动态规划

与分治法的异同点

相同点

通过合并子问题来解决整个问题

不同点

分治法：将问题划分为独立的子问题，递归子问题然后合并

动态规划：当分解子问题，但他们共享子问题时，可以采用动态规划

步骤

- 描述最优解的结构特征
- 递归定义一个最优值的解
- 自底向上计算一个最优值的解
- 从已计算的信息中构造一个最优解

实例

适合用动态规划求解的问题

- 最优子结构
- 重叠子问题

矩阵链乘

递归分析

设$m[i,j]$是计算$A_{i...j}$ 所需乘法的最小次数（最优解的值），则$A1_{1...n}$ 的最小计算成本是$m[1,n]$。

- $i=j$时链上只有一个A_i，无需乘法
- 当$i \neq j$时，递推公式如下：

$$m[i,j]=\begin{cases} 0 & i=j \\ \min_{i \le k < j}\{m[i,k]+m[k+1][j]+p_{i-1}p_kp_j\} & \end{cases}$$

其中p_{i-1}指的是A_i的行，p_k是A_k的列，p_j是A_j的列。

禁止商用

$p_{i-1}p_kp_j$是以A_k的列数为列数，以A_i的行数为行数的矩阵和以以A_j的列数为列数，以A_{k+1}的行数为行数的矩阵相乘需要的次数，其中A_k的行数和A_{k+1}的列数相等

注意计算顺序，从$i-j=0$到$i-j=n$

时间复杂度为$T(n)=O(n^3)$

最长公共子序列

子序列

B是A的子列，若B是A中删去某些元素（亦可不删）所得的序列。即B不一定是由A的连续元素构成的子列。

定义

给定序列 $X=<x_1,x_2,\dots,x_m>$，序列$Z=<z_1,z_2,\dots,z_k>$是X的一个子序列须满足：若X的索引中存在一个严格增的序列 $i_1<i_2<\dots<i_k$使得对所有的$j(1\le j\le k)$，均有$x_{i_j}=z_j$成立。

n两个序列的公共子序列（CS)

若Z是X的子列，又是Y的子列，则Z是序列X和Y的公共子序列

n最长公共子序列（LCS)

X和Y的公共子序列中长度最大者

步骤

- 刻化LCS结构特征
- 子问题的递归解
- 计算最优解
- 构造一个LCS

最优子序列结构

设$X=<x_1,x_2,\dots,x_m>$和$Y=<y_1,y_2,\dots,y_n>$是序列，$Z=<z_1,z_2,\dots,z_k>$是X和Y的任意一个LCS

- 若$x_m=y_n$，则$z_k=x_m=y_n$且Z_{k-1}是X_{m-1}和Y_{n-1}的一个LCS
- 若$x_m\neq y_n$，且$z_k\neq x_m$，则Z是X_{m-1}和Y的一个LCS

禁止商用

- 若$x_m \neq y_n$，且$z_k \neq y_n$，则Z是X和Y_{n-1}的一个LCS

递归结构

- 如果$x_m = y_n$则需要解决一个子问题：找到X_{m-1}和Y_{n-1}的一个LCS
- 如果$x_m \neq y_n$则需要解决两个个子问题：
  - 找X_{m-1}和Y的一个LSC
  - 找X和Y_{n-1}的一个LSC

递推式如下

$$c[i,j] = \begin{cases} 0 & i = 0\text{或者}j = 0 \\ C[i-1,j-1]+1 & i,j>0, x_i = y_j \\ \max\{C[i,j-1], C[i-1,j]\} & i,j>0, x_i \neq y_j \end{cases}$$

计算顺序，已知$C[0,0]$~$C[0,n]$，$C[0,0]$~$C[m,0]$均等于0，接着计算$C[1,1]$~$C[1,n]$以此类推。最后计算出的$C[m,n]$为最后答案

禁止商用