数据结构与算法

Data Structure and Algorithm

VI. 分治法

授课人：Kevin Feng

翻译　　：孙　　兴

前期回顾

- 数据结构和算法
- 数学回顾
- 数组和数组列表
- 递归
- 排序和查找

排序和搜索

- 排序
- 二分搜索
- 分治法
- 双指针
- 滑动窗口
- 其他方法
- 贪婪算法
- 动态规划

分治法

- 将问题拆分为几个子问题，而且这些子问题和原问题相似只是量级上小一些。
- 递归地解决每一个子问题，然后结合这些子问题的解决方案构造出原问题的解决方案。
- 我们已经遇到过的问题：
  - 二分搜索
  - 归并排序
  - 快速排序

分治法—习题

- 习题1：快速指数
  - 计算 a^n

- 习题2：搜索峰值
  - 数组没有重复值，但可能存在多个峰值，返回任意一个峰值的index.
  - You may imagine that num[-1] = num[n] = -∞.

- 习题3：查找中值/查找第K个元素

分治法—习题

- 习题4：计算逆序对
  - 对数组做逆序对计数—距离数组的排序结果还有“多远”。如果一个数组已经排好序（升序），那么逆序对个数为0；如果数组是降序排列的，则逆序对个数最多。
  - 在形式上，如果有两个元素a[i]，a[j]，如果a[i] > a[j] 且 i < j，那么a[i]，a[j]构成一个逆序对。
  - 例如序列2，4，1，3，5 有三个逆序对，分别是(2，1)，(4，1)，(4，3)
- 习题5：在已排序数组中找到多余元素的索引
  - 给定两个排好序的数组。这两个数组只有一个不同的地方：在第一个数组某个位置上多一个元素。请找到这个元素的索引位置。

加和值最大的子序列问题

- 在一个一维数组中找到连续的子序列，且这个子序列的加和值最大。
- 例如，一位数组序列为 –2， 1， –3， 4， –1， 2， 1， –5， 4
- 则这个序列对应的加和值最大的子序列为4， –1， 2， 1， 其加和值为6.

加和值最大的子序列问题--解法1

Brute force solution

- Compute the sum of every subarray and pick the maximum
  Try every pair of indices i,j with $1 \le i \le j \le n$, and for each one compute $s(i,j) = \sum_{k=i}^{j} A[k]$

- Time complexity $\Theta(n^3)$

- With a little more care, can improve to $\Theta(n^2)$:
  can compute the sums of all the subarrays in time $\Theta(n^2)$.

Algorithm 0

Brute force

```
1) ans = 0
2) for i = 1 to n do
3)     for j = i to n do
4)         sum = 0
5)         for k = i to j do
6)             sum += a_k
7)         if sum > ans then ans = sum
8) return ans
```

Analysis

Lines 5-6: $c(j-i+1) \le cn$ for some constant c. c depends on compiler/ machine

Lines 3-7: $\le n(cn+c')$ for some constant c' depending upon compiler or machine

$= n \cdot O(n) = O(n^2)$

Total time: $n \cdot O(n^2) = O(n^3)$

In fact, is $\Theta(n^3)$

e.g. $n = 10^6$, $n^3 = 10^{18}$, 10^9 ops/sec $\Rightarrow$ 10^9sec ~ 30 years

Algorithm 1

Idea - don't recompute sum from scratch

```
ans = 0
for i = 1 to n do
    sum = 0
    for j = i to n do
        sum += a_j
        if sum > ans then ans = sum
return ans
```

Analysis

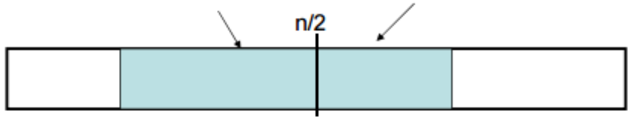
$O(n^2)$

加和值最大的子序列问题--解法2

Divide and Conquer

- A subarray A[i*…j*] with maximum sum is
  - Either contained entirely in the first half , i.e. j* ≤ n/2
  - Or contained entirely in the right half , i.e. i* ≥ n/2
  - Or overlaps both halfs: i* ≤ n/2 ≤ j*

- We can compute the best subarray of the first two types with recursive calls on the left and right half.
- The best subarray of the third type consists of the best subarray that ends at n/2 and the best subarray that starts at n/2. We can compute these in O(n) time.



Algorithm 2

Idea - divide and conquer

```
solve(a_1, ..., a_n)
if n = 1 then return max(a_1, 0)
ans = max{solve(a_1, ..., a_⌊n/2⌋), solve(a_⌊n/2⌋+1, ..., a_n)}
ansl = sum = 0
for i = ⌊n/2⌋ to 1 do
    sum = sum + a_i
    if sum > ansl then ansl = sum
ansr = sum = 0
for i = ⌊n/2⌋ + 1 to n do
    sum = sum + a_i
    if sum > ansr then ansr = sum
return max(ans, ansl + ansr)
```

Analysis

$$T(n) = \begin{cases} O(1) & \text{if } n = 1 \\ 2T\left(\frac{n}{2}\right) + O(n) & \text{otherwise} \end{cases}$$

$O(n \log n)$

Algorithm 3

Dynamic programming

Let b_j be the maximum sum over all blocks that end at j

$b_j = \max\left(b_{j-1} + a_j, a_j\right)$

```
ANS = 0
b = 0
FOR j = 1 .. n
     b = max{b + a_j, a_j}
     ANS = max{b, ANS}
}
RETURN ANS
```

$O(n)$ runtime

Any algorithm must take $\Omega(n)$, so it is $\Theta(n)$

分治法应用

- 习题7：VLSI芯片测试
  - Diogenes教授有n个相同的VLSI芯片，理论上这些芯片可以相互测试。教授的测试夹具可以同时容纳2个芯片。每次装载好夹具，则两个芯片可以互相检测，并且对另一个芯片是好是坏给出报告。一个好的芯片总能准确报告出另一个芯片是好是坏，但是一个坏的芯片的报告结果不可信。因此，我们可以得到如下图所示的四种可能结果：

Chip A says	Chip B Says	Conclusion
B is good	A is good	both are good, or both are bad
B is good	A is bad	at least one is bad
B is bad	A is good	at least one is bad
B is bad	A is bad	at least one is bad

a) Show that if more than $n/2$ chips are bad, the professor cannot necessarily determine which chips are good using any strategy based on this kind of pairwise test. Assume that the bad chips can conspire to fool the professor.
b) Consider the problem of finding a single good chip from among n chips, assuming that more than $n/2$ of the chips are good. Show that $\lfloor n/2 \rfloor$ pairwise tests are sufficient to reduce the problem to one of nearly half the size.
c) Show that the good chips can be identified with $\Theta(n)$ (proportional to n) pairwise tests, assuming that more than $n/2$ of the chips are good. Give and solve the recurrence that describes the number of tests.

分治法—问题

- 习题8：　快速整数乘法
- 习题9：　对于多项式乘法的快速傅里叶变换
- 习题10：矩阵乘法
- 习题11：最近数对
- 习题12：水槽问题
- 习题13：奇-偶数 换序问题
- 习题14：用最少步数收集所有硬币
- 习题15：拼接问题

水槽问题

⊙ 给定一个容量为C升的水槽，初始时给这个水槽装满水。每天都会给水槽中加入l升的水，若有溢出，则舍弃多余的水。另外在第i天，水槽中有i升的水会被喝掉。请计算在哪一天水槽中的水第一次用完。

⊙ 举例：

- 输入：容量C=5，l=2。
- 输出：4
- 在第一天开始时，水槽中有5升水；第一天结束时有（5-1=）4升水。
- 在第二天开始时，水槽中有4+2=6升水，但是容量为5，所以有5升水。在第二天结束时，有5-2=3升水。
- 在第三天开始时，水槽中有3+2=5升水，在第三天结束时，有5-3=2升水。
- 在第四天开始时，水槽中有2+2=4升水，在第四天结束时，有4-4=0升水。
- 所以最后结果为4.

奇-偶数换序问题

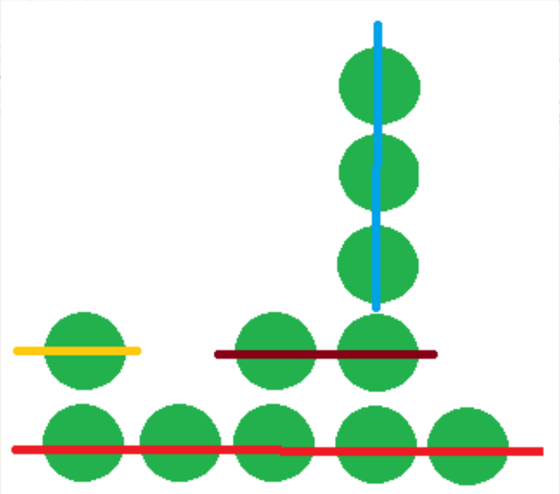
- 给定一个含有2n个元素的数组，形式如：｛ a1， a2， a3， a4， …．．， an， b1， b2， b3， b4， …．， bn ｝.

- 举例：
  - 输入： arr[] = { 1， 2， 9， 15 }
  - 输出：1 9 2 15
  - 输入： arr[] = { 1， 2， 3， 4， 5， 6 }
  - 输出： 1 4 2 5 3 6

用最少步数收集所有硬币

- 给定几摞硬币，这些摞列的硬币相邻排列。我们要用最少的步数收集所有的硬币，其中每一步可以**沿水平线**或者**垂直线连续** 收集。

- 举例：
  - 输入： height[] = [2 1 2 5 1]
  - 数组中每个值代表对应摞的高度（即硬币个数），此处给了我们5摞硬币。其中第一摞有2个硬币，第二摞有1个硬币，其余依次对应。
  - 输出：4



拼接问题

- 给定一个n*n的棋盘，这里n为2^k （k>=1, 即n是2的指数，且n最小为2）。这个棋盘缺失一个网格（大小为1*1）。请用L形的贴图填充此棋盘，一个L形的贴图是一个的2*2方形去掉一个1*1的部分构成的。

数据结构与算法

Data Structure and Algorithm

VI. 分治法

结束

授课人：Kevin Feng