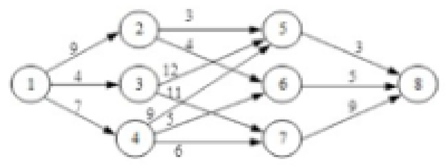
第六章 动态规划作业

1 对于多阶段决策问题，最优性原理总是成立的吗？若否，请给出一个最优性原理不成立的例子，并解释原因。

2 如下多段图，用多段图向前处理算法，求结点1到结点8的最小成本及最小成本路径。



3 采用序偶法求0/1背包问题，请给出一个出现最坏情况的例子，它使得$|S^i|=2^i$（$0\le i\le n$）。

4 对于0-2背包问题如下实例：有n=4种物品，每种物品最多有2件可用，且物品具有不可分割的特性，物品效益值分别为(p1,p2,p3,p4) =(6,3,8,4)，物品重量分别为(w1,w2,w3,w4) = (3,3,5,3)，背包容量M=14，求解目标是在满足背包容量限制的条件下，放入背包中物品效益值最大。

请给出使用动态规划法求解该0-2背包问题的递推关系式；若采用序偶法求最优解，给出求解过程及求解结果（包括最大效益值和最优决策序列）。

5 对于标识符集（a_1,a_2,a_3,a_4）=（end, goto, print, stop），已知各标识符被成功检索的概率分别为P(1)=1/20, P(2)=1/5, P(3)=1/10, P(4)=1/20，检索失败的概率分别为Q(0)=1/5, Q(1)=1/10, Q(2)=1/5, Q(3)=1/20, Q(4)=1/20；用构造最优二分检索树的算法OBST，对其计算W(i ,j)，R(i, j)和C(i, j)（0≤i, j≤4），并画出所构造的最优二分检索树的结构。

6 构造最优二分检索树的优化算法，把根结点k的检索区间限制在R(i,j-1)≤k≤R(i-1,j)，即求解的递推关系式为：

$$C(i,j)=\min_{R(i,j-1)\le k\le R(i+1,j)}\{C(i,k-1)+C(k,j)\}+W(i,j)$$

证明：此时构造最优二分检索树算法的计算时间为$O(n^2)$

7 【找零钱问题】已知有m种面值的硬币，其面值d1<d2<…<dm，且d1=1，每种硬币数量无限。现需找零钱的金额为n，问至少需要多少枚上述硬币。采用动态规划法求解该问题，请给出递推关系式。

8 【构造回文词问题】回文词是一种对称的字符串，一个回文词从左向右读和从右向左读得到的结果是一样的。对于任意给定的一个字符串，可以通过插入若干字符，变成一个回文词。要将给定字符串变成回文词，求所需插入的最少字符数。采用动态规划法解决该问题，请给出递推关系式。

9 【币值最大化问题】给定一排n个硬币，其面值均为正整数，分别表示为c1, c2,…cn。这些面值未必两两不同（也就是允许存在面值相同的硬币）。如何选择硬币，使得所选出硬币的原始位置互不相邻，且所选硬币的总金额最大?

设所选硬币的最大总金额为F(n)，要求

(1)给出F(n)的递推式;

(2)给出动态规划法求该问题的思路和步骤，并给出算法描述。

(3)针对币值最大化问题的一个实例：一排硬币面值分别为5,1,2,10,6,2 ，给出求解过程及结果。

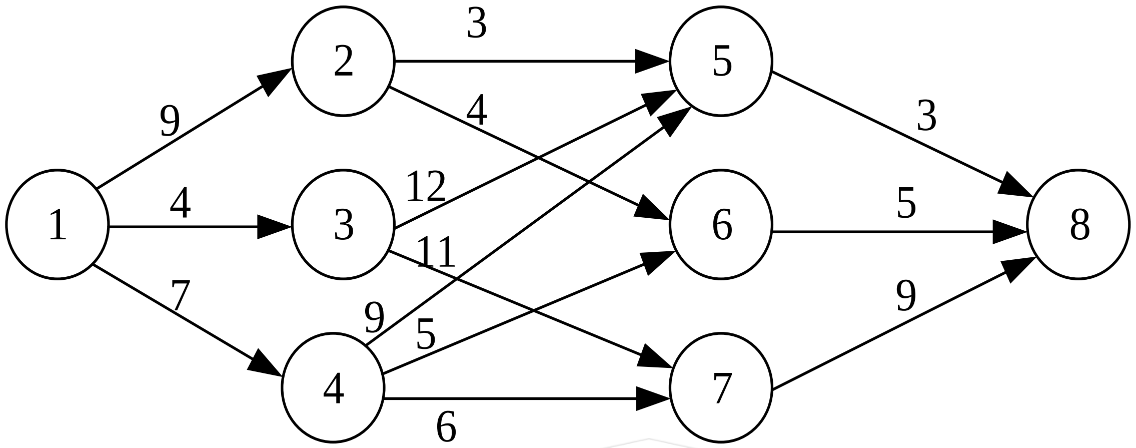
1. 最优性原理

- 对于多阶段决策问题，最优性原理总是成立的吗？若否，请给出一个最优性原理不成立的例子，并解释原因。

- 最优性原理：无论过程的初始状态和初始决策是什么，其余的决策都必须相对于初始决策所产生的状态构成一个最优决策序列。
- 也就是原问题的最优解依赖于子问题的最优解。
- 若多段图问题，以乘法为路径长度，当含有负权时，全局最优解不依赖于子问题的最优解，最优性原理不成立；
- 包含负长度环的任意两点间最短路径问题，最优性原理也不成立。

2. 多段图实例计算

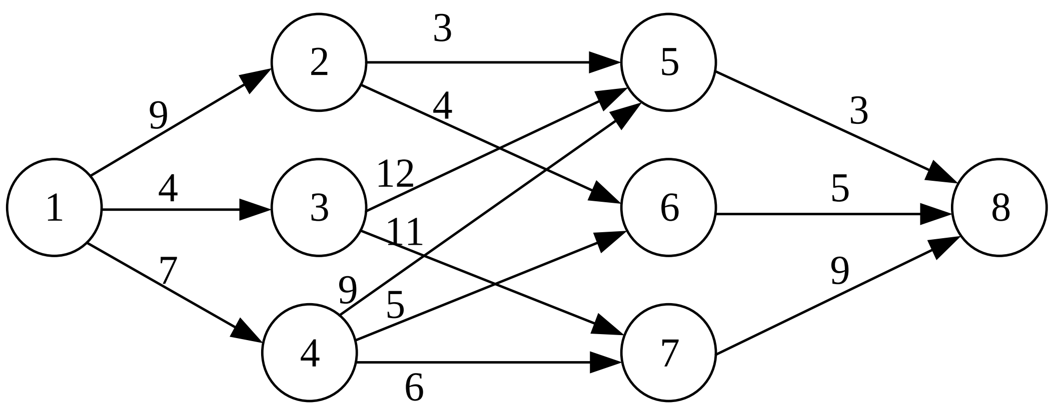
- 对如下多段图实例，用向前处理算法，求结点1到结点8的最小成本路径及其最小成本。要求给出计算公式及求解过程。



多段图向前处理的计算过程

COST(i, j)=min{ c(j, l)+COST(i+1, l)}

COST(8)=0
COST(7)=9
COST(6)=5
COST(5)=3



COST(4)=min{c(4,5)+COST(5) , c(4,6)+COST(6) , c(4,7)+COST(7)} =min{12,10, 15}=10
COST(3)=min{c(3,5)+COST(5) , c(3,7)+COST(7)} =min{15, 20}=15
COST(2)=min{c(2,5)+COST(5) , c(2,6)+COST(6)} =min{6, 9}=6

COST(1)=min{c(1,2)+COST(2) , c(1,3)+COST(3) , c(1,4)+COST(4)}
=min{15, 19, 17}=15

D(1)=2 D(2)=5 D(5)=8 D(8)=-1 最小成本:15，最小成本路径：1 2 5 8

3. 0/1背包问题出现最坏情况

- 给出一个使得0/1背包问题出现最坏情况的例子，它使得$|S^i|=2^i$，$0 \le i < n$。还要求对n的任意取值都适用。

取 $(P_1,P_2,\ldots,P_i,\ldots)$

$=(W_1,W_2,\ldots,W_i,\ldots)$

$=(2^0,2^1,\ldots,2^{i-1},\ldots)$

没有重复元素，也不会因为支配原则而删除元素，即出现最坏情况。可用归纳法证明此结论。

4. 0-2背包问题

- 对于0-2背包问题如下实例：有n=4种物品，每种物品最多有2件可用，且物品具有不可分割的特性，物品效益值分别为(p_1,p_2,p_3,p_4) =(6,3,8,4)，物品重量分别为(w_1,w_2,w_3,w_4) = (3,3,5,3)，背包容量M=14，求解目标是在满足背包容量限制的条件下，放入背包中物品效益值最大。
- 请给出使用动态规划法求解该0-2背包问题的递推关系式；若采用序偶方法求最优解，给出求解过程及求解结果（包括最优解及最大效益值）。

- 递推关系式：
  - 设$f_i(x)$是KNAP(1,i,x)最优解的值
  - $f_i(x)=\max\{f_{i-1}(x), f_{i-1}(x-w_i)+p_i, f_{i-1}(x-2w_i)+2p_i\}$
- 序偶集合
  - S0={(0,0)}
  - S1={(0,0)(6,3)(12,6)}
  - S2={(0,0)(6,3)(12,6)(15,9)(18,12)}
  - S3={(0,0)(6,3)(12,6)(14,8)(15,9)(16,10)(20,11)(22,13)(23,14) }
  - S4={(0,0)(6,3)(12,6)(14,8)(16,9)(20,11)(22,13)(24,14) }
- 解向量
  - (2,0,1,1)

5. 构造最优二分检索树

- 对于标识符集(a_1,a_2,a_3,a_4)=(end, goto, print, stop)，已知成功检索概率为P(1)=1/20, P(2)=1/5, P(3)=1/10, P(4)=1/20; 不成功检索概率为Q(0)=1/5, Q(1)=1/10, Q(2)=1/5, Q(3)=1/20, Q(4)=1/20;
- 用构造最优二分检索树的算法OBST，对其计算W(i ,j)，R(i, j)和C(i, j) ($0 \le i, j \le 4$)，并画出所构造的最优二分检索树的结构。

- P: P(1)=1/20, P(2)=1/5, P(3)=1/10, P(4)=1/20
- Q: Q(0)=1/5, Q(1)=1/10, Q(2)=1/5, Q(3)=1/20, Q(4)=1/20

- P: P(1)=1, P(2)=4, P(3)=2, P(4)=1
- Q: Q(0)=4, Q(1)=2, Q(2)=4, Q(3)=1, Q(4)=1

$$W(i, j) = W(i, j-1) + P(j) + Q(j)$$

$$C(i, j) = \min_{i<k\leq j}\{C(i,k-1) + C(k, j)\} + W(i, j)$$

$$R(i, j) = k$$

矩阵W(i,j)

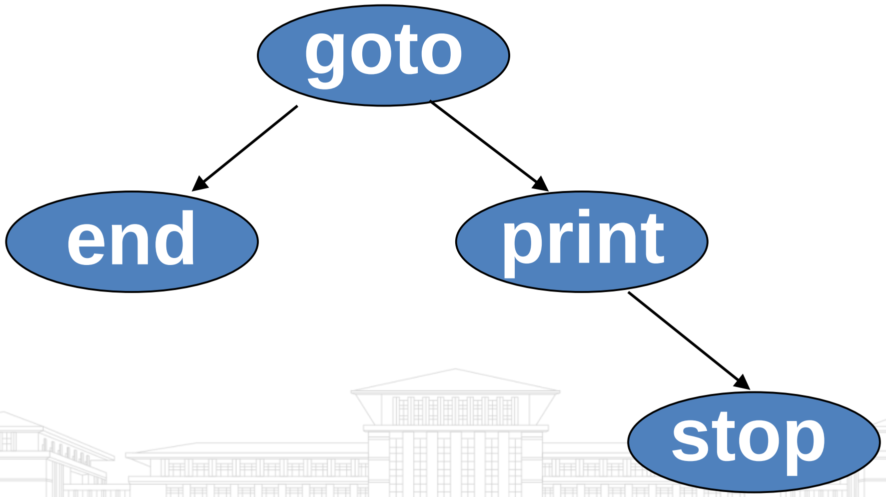
4	7	15	18	20
	2	10	13	15
		4	7	9
			1	3
				1

矩阵C(i,j)

0	7	22	32	39
	0	10	20	27
		0	7	12
			0	3
				0

矩阵R(i,j)

0	1	2	2	2
	0	2	2	2
		0	3	3
			0	4
				0



6. 构造最优二分检索树的优化算法

- 构造最优二分检索树的优化算法，把根结点k的检索区间限制在 $R(i,j-1) \le k \le R(i-1,j)$，即求解的递归关系式为：

$$C(i,j)=\min_{R(i,j-1)\le k\le R(i+1,j)}\{C(i,k-1)+C(k,j)\}+W(i,j)$$

- 证明：此时构造最优二分检索树算法的计算时间为 $O(n^2)$

```
procedure OBST(P,Q,n)  //最优二分检索树T_ij 成本C(i,j), 根R(i,j), 权W(i,j)
  real C(0:n,0:n),W(0:n,0:n); integer R(0:n,0:n)
  for i←0 to n-1  do //置初值
    (W(i,i),R(i,i),C(i,i))←(Q(i),0,0)
    (W(i,i+1),R(i,i+1),C(i,i+1))←(Q(i)+P(i+1)+Q(i+1), i+1, Q(i)+P(i+1)+Q(i+1))
  repeat
  (W(n,n),R(n,n),C(n,n))←(Q(n),0,0)
  for  m←2 to n do  //找有m个结点的最优树
     for  i←0 to n-m do
        j←i+m
        W(i,j)←W(i,j-1)+P(j)+Q(j)
        k←区间 [R(i,j-1),R(i+1,j)]中使{C(i,p-1)+C(p,j)}取最小值的p值
        C(i,j)←W(i,j)+C(i,k-1)+C(k,j)
        R(i,j)←k
     repeat
  repeat
end OBST
```

时间复杂度分析

$$\sum_{m=2}^{n}\sum_{i=0}^{n-m}(R(i+1,j)-R(i,j-1)+1)$$

$$=\sum_{m=2}^{n}\sum_{i=0}^{n-m}(R(i+1,i+m)-R(i,i+m-1)+1)$$

$$=\sum_{m=2}^{n}(R(n-m+1,n)-R(0,m-1)+n-m+1)$$

$O(n^2)$

7. 找零钱问题

- 已知有m种面值的硬币，其面值$d_1<d_2<\ldots<d_m$，且$d_1=1$，每种数量无限。现需找零钱的金额为n，问至少需要多少枚上述硬币。
- 采用动态规划法求解该问题，请给出递推关系式。

n是金额，F(n)是最少硬币数：

$F(0)=0$, $F(1)=1$

$F(n) = \min\{F(n-d_j)\} + 1$ 其中，$1 \le j \le m$

8. 构造回文词问题

- 回文词是一种对称的字符串，一个回文词从左向右读和从右向左读得到的结果是一样的。对于任意给定的一个字符串，可以通过插入若干字符，变成一个回文词。要将给定字符串变成回文词，求所需插入的最少字符数。采用动态规划法解决该问题，请给出递推关系式。
- 例: 对于字符串Ab3bd
  - 需要增加2个字符， Adb3bdA 或 dAb3bAd

回文词问题分析

- 将字符串表示为$C_1C_2..C_n$，不失一般性，原问题范围从C_i到C_j，n(i,j)表示需要增加的最小字符个数。

$$\mathbf{C_i\ C_{i+1}\ \cdots\ C_j}$$

- 若$C_i=C_j$，则不需添加字符，考虑$C_{i+1}\ldots C_{j-1}$；否则，考虑$C_{i+1}\ldots C_j$和$C_i\ldots C_{j-1}$这两种情况，从中选优：
  - 对于前者：需要在C_j之后插入C_i
  - 对于后者：需要在C_i之前插入C_j

递推关系式

- 当j-i≥1时
  - 若 $C_i=C_j$, 则n(i,j)=n(i+1,j-1)
  - 若 $C_i \neq C_j$, 则n(i,j)=min{n(i+1,j),n(i,j-1)}+1
- 当j-i<1时(初值)
  - n(i,j)=0

9. 币值最大化问题

- 给定一排n个硬币，面值均为正整数$c_1, c_2, \ldots c_n$，这些面值未必两两不同（换句话说，允许面值相同的硬币）.如何选择硬币，使得所选出硬币的原始位置互不相邻，且所选硬币的总金额最大？
- 设所选硬币的最大总金额为F(n)，要求:

  (1) 给出F(n)的递推关系式；

  (2) 给出动态规划法求该问题的思路和步骤，并给出算法描述。

  (3) 针对币值最大化问题的一个实例：一排硬币面值分别为5, 1, 2, 10, 6, 2，给出求解过程及结果。

（1） 递推关系式：F(n)为所选硬币的最大总金额：

$F(0)=0,\ F(1)=c_1$

$F(i) = \max\{c_i+F(i-2),\ F(i-1)\}$ 其中，$2 \le i \le n$

（2）对上述递归关系式，从前往后迭代求解，并记录决策值。

算法描述：

```
procedure  KNAPSACK02(C,n)
real F(n), j
F(0) ← 0  F(1) ← C(1)
for j←2 to n by 1do
    if F(j-1)>C(j)+F(j-2) then  F(j)← F(j-1)
      else F(j)← C(j)+F(j-2);
endif
repeat
回溯求决策序列
end KNAPSACK02
```

（3）一排硬币面值分别为5, 1, 2, 10, 6, 2， 给出求解过程及结果。

F(0)=0, F(1)=c1=5

F(2)=max{F(1),F(0)+c2}=max{5,1}=5

F(3)=max{F(2),F(1)+c3}=max{5,5+2}=7

F(4)=max{F(3),F(2)+c4}=max{7,5+10}=15

F(5)=max{F(4),F(3)+c5}=max{15,7+6}=15

F(6)=max{F(5),F(4)+c6}=max{15,15+2}=17

最大币值17.

回溯求得决策序列为(1,4,6)

本 章 结 束