

\*c. 求最大子序列乘积。

- 2.18 数值分析中一个重要的问题是对某个任意的函数  $f$  找出方程  $f(X)=0$  的一个解。如果该函数是连续的并有两个点  $low$  和  $high$  使得  $f(low)$  和  $f(high)$  符号相反, 那么在  $low$  和  $high$  之间必然存在一个根, 并且这个根可以通过折半查找求得。写出一个函数, 以  $f$ 、 $low$  和  $high$  为参数, 并且解出一个零点。(为了实现一个泛型函数作为参数, 我们传递一个函数对象, 让该对象实现 Function 接口, 而这个 Function 接口含有一个方法  $f$ ) 为保证能够终止, 你必须要做什么?
- 2.19 课文中最大相连子序列和算法均不给出具体序列的任何指示。将这些算法修改使得它们以单个对象的形式返回最大子序列的值以及具体序列的那些相应下标。
- 2.20 a. 编写一个程序来确定正整数  $N$  是否是素数。  
b. 你的程序在最坏情形下的运行时间是多少(用  $N$  表示)? (你应该能够以  $O(\sqrt{N})$  来完成这项工作)  
c. 令  $B$  等于  $N$  的二进制表示法中的位数。 $B$  的值是多少?  
d. 你的程序在最坏情形下的运行时间是什么(用  $B$  表示)?  
e. 比较确定一个 20(二进制)位的数是否是素数和确定一个 40(二进制)位的数是否是素数的运行时间。  
f. 用  $N$  或  $B$  给出运行时间更合理吗? 为什么?
- \*2.21 厄拉多塞(Eratosthenes)筛是一种用于计算小于  $N$  的所有素数的方法。我们从制作整数 2 到  $N$  的表开始。找出最小的未被删除的整数  $i$ , 打印  $i$ , 然后删除  $i, 2i, 3i, \dots$ 。当  $i > \sqrt{N}$  时, 算法终止。该算法的运行时间是多少?
- 2.22 证明  $X^{62}$  可以只用 8 次乘法算出。
- 2.23 不用递归, 写出快速求幂的程序。
- 2.24 给出用于快速取幂运算中的乘法次数的精确计数。(提示: 考虑  $N$  的二进制表示)
- 2.25 程序 A 和 B 经分析发现其最坏情形运行时间分别不大于  $150N \log_2 N$  和  $N^2$ 。如果可能, 请回答下列问题:  
a. 对于  $N$  的大值( $N > 10\,000$ ), 哪一个程序的运行时间有更好的保障?  
b. 对于  $N$  的小值( $N < 100$ ), 哪一个程序的运行时间有更好的保障?  
c. 对于  $N = 1\,000$ , 哪一个程序平均运行得更快?  
d. 对于所有可能的输入, 程序 B 是否总能够比程序 A 运行得更快?
- 2.26 大小为  $N$  的数组 A, 其主元素是一个出现超过  $N/2$  次的元素(从而这样的元素最多有一个)。例如, 数组

3, 3, 4, 2, 4, 4, 2, 4, 4

有一个主元素 4, 而数组

3, 3, 4, 2, 4, 4, 2, 4

没有主元素。如果没有主元素, 那么你的程序应该指出来。下面是求解该问题的一个算法的概要:

首先, 找出主元素的一个候选元(这是困难的部分)。这个候选元是唯一有可能是主元素的元素。第二步确定是否该候选元实际上就是主元素。这正好是对数组的顺序搜索。为找出数组 A 的一个候选元, 构造第二个数组 B。比较  $A_1$  和  $A_2$ 。如果它们相等, 则取其中之一加到数组 B 中; 否则什么也不做。然后比较  $A_3$  和  $A_4$ , 同样, 如果它们相等, 则取其中之一加到 B 中; 否则什么也不做。以该方式继续下去直到读完整个的数组。然