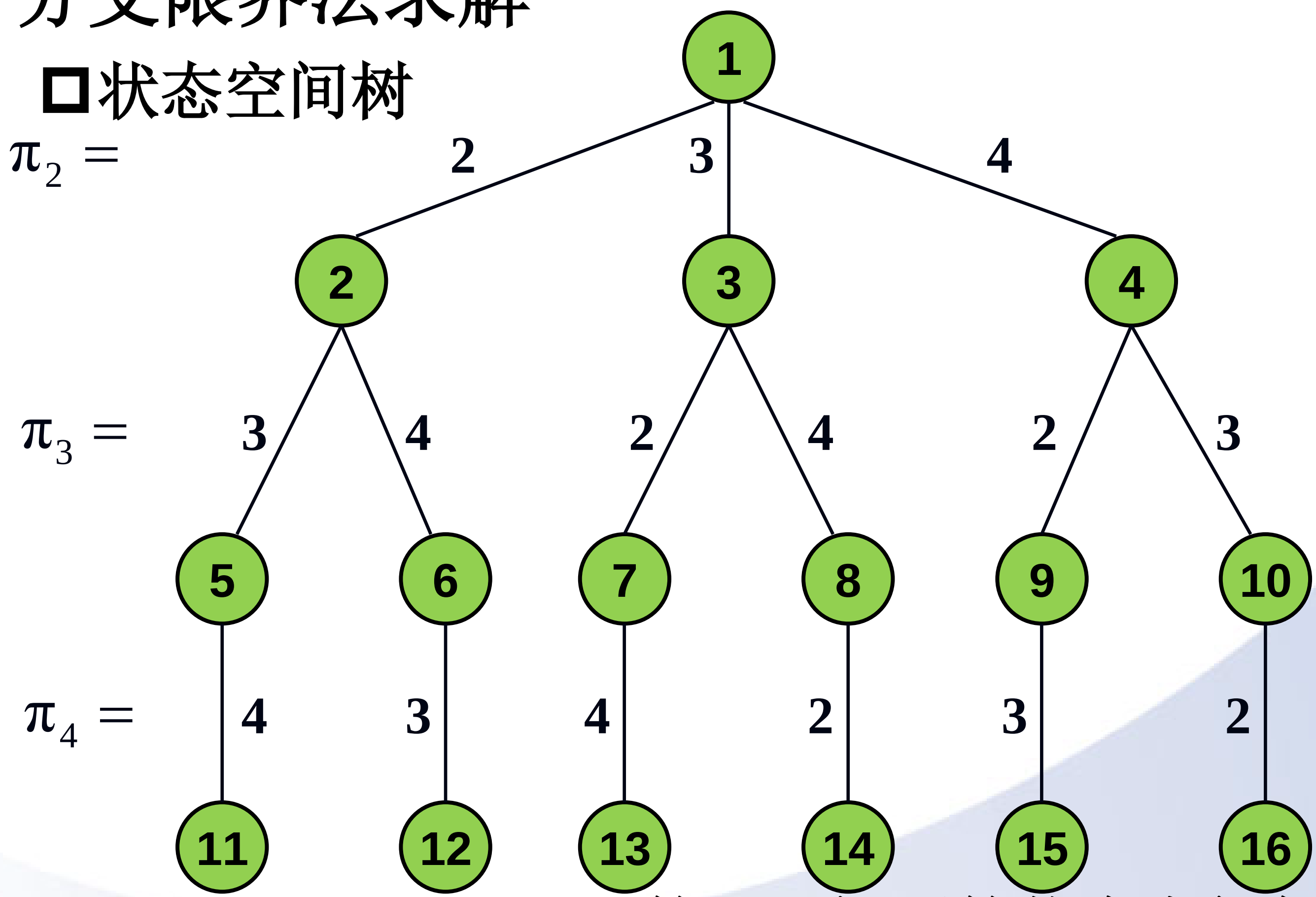
$$\begin{cases} S=\{\pi_1\pi_2\ldots\pi_n\pi_{n+1}\} \\ \pi_1=\pi_{n+1}=1 \\ \pi_i|_{i\in[2,n]} \in \{2, 3,\ldots,n\}, \pi_i \neq \pi_j \\ <\pi_i,\pi_{i+1}>|_{\forall i\in[1,n]} \in E,\ \pi_1\ldots\pi_{n+1} \in S \\ |S|=(n-1)! \end{cases}$$

□目标函数

➢路径长度最小。

# 7.7 货郎担问题

- 分支限界法求解
  - 状态空间树




**n=4, $\pi_1 = \pi_5 = 1$的TSP问题的状态空间树**

# 7.7 货郎担问题

■ 分支限界法求解

□C(X)的计算

$$C(X)=\begin{cases}\text{由根到X 的路径确定的周游路线成本} & \text{X是叶节点}\\ \text{子树X中最小成本叶节点的成本} & \text{X不是叶节点}\end{cases}$$

□归约矩阵

- 已规约行(列): 如果矩阵的一行(列)中至少包含一个零且其余元素非负，则称此行(或列)已归约，
- 归约矩阵: 如果一个矩阵的所有行和列均已归约, 则称此矩阵为归约矩阵，
- 归约的方法:对矩阵的一行(列)进行归约，可通过将该行(列)中的每个元素减去该行(列)的最小数进行，此最小数称为该行(列)的约数。

# 7.7 货郎担问题

- 分支限界法求解
  - 归约矩阵
    - 归约矩阵可通过逐行逐列归约一个代价矩阵而得到。
    - 矩阵归约的过程可以理解为:
      - ✓对原图象进行某种处理，使得边上权值变小，但图的结构不变。
    - 矩阵约数:
      - ✓所有行和所有列的约数之和称为矩阵约数：

$$L=\sum_{i=1}^{n} t_i+\sum_{j=1}^{n} r_j$$

# 7.7 货郎担问题

■ 分支限界法求解

□ 归约矩阵

$$C=\begin{bmatrix} \infty & 20 & 30 & 10 & 11 \\ 15 & \infty & 16 & 4 & 2 \\ 3 & 5 & \infty & 2 & 4 \\ 19 & 6 & 18 & \infty & 3 \\ 16 & 4 & 7 & 16 & \infty \end{bmatrix} \xrightarrow{\text{行规约}} \begin{bmatrix} \infty & 10 & 20 & 0 & 1 \\ 13 & \infty & 14 & 2 & 0 \\ 1 & 3 & \infty & 0 & 2 \\ 16 & 3 & 15 & \infty & 0 \\ 12 & 0 & 3 & 12 & \infty \end{bmatrix}$$

$$\xrightarrow[\text{得到规约矩阵}]{\text{列规约}} C'=\begin{bmatrix} \infty & 10 & 17 & 0 & 1 \\ 12 & \infty & 11 & 2 & 0 \\ 0 & 3 & \infty & 0 & 2 \\ 15 & 3 & 12 & \infty & 0 \\ 11 & 0 & 0 & 12 & \infty \end{bmatrix} \Rightarrow L=(10+2+2+3+4)+(1+3)=25$$

# 7.7 货郎担问题

- 分支限界法求解
  - 根的下界函数
    - ∵ 对G的每次周游只含有由i出发的n条边中的一条边，同样也只含有进入j的n条边中的一条。
    - ∴若对i行或者j列规约，即将此行或此列的元素减去t，则此次周游的成本减少t。
    - ∴ 原矩阵中的周游路线成本 = 规约矩阵的周游路线成本+矩阵约数。
    - ∴归约代价矩阵所代表的周游路径长度非负，故矩阵约数L≤原图的任意一条周游路径长度。
    - ∴可将L作为旅行商问题状态空间树根R的下界函数值。

# 7.7 货郎担问题

- 分支限界法求解
  - 非叶状态的下界函数
    - 为了得到C’(X)，对每个节点都定义一个规约矩阵；已知节点R的规约矩阵A，求儿子节点S(非叶节点)的规约矩阵的步骤：
      - ✓为保证这条周游路线采用边<i, j>，而不采用其他由i出发或者进入j的边，将A中i行j列置为∞；
      - ✓为了防止这条周游路线采用边<j, 1>从而构成环，将$A_{j1}$置为∞；
      - ✓对于那些不全为∞ 的行施行规约，得到S的规约矩阵B，设其约数为r, 得：
        $C'(S)=C'(R)+A_{ij}+r$

# 7.7 货郎担问题

■ 分支限界法求解

- □叶状态的下界函数
  - ➢若S是叶状态结点，由于一个叶状态唯一确定一条周游路径，可用此周游路径作为结点S的代价值，即 $$\hat{c}(X) = c(S)$$
- □上界函数
  - ➢对于树中任何状态结点X，令上界函数值u(X)=∞。

中国科学院大学
University of Chinese Academy of Sciences

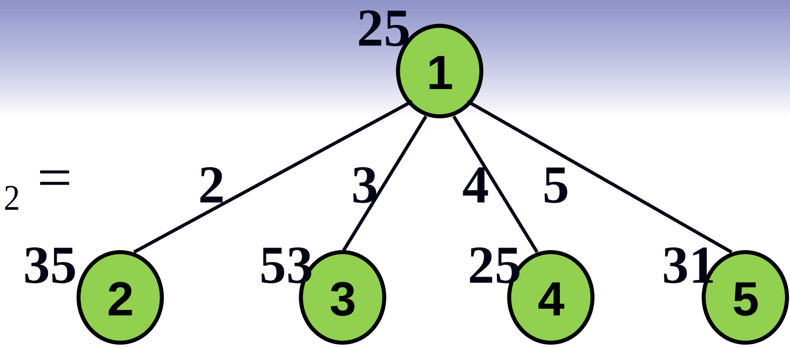
# 7.7 货郎担问题

- 分支限界法求解
  - 分支限界法求解该问题的过程
    - ① 生成状态空间树的根结点，作为扩展结点E，并令u=∞
    - ② 若该结点为答案结点，则计算其代价，修改u，否则，进第3步
    - ③ 生成扩展节点E的孩子结点，并分别计算他们的下界函数值，作为优先权，进优先权队列，选择优先级最高(即下界函数值最小) 的结点继续扩展，重复2, 3
    - ④ 若优先权队列中的所有结点都下界函数值都大于u，则结束。

# 7.7 货郎担问题

■ 实例

$\pi_2 =$




$$C=\begin{bmatrix} \infty & 20 & 30 & 10 & 11 \\ 15 & \infty & 16 & 4 & 2 \\ 3 & 5 & \infty & 2 & 4 \\ 19 & 6 & 18 & \infty & 3 \\ 16 & 4 & 7 & 16 & \infty \end{bmatrix} \xrightarrow{L=25} A(1)=\begin{bmatrix} \infty & 10 & 17 & 0 & 1 \\ 12 & \infty & 11 & 2 & 0 \\ 0 & 3 & \infty & 0 & 2 \\ 15 & 3 & 12 & \infty & 0 \\ 11 & 0 & 0 & 12 & \infty \end{bmatrix}$$

$\hat{c}(2)=25+10+0=35$

$\hat{c}(3)=25+17+11=53$

$\hat{c}(4)=25+0+0=25$

$\hat{c}(5)=25+1+5=31$

$$A(1)\xrightarrow{L=0} A(2)=\begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & 11 & 2 & 0 \\ 0 & \infty & \infty & 0 & 2 \\ 15 & \infty & 12 & \infty & 0 \\ 11 & \infty & 0 & 12 & \infty \end{bmatrix}, A(1)\xrightarrow{L=11} A(3)=\begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ 1 & \infty & \infty & 2 & 0 \\ \infty & 3 & \infty & 0 & 2 \\ 4 & 3 & \infty & \infty & 0 \\ 0 & 0 & \infty & 12 & \infty \end{bmatrix}$$

$$A(1)\xrightarrow{L=0} A(4)=\begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ 12 & \infty & 11 & \infty & 0 \\ 0 & 3 & \infty & \infty & 2 \\ \infty & 3 & 12 & \infty & 0 \\ 11 & 0 & 0 & \infty & \infty \end{bmatrix}, A(1)\xrightarrow{L=5} A(5)=\begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ 10 & \infty & 9 & 0 & \infty \\ 0 & 3 & \infty & 0 & \infty \\ 12 & 0 & 9 & \infty & \infty \\ \infty & 0 & 0 & 12 & \infty \end{bmatrix}$$

# 7.7 货郎担问题

■ 实例

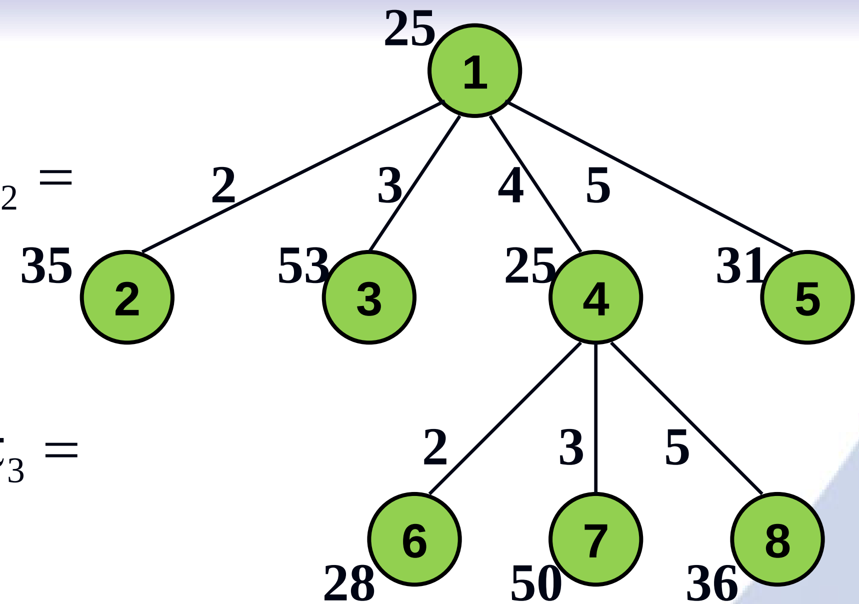
$$A(4)=\begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ 12 & \infty & 11 & \infty & 0 \\ 0 & 3 & \infty & \infty & 2 \\ \infty & 3 & 12 & \infty & 0 \\ 11 & 0 & 0 & \infty & \infty \end{bmatrix}$$

$$\xrightarrow{L=0} A(6)=\begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & 11 & \infty & 0 \\ 0 & \infty & \infty & \infty & 2 \\ \infty & \infty & \infty & \infty & \infty \\ 11 & \infty & 0 & \infty & \infty \end{bmatrix},$$

$$\xrightarrow{L=13} A(7)=\begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ 1 & \infty & \infty & \infty & 0 \\ \infty & 1 & \infty & \infty & 0 \\ \infty & \infty & \infty & \infty & \infty \\ 0 & 0 & \infty & \infty & \infty \end{bmatrix}, \xrightarrow{L=11} A(8)=\begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ 1 & \infty & 0 & \infty & \infty \\ 0 & 3 & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty \\ \infty & 0 & 0 & \infty & \infty \end{bmatrix}$$

$\pi_2 =$

$\pi_3 =$




ĉ(6)=25+3+0=28

ĉ(7)=25+12+13=50

ĉ(8)=25+0+11=36

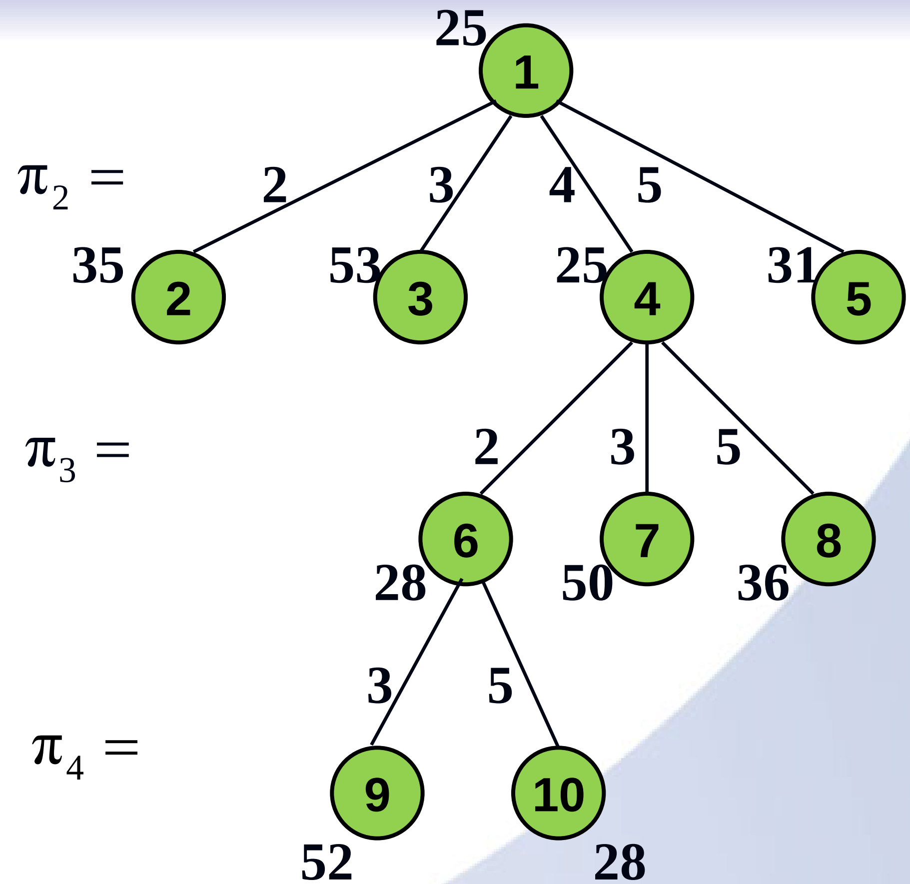
# 7.7 货郎担问题

■ 实例

$$A(6)=\begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & 11 & \infty & 0 \\ 0 & \infty & \infty & \infty & 2 \\ \infty & \infty & \infty & \infty & \infty \\ 11 & \infty & 0 & \infty & \infty \end{bmatrix}$$

$$\xrightarrow{L=13} A(9)=\begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & 0 \\ \infty & \infty & \infty & \infty & \infty \\ 0 & \infty & \infty & \infty & \infty \end{bmatrix},$$

$$\xrightarrow{L=0} A(10)=\begin{bmatrix} \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty \\ 0 & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & 0 & \infty & \infty \end{bmatrix},$$

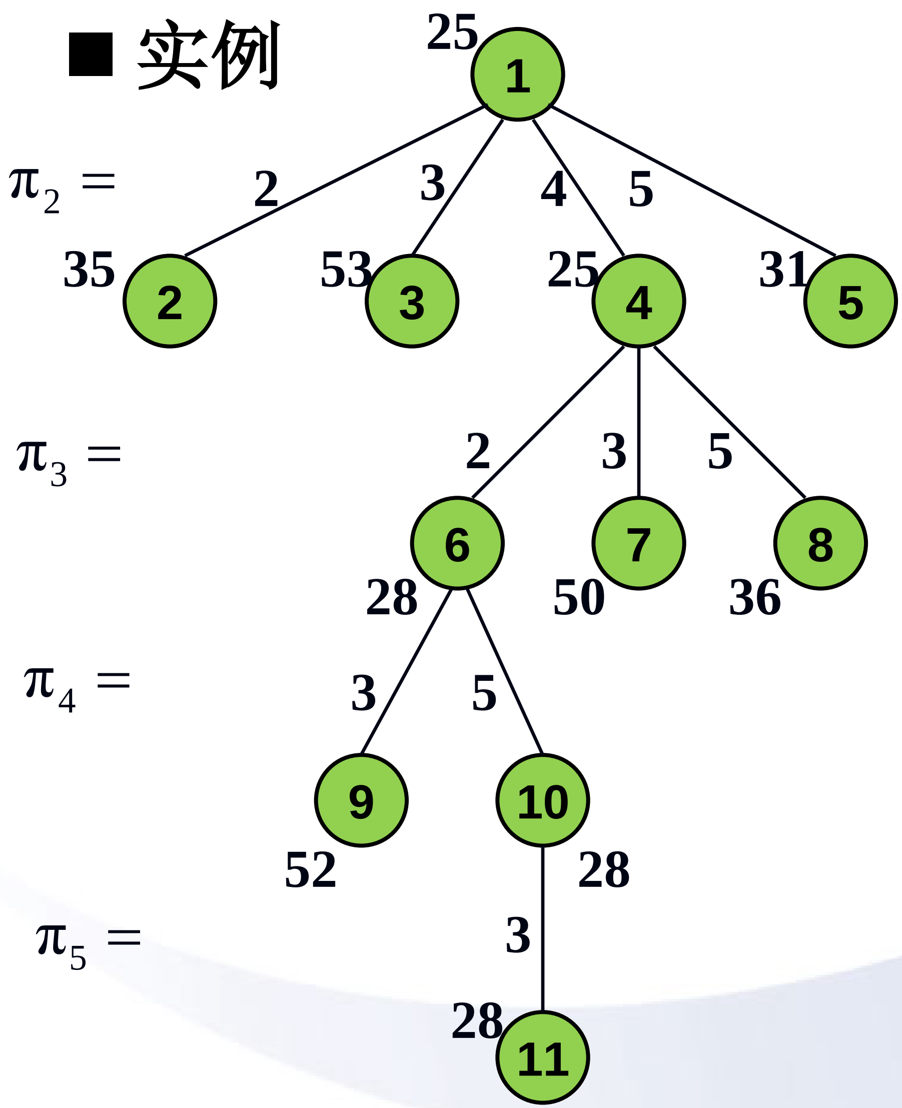



$\hat{c}(9)=28+11+13=52$

$\hat{c}(10)=28+0+0=28$

# 7.7 货郎担问题

■ 实例




最小成本周游路线：

**1，4，2，5，3，1**

$$
C=\begin{bmatrix} \infty & 20 & 30 & 10 & 11 \\ 15 & \infty & 16 & 4 & 2 \\ 3 & 5 & \infty & 2 & 4 \\ 19 & 6 & 18 & \infty & 3 \\ 16 & 4 & 7 & 16 & \infty \end{bmatrix}
$$

总成本为：**28**

# 作业-课后练习22

■ 问题描述

- 用LC分枝限界算法求解下面的0-1背包问题，并画出所生成的状态空间树。

① $N= 5, M=12, (p_1, p_2, ..., p_5) = (10, 15, 6, 8, 4), (w_1, w_2, ..., w_5) = (4, 6, 3, 4, 2)$ 。

- 用FIFO分枝限界算法求解下面的0-1背包问题，并画出所生成的状态空间树。

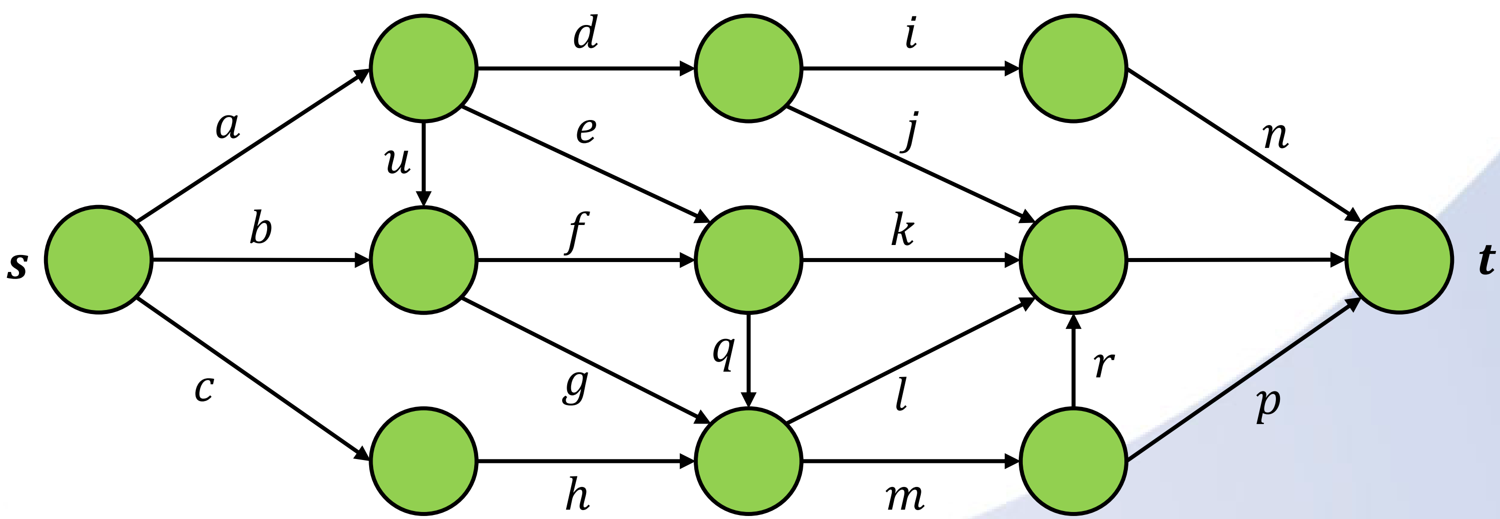
② $N= 5, M=15, (w_1, w_2, ..., w_5) = (p_1, p_2, ..., p_8) = (4, 4, 5, 8, 9)$。

■ 要求

- 作业提交到课程网站上

# 作业-算法实现8

- ■ 单源最短路径问题
  - □在下图所给的有向图G中，每一边都有一个非负边权。
  - □求图G的从源顶点s到目标顶点t之间的最短路径



# 作业-算法实现8

- 单源最短路径问题
  - 第一组测试参数

    const int n = 6;　//图顶点个数加1

    int c[n][n] = {{0,0,0,0,0,0}, {0,0,2,3,5000,5000}, {0,5000,0,1,2,5000}, {0,5000,5000,0,9,2}, {0,5000,5000,5000,0,2}, {0,5000,5000,5000,5000,0}}; //图的邻接矩阵
  - 第二组测试参数

    const int n = 5;　//图顶点个数加1

    int c[][n] = {{0,0,0,0,0}, {0,0,2,3,5000}, {0,5000,0,1,2}, {0,5000,5000,0,9}, {0,5000,5000,5000,0}};

# 作业-算法实现8

- 单源最短路径问题
  - 第三组测试参数

```
const int n = 12;   //图顶点个数加1
int c[n][n] = {{0,0,0,0,0,0,0,0,0,0,0,0},
{0,0,2,3,4,inf,inf,inf,inf,inf,inf,inf},
{0,inf,0,3,inf,7,2,inf,inf,inf,inf,inf},
{0,inf,inf,0,inf,inf,9,2,inf,inf,inf,inf},
{0,inf,inf,inf,0,inf,inf,2,inf,inf,inf,inf},
{0,inf,inf,inf,inf,0,inf,inf,3,3,inf,inf},
{0,inf,inf,inf,inf,inf,0,1,inf,3,inf,inf},
{0,inf,inf,inf,inf,inf,inf,0,inf,5,1,inf},
{0,inf,inf,inf,inf,inf,inf,inf,0,inf,inf,3},
{0,inf,inf,inf,inf,inf,inf,inf,inf,0,inf,2},
{0,inf,inf,inf,inf,inf,inf,inf,inf,2,inf,2},
{0,inf,inf,inf,inf,inf,inf,inf,inf,inf,inf,0},}; //邻接矩阵
```

# 考试时间

- 时间
  - 12月31日
  - 上午10:30至12:10
- 地点
  - 教1-107
  - 教1-113

# End