

后,递归地寻找数组 B 中的候选元;它也是 A 的候选元(为什么)。

a. 递归如何终止?

*b. 当 N 是奇数时的情形如何处理?

*c. 该算法的运行时间是多少?

d. 我们如何避免使用附加数组 B ?

*e. 编写一个程序求解主元素。

2.27 输入是一个 $N \times N$ 数字矩阵并且已经读入内存。每一行均从左到右递增。每一列则从上到下递增。给出一个 $O(N)$ 最坏情形算法以决定数 X 是否在该矩阵中。

2.28 使用正数的数组 a 设计有效的算法以确定:

a. $a[j] + a[i]$ 的最大值, 其中 $j \geq i$ 。

b. $a[j] - a[i]$ 的最大值, 其中 $j \geq i$ 。

c. $a[j] * a[i]$ 的最大值, 其中 $j \geq i$ 。

d. $a[j] / a[i]$ 的最大值, 其中 $j \geq i$ 。

*2.29 在我们的计算机模型中为什么假设整数具有固定长度是重要的?

2.30 考虑第1章中描述的字谜游戏。假设我们固定最长单词的大小为10个字符。

a. 设 R 、 C 和 W 分别表示字谜游戏中的行数、列数和单词个数, 那么在第1章所描述的那些算法用 R 、 C 和 W 表示的运行时间是多少?

b. 设单词表是预先排序过的。指出如何使用折半查找得到一个具有少得多的运行时间的算法。

2.31 设在折半查找程序的第15行的语句是 $low = mid$ 而不是 $low = mid + 1$ 。这个程序还能正确运行吗?

2.32 实现折半查找使得在每次迭代中只执行一次二路比较。(课文中的实现使用三路比较。假设只有方法 `lessThan` 是可用的。)

2.33 设算法3(见图2-7)的第15行和第16行由

```
15      int maxLeftSum = maxSubSum( a, left, center - 1 );  
16      int maxRightSum = maxSubSum( a, center, right );
```

代替, 这个程序还能正确运行吗?

*2.34 三次的最大子序列和算法的内循环执行 $N(N+1)(N+2)/6$ 次最内层代码的迭代。相应的二次算法执行 $N(N+1)/2$ 次迭代。而线性算法执行 N 次迭代。哪种模式是显然的? 你能给出这种现象的组合学解释吗?

参考文献

算法的运行时间分析最初因 Knuth 在其三卷本丛书[5], [6]和[7]中使用而流行。gcd 算法的分析出现在[6]中。这方面的另一本早期著作参见[1]。

大 O 、大 Ω 、大 Θ , 以及小 o 记号由 Knuth 在[8]中提倡。但是对于这些记号尚无统一的规定, 特别是在使用 $\Theta()$ 时。许多人更愿意使用 $O()$, 虽然它表达的精度要差得多。此外, 当需要用到 $\Omega()$ 时, 迫不得已还用 $O()$ 表示下界。

最大子序列和问题出自[3]。丛书[2]、[3]和[4]指出如何优化程序以求得运行速度的提高。

1. A. V. Aho, J. E. Hopcroft, and J. D. Ullman, *The Design and Analysis of Computer Algorithms*, Addison-Wesley, Reading, Mass., 1974.