运行程序(3)。

d. 将实际的运行时间与你的分析进行比较。

e. 每个算法的最坏情形的运行时间是什么?

2.9 用运行时间的估计值完成图 2-2 中的表，这些时间太长无法模拟。插入上述三个算法的运行时间并估计计算 100 万个数的最大子序列和所需要的时间。你得出哪些假设?

2.10 对于手工进行计算所使用的典型算法，确定下列计算的运行时间：

a. 将两个 N 位数字的整数相加。

b. 将两个 N 位数字的整数相乘。

c. 将两个 N 位数字的整数相除。

2.11 一个算法对于大小为 100 的输入花费 0.5 ms。如果运行时间如下，则解决输入量大小为 500 的问题需要花费多长的时间(设低阶项可以忽略)：

a. 是线性的

b. 为 $O(N \log N)$

c. 是二次的

d. 是三次的

2.12 一个算法对于大小为 100 的输入花费 0.5 ms。如果运行时间如下，则用 1 分钟可以解决多大的问题(设低阶项可以忽略)：

a. 是线性的

b. 为 $O(N \log N)$

c. 是二次的

d. 是三次的

2.13 计算 $f(x) = \sum_{i=0}^{N} a_i x^i$ 需要多少时间?

a. 用简单的例程执行取幂运算。

b. 使用 2.4.4 节的例程计算。

2.14 考虑下述算法(称为 Horner 法则)计算 $f(X) = \sum_{i=0}^{N} a_i x^i$ 的值：

```
poly = 0;
for( i = n; i >= 0; i-- )
    poly = x * poly + a[i];
```

a. 对 $x=3$，$f(x)=4x^4+8x^3+x+2$ 指出该算法的各步是如何进行的。

b. 解释该算法为什么能够解决这个问题。

c. 该算法的运行时间是多少?

2.15 给出一个有效的算法来确定在整数 $A_1<A_2<A_3<\cdots<A_N$ 的数组中是否存在整数 i 使得 $A_i=i$。你的算法的运行时间是多少?

2.16 基于下列各式编写另外的 *gcd* 算法(其中 $a>b$)

- $gcd(a,b)=2gcd(a/2,b/2)$ 若 a 和 b 均为偶数。
- $gcd(a,b)=gcd(a/2,b)$ 若 a 为偶数，b 为奇数。
- $gcd(a,b)=gcd(a,b/2)$ 若 a 为奇数，b 为偶数。
- $gcd(a,b)=gcd((a+b)/2,(a-b)/2)$ 若 a 和 b 均为奇数。

2.17 给出有效的算法(及其运行时间分析)：

a. 求最小子序列和。

*b. 求最小的正子序列和。