动态规划算法

动态规划的实现分为迭代实现以及递归实现两种。

递归实现：时间复杂性高，空间较小

迭代实现：时间复杂性低，空间消耗多

动态规划算法的要素

(1) 划分子问题

用参数表达子问题的边界，将问题求解转 变成多步判断的过程.

(2) 确定优化函数

以该函数的极大(或极小)作为判断的依据，确定是否满足优化原则.

(3) 列出关于优化函数的递推方程 (或不等式)和边界条件

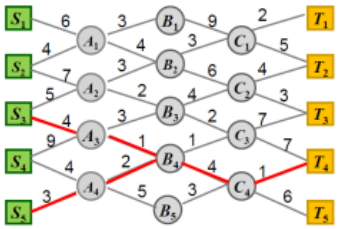
(4) 考虑是否需要设立标记函数

(5) 自底向上计算，以备忘录方法 (表格)存储中间结果

最短路径问题 P53页

最短路径问题

- 问题：
- 输入：
  - 起点集合 $\{S_1, S_2, \dots, S_n\}$，
  - 终点集合$\{T_1, T_2, \dots, T_m\}$，
  - 中间结点集，
  - 边集 E，对于任意边 e 有长度

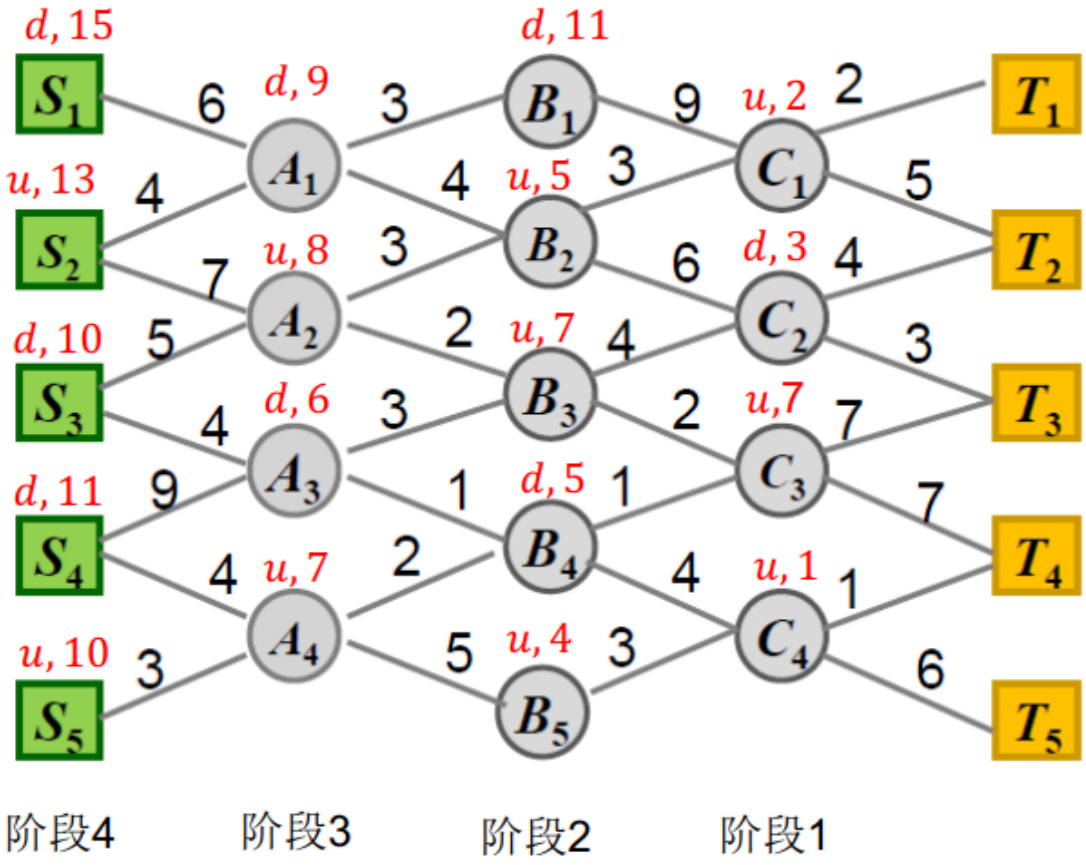


- 输出：一条从起点到终点的最短路

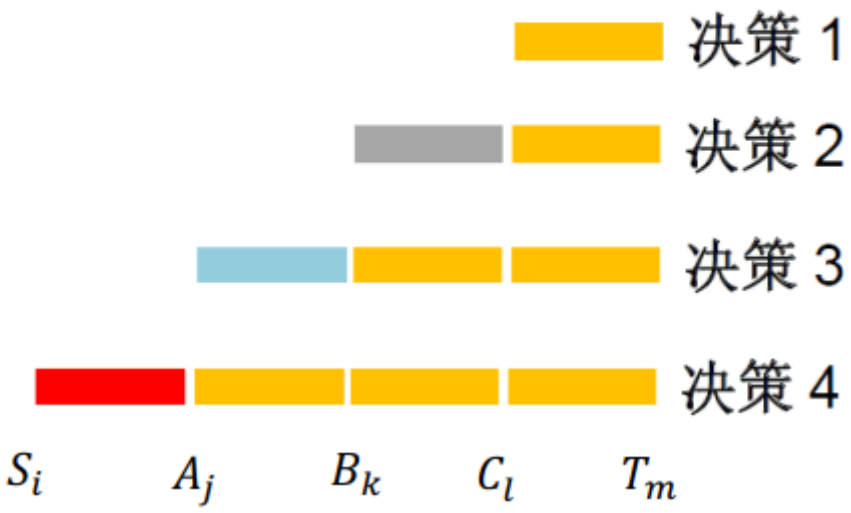
蛮力算法：考察每一条从某个起点到某个终点的路径，计算长度，从其中找出最短路径。

若网络层数未k，那么路径条数近似于2的k次方。

动态规划算法：多阶段决策过程，每步求解的问题都是后面阶段求解问题的子问题，即每步决策依赖于以前的决策结果。



后边界不变，前边界前移



$$F(C_l) = \min_m\{C_l T_m\}$$ 决策1

$$F(B_k) = \min_l\{B_k C_l + F(C_l)\}$$ 决策2

$$F(A_j) = \min_k\{A_j B_k + F(B_k)\}$$ 决策3

$$F(S_j) = \min_j\{S_i A_j + F(A_j)\}$$ 决策4

优化函数值之间存在依赖关系

动态规划算法需要符合优化原则！

优化原则：一个最优决策序列的任何子序列本身一定是相对于子序列的初始和结束状态的最优决策序列。

如果一个问题的最优解包含其子问题的最优解，我们就称此问题具有最优子结构。

- 动态规划(Dynamic Programming)
  - 求解过程是多阶段决策过程，每步处理一个子问题，可用于求解组合优化问题
  - 适用条件：问题要满足优化原则或最优子结构性质，即：一个最优决策序列的任何子序列本身一定是相对于子序列的初始和结束状态的最优决策序列

矩阵链相乘问题 P55页下

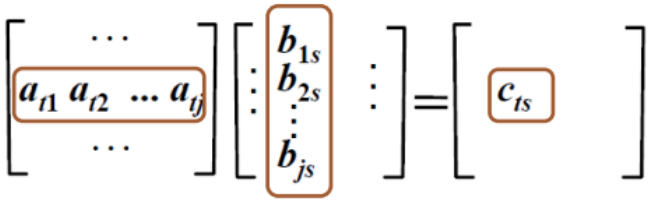
- 问题：设 $A_1, A_2, \ldots, A_n$ 为矩阵序列，A_i 为 $P_{i-1} \times P_i$ 阶矩阵，$i = 1, 2, \ldots, n$. 试确定矩阵的乘法顺序，使得元素相乘的总次数最少.
- 输入：向量

$$P = < P_0, P_1, \cdots, P_n >$$

其中 $P_0, P_1, \ldots, P_n$ 为 n 个矩阵的行数与列数

- 输出：矩阵链乘法加括号的位置.

矩阵A： i 行 j 列，B: j 行 k 列，以元素相乘作基本运算,计算 AB的工作量



$$c_{ts} = a_{t1}\,b_{1s} + a_{t2}\,b_{2s} + \ldots + a_{tj}\,b_{js}$$

AB: i 行 k列，计算每个元素需要做j次乘法，总计乘法次数 为 $i\,j\,k$

蛮力算法

- 加n个括号的方法有$\frac{1}{n+1}\binom{2n}{n}$种，是一个Catalan数，是指数级别

$$W(n) = \Omega\left(\frac{1}{n+1}\frac{(2n)!}{n!n!}\right)$$

带入Stirling公式

$$= \Omega\left(\frac{2^{2n}}{n^{\frac{3}{2}}}\right)$$

卡塔兰数，是组合数学中一个常出现在各种计数问题中出现的数列。输入一个整数n，计算h(n)。其递归式如下：h(n)= h(0)*h(n-1)+h(1)*h(n-2) + ... + h(n-1)h(0) (其中n>=2，h(0) = h(1) = 1) 该递推关系的解为：h(n)=C(2n,n)/(n+1) (n=1,2,3,...)

动态规划算法

- 子问题划分

$A_{i\cdots j}$：矩阵链 $A_i\,A_{i+1}\cdots A_j$，边界i,j

输入向量：$< P_{i-1},P_i,\ldots,P_j >$

其最好划分的运算次数：$m[i,j]$

- 子问题的依赖关系

最优划分最后一次相乘发生在矩阵k的位置，即

$$A_{i\cdots j} = A_{i\cdots k}A_{k+1\cdots j}$$

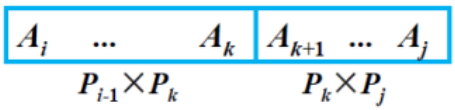
$A_{i\cdots j}$最优运算次数依赖于$A_{i\cdots k}$与$A_{k+1\cdots j}$的最优运算次数

优化函数的递推方程

递推方程：

$m[i,j]$：得到 $A_{i..j}$的最少的相乘次数

$$m[i,j] = \begin{cases} 0, i = j \\ \min\limits_{i\le k<j}\{m[i,k] + m[k+1,j] + P_{i-1}P_kP_j\}, i < j \end{cases}$$



该问题满足优化原则

投资问题 P60页

投资问题的建模

问题：m元钱，n项投资，$f_i(x)$: 将 x 元投入第 i 个项目，求使得总效益最大的投资方案.

建模：

问题的解：$< x_1, x_2, \cdots, x_n >$

目标函数：$\max \{f_1(x_1) + f_2(x_2) + \ldots + f_n(x_n)\}$

约束条件：$x_1 + x_2 + \ldots + x_n = m$，$x_i \in \mathbf{N}$

优化函数的递推方程

$F_k(x)$ 表示 x元钱投给前k个项目的最大效益

多步判断：假设知道 p 元钱$(p \le x)$投给前 $k-1$个项目的最大效益$F_{k-1}(p)$，确定 x元钱投给前 k 个项目的分配方案

递推方程和边界条件

$$F_k(x) = max\{f_k(x_k) + F_{k-1}(x - x_k) | 0 \le x_k < x\}, k > 1$$

$$F_1(x) = f_1(x)$$

$k=2$时实例计算

方案(项目2,其他)： (1,0), (0,1)

$F_2(1) = \max\{f_1(1), f_2(1)\}=11$

x	$f_1(x)$	$f_2(x)$
0	0	0
1	11	0
2	12	5
3	13	10
4	14	15
5	15	20

方案： (2,0), (1,1), (0,2)

$F_2(2) = \max\{f_2(2)+F_1(0), f_2(1)+F_1(1), f_2(0)+F_1(2)\}=12$

方案： (3,0), (2,1), (1,2), (0,3)

$F_2(3) = \max\{f_2(3)+ F_1(0), f_2(2)+F_1(1),$

$f_2(1)+F_1(2), f_2(0)+F_1(3)\}=16$

类似地计算

$F_2(4) = ?$， $F_2(5)= ?$

$$F_k(x) = max\{f_k(x_k) + F_{k-1}(x - x_k) | 0 \le x_k < x\}, k > 1$$

$$F_1(x) = f_1(x)$$

背包问题 P63页

背包问题 (Knapsack Problem)

一个旅行者随身携带一个背包. 可以放入背包的物品有n 种, 每种物品的重量

和价值分别为 w_i, v_i. 如果背包的最大重量限制是b, 每种物品可以放多个. 怎

样选择放入背包的物品以使得背包的价值最大? 不妨设上述 w_i , v_i , b 都是正整数

实例： $n = 4$， $b =10$

$v_1= 1, v_2= 3, v_3= 5, v_4= 9,$

$w_1= 2, w_2= 3, w_3= 4, w_4= 7,$

建模

解是$<x_1, x_2, ..., x_n>$，其中x_i是装入背包的第 i 种物品个数

目标函数： $\max \sum_{i=1}^{n} x_i v_i$

约束条件： $\sum_{i=1}^{n} x_i w_i \leq b, x_i \in N$

线性规划问题: 由线性条件约束的线性函数取最大或最小的问题

整数规划问题: 线性规划问题的变量x_i都是非负整数

优化函数的递推方程

$F_k(y)$: 装前k 种物品, 总重不超过y, 背包达到的最大价值

$$F_k(y) = \max\{F_{k-1}(y), F_k(y - w_k) + v_k\}$$

$$F_0(y) = 0,\ 0 \leq y \leq b, F_k(0) = 0, 0 \leq k \leq n$$

$$F_1(y) = \left\lfloor \frac{y}{w_1} \right\rfloor v_1,\ F_k(y) = -\infty\ y < 0$$

类似于投资问题,如何写递推方程这两种写法有什么区别

$$F_k(y) = \max\left\{F_{k-1}(y - x_k w_k) + x_k v_k | 0 \leq x_k \leq \left\lfloor \frac{y}{w_k} \right\rfloor\right\}$$