# 电子科技大学

# 数据结构与算法

# 习
# 题
# 册

姓名：_____

学号：_____

专业：□ 计算机大类

□ 网络空间安全

□ 物联网+实验班

□ 英才实验班

仅限 2020 年秋季使用
P0800335.06、P2700530.01、P0800335.01、H0816130.01

周益民 博士 教授
2020 年 9 月 7 日

# 算法分析

## 1. 简答题

(1) 描述递归方法及其特点

(2) 描述分治方法及其特点

(3) 描述动态规划法及其特点

## 2. 递归方程证明

(1) 用数学归纳法证明递归关系式$T(n)=2T(n/2)+1$的渐近解为$T(n)=O(n)$。

(2) 用数学归纳法证明递归式$T\left(n\right)=9T\left(n/3\right)+n^2$的渐近界为$T\left(n\right)=O\left(n^2\log n\right)$。

(3) 直接展开法求解递归式$T\left(n\right)=4T\left(n/2\right)+n^2$的渐近界。

## 3. 主方法求解递归方程

主方法是一类求解递归方程的快速便捷方法。它适用于求下面类型的递归形式：

$$T(n)=a\cdot T(\frac{n}{b})+f(n) \tag{1}$$

其中 $a\geq 0,b\geq 0$， $f(n)$ 为渐进函数。

求解方法是比较渐进函数 $f(n)$ 与 $a,b$ 的指数形式 $n^{\log_b a}$ 。分三种情况进行讨论。

**第一种情况，**如果 $f(n)$ 可以被写作

$$f(n)=n^{\log_b a-\varepsilon}=n^{\log_b a}\cdot\frac{1}{n^{\varepsilon}} \tag{2}$$

其中 $\varepsilon\geq 0$ 。这说明 $f(n)$ 的增长速度比 $n^{\log_b a}$ 要慢，且慢 $n^{\varepsilon}$ 倍。则可以直接解得 $T(n)$ 的时间复杂度为

$$T(n)=\Theta(n^{\log_b a}) \tag{3}$$

**第二种情况，**如果 $f(n)$ 可以被写作

$$f(n)=n^{\log_b a+\varepsilon}=n^{\log_b a}\cdot n^{\varepsilon} \tag{4}$$

其中 $\varepsilon\geq 0$ 。这说明 $f(n)$ 的增长速度比 $n^{\log_b a}$ 要快，且快 $n^{\varepsilon}$ 倍。则可以直接解得 $T(n)$ 的时间复杂度为

$$T(n)=\Theta(f(n)) \tag{5}$$

**第三种情况，**如果 $f(n)$ 可以被写作

$$f(n)=C\cdot n^{\log_b a} \tag{6}$$

其中 $C=(\log n)^k,k\geq 0$ 。这说明 $f(n)$ 的增长速度与 $n^{\log_b a}$ 相同。则可以直接解得 $T(n)$ 的时间复杂度为

$$T(n)=\Theta((\log n)^{k+1}\cdot f(n)) \tag{7}$$

**【例 1】求解下列递归关系式。**

$$g(n)=4\cdot g(n/3)+2\cdot n$$

【解】根据主方法，$a=4$，$b=3$，$f(n)=2\cdot n$。
那么，将 $f(n)$ 可以写为

$$f(n)=2\cdot n=2\cdot n^{\log_3 4-\varepsilon}\mid \varepsilon=\log_3 4-1>0$$

则解得 $g(n)$ 的时间复杂度为

$$g(n)=\Theta(n^{\log_3 4})$$

**【例 2】求解下列递归关系式。**

$$f(n)=3\cdot f(n/2)+n^2$$

【解】根据主方法，$a=3$，$b=2$，$g(n)=n^2$。
那么，将 $g(n)$ 可以写为

$$g(n)=n^2=n^{\log_2 3+\varepsilon}\mid \varepsilon=2-\log_2 3$$

则解得 $f(n)$ 的时间复杂度为

$$f(n)=\Theta(g(n))=\Theta(n^2)$$

**【例 3】求解下列递归关系式。**

$$T(n)=2\cdot T(n/2)+n$$

【解】根据主方法，$a=2$，$b=2$，$f(n)=n$。
那么，将 $f(n)$ 可以写为

$$f(n)=n=(\log n)^0\cdot n^{\log_2 2}$$

则解得 $T(n)$ 的时间复杂度为

$$T(n)=\Theta((\log n)^{0+1}\cdot f(n))=\Theta(n\cdot\log n)$$

## 4. 动态规划求解

动态规划所处理的问题是一个多阶段决策问题。动态规划求解过程就可以用一个最优决策表来描述，最优决策表是一个二维表，其中行表示决策的阶段，列表示问题状态，表格需要填写的数据一般对应此问题的在某个阶段某个状态下的最优值，填表的过程就是根据递推关系，从 1 行 1 列开始，以行或者列优先的顺序，依次填写表格，最后根据整个表格的数据通过简单的取舍或者运算求得问题的最优解。

**【例 1】计算矩阵连乘积。给定 $n=4$ 个矩阵 $\{A,B,C,D\}$。要求以最小的计算量完成这 $n$ 个矩阵的连乘积 $S=A_{35,40}\times B_{40,20}\times C_{20,10}\times D_{10,15}$。**

【解】建立矩阵连乘下标为

$$S=M_1\times M_2\times M_3\times M_4$$

令 $m_{i,j}$ 代表矩阵 $M_i\times\cdots\times M_j$ 之间的乘法的最小次数，那么建立动态规划递归方程为

$$m_{i,j}=\begin{cases}0 & i=j\\ \min\limits_{i\le k<j}\{m_{i,k}+m_{(k+1),j}+r_i\cdot r_{k+1}\cdot r_{j+1}\} & i<j\end{cases}$$

$$\begin{aligned}
m_{1,2}&=35\times 40\times 20=28000\\
m_{2,3}&=40\times 20\times 10=8000\\
m_{3,4}&=20\times 10\times 15=3000\\
m_{1,3}&=\min\{m_{1,2}+m_{3,3}+35\times 20\times 10,\\
&\qquad m_{1,1}+m_{2,3}+35\times 40\times 10\}\\
&=22000\\
m_{2,4}&=\min\{m_{2,3}+m_{4,4}+40\times 10\times 15,\\
&\qquad m_{2,2}+m_{3,4}+40\times 20\times 15\}\\
&=14000\\
m_{1,4}&=\min\{m_{1,1}+m_{2,4}+35\times 40\times 15,\\
&\qquad m_{1,2}+m_{3,4}+35\times 20\times 15,\\
&\qquad m_{1,3}+m_{4,4}+35\times 10\times 15\}\\
&=\min\{14000+21000,28000+3000+10500,22000+5250\}\\
&=\min\{35000,41500,27250\}=27250
\end{aligned}$$

**【例 2】0/1 背包问题。给定 $N$ 中物品和一个背包。物品 $i$ 的重量是 $w_i$,其价值 $v_i$，背包的容量为 C。问应该如何选择装入背包的物品，使得转入背包的物品的总价值为最大？**

【解】0/1 背包的状态转换方程为

$$f(i,j)=\max\{f(i-1,j-w_i)+v_i,f(i-1,j)\}$$

其中，$f(i,j)$ 表示在前 $i$ 件物品中选择若干件放在承重为 $j$ 的背包中，可以取得的最大价值。隐含的意思是，若要选择 $i$ 号物品的价值 $v_i$ 被加入，则在 $(j-w_i)$ 容量时候其最优解必被包括。这非常类似最短路径问题中，如果最短路径存在，则原点到中继各点的路径皆是最短路径。

**【实例】有编号分别为 a，b，c，d，e 的五件物品；它们的重量分别是 2，2，6，5，4；它们的价值分别是 6，3，5，4，6。现在给你个承重为 10 的背包，如何让背包里装入的物品具有最大的价值总和。**

【解】先考虑这个物品放入的时候可以获得的最大价值，这个物品的价值+剩余背包(背包总容量-该物品重量)容量可以获得的最大价值，再考虑不放入这个物品的时候可以获得的最大价值（剩余背包容量，此时就是背包总重量，可以获得的最大价值），然后将两者进行比较，那种结果的价值大就将哪种结果的价值保存下来，依次类推。

第一种解，自上而下求解，按行填表

| | $w_i$ | $v_i$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 6 | 0 | 0 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 |
| 2 | 2 | 3 | 0 | 0 | 6 | 6 | 9 | 9 | 9 | 9 | 9 | 9 | 9 |
| 3 | 6 | 5 | 0 | 0 | 6 | 6 | 9 | 9 | 9 | 9 | 11 | 11 | 14 |
| 4 | 5 | 4 | 0 | 0 | 6 | 6 | 9 | 9 | 9 | 10 | 11 | 13 | 14 |
| 5 | 4 | 6 | 0 | 0 | 6 | 6 | 9 | 9 | 12 | 12 | 15 | 15 | 15 |

第二种解，自下而上求解，按行填表

| | $w_i$ | $v_i$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 6 | 0 | 0 | 6 | 6 | 9 | 9 | 12 | 12 | 15 | 15 | 15 |
| 2 | 2 | 3 | 0 | 0 | 3 | 3 | 6 | 6 | 9 | 9 | 9 | 10 | 11 |
| 3 | 6 | 5 | 0 | 0 | 0 | 0 | 6 | 6 | 6 | 6 | 6 | 10 | 11 |
| 4 | 5 | 4 | 0 | 0 | 0 | 0 | 6 | 6 | 6 | 6 | 6 | 10 | 10 |
| 5 | 4 | 6 | 0 | 0 | 0 | 0 | 6 | 6 | 6 | 6 | 6 | 6 | 6 |

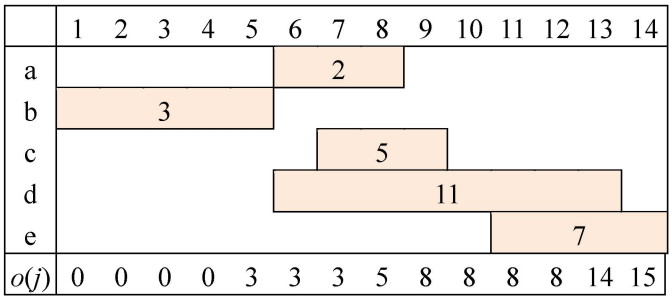
**【例 3】旅行计划规划。某同学想利用悠长寒假中的连续 14 天安排一系列激动人心旅行计划。他拟出的旅行活动目的地有五个$\{a,b,c,d,e\}$，以 14 天为单位，五项活动的启动日期分别是$S=\{6,1,7,6,11\}$，结束日期分别是$T=\{8,5,9,13,14\}$，将获得经验值$V=\{2,3,5,11,7\}$。请安排其旅行计划，以使得获得经验值最多。要求写出动态规划的递归关系式，并求解其最优解。**

【解】定义$f(j)$表示与日期$j$不冲突的活动最晚结束日期$i$，即$i<j$。定义$o(j)$表示在日期$j$时能够获取的最大经验值，即最优解。

分两种情况讨论：第一，如果$o(j)$选择日期$j$结束的活动，则不能选取不兼容的活动$\{f(j)+1,f(j)+2,\cdots,j-1\}$，且必须包括剩余的$1,2,\cdots,f(j)$个活动所构成最优解。第二，如果$o(j)$不选择日期$j$结束的活动，则必定包括$1,2,\cdots,j-1$所形成的最优解。那么递归关系式为

$$o(j)=\begin{cases}0 & j=0\\ \max\{o(j-1),o(f(i))+v_j\} & j>0\end{cases}$$

求解可得最优解$\{b,c,e\}$，最优值 15。




## 5. 分治策略设计

(1) 给定数组 int array[n]，设计一个算法，在最坏情况下用$n+\log n-2$次比较找出 array 中元素的最大数和次大数。

【解】

① 将 array[n]分为 2 组，下标分别是 0..k-1 和 k..n-1，其中 k＝n/2。作 k 次比较：array[i]和 array[i+k]，其中 0≤i≤k-1。当 array[i]>array[i+k]时，交换它们的位置。那么经过 k 次比较后，有 array[i]≤array[i+k]。

② 递归地在 array 下标 k..n-1 中进行如①的比较、交换操作，直至 k 降至 0。

③ 容易看出，array[n-1]即为 array 中的最大教。而 array 中的次大数只能是 array[n-2]或 array[n-3]。再用 1 次比较即可求得 array 中的次大数。

(2) 给定数组 int array[n]，n 为偶数。设计一个算法，在最坏情况下用 $3n/2-2$ 次比较找出 array 中元素的最大值和最小值。

【解】

① 将 array 中的数据元素两两分成一组，即为 array[0]和 array[1], array[2]和 array[3],.., array[n-2]和 array[n-1]共计 n/2 组，每组中的两个元素进行比较，得到最大值和最小值，共计 n/2 次比较；

② 最大值组成逻辑数组 array[1],array[3],array[5],.., array[n-1]，采用冒泡排序(或其他排序方法)求得这 n/2 个元素中的最大值，即为原数组 array 的最大值，需 n/2-1 次比较；

③ 最小值组成逻辑数组 array[0],array[2],array[4],.., array[n-2]，采用冒泡排序(或其他排序方法)求得这 n/2 个元素中的最小值，即为原数组 array 的最小值，需 n/2-1 次比较；

至此，共需 $3n/2-2$ 次比较

编写者的话：此数据结构习题册共计 52 页系周益民选编，前后增删已历九年，旨在普适性地选择最典型的题型例子给予本科二年级学生用作日常练习、复习。周益民教学班所属学员当使用此习题册无虞。恳请选课同学，若发现有字词、语句、公式、数字、参考答案等任何疏漏，及时反馈给周益民以作修订为要，yiminzhou@uestc.edu.cn 。非常期待您的宝贵意见！