第1章 习题

算法正确性

1.1 欧几里德算法，描述：

GCD(a, b):

1. $r = a \% b$
2. while $r \neq 0$, Do
   $a \leftarrow b$
   $b \leftarrow r$
   $r \leftarrow a \% b$
3. return b.

假设 $a > b$，$b = \min(a, b)$.

$r_1 = a\%b \leq \min(a-b, b) \leq \frac{a}{2}$

$r_2 = b\%(a\%b) \leq \frac{b}{2}$

迭代2k次之后.

$r_{2k} \leq b/2^k$. $k = \log_2 b$.

求余次数：2k

赋值次数：4k，其中 $k = \log_2(\min(a, b))$.

渐近复杂度为 $O(\log_2(\min(a, b)))$.

1.2 插入排序算法描述

Insert Sort

输入：$A[1:n]$

输出：排序后的数组 $A[1:n]$.

```
1. For i ← 2 To n Do
2.     key ← A[i]
3.     j ← i-1
4.     While j>0 且 A[j]>key Do
5.         A[j+1] ← A[j];
6.         j ← j-1
7.     A[j+1] ← key
```

(a1) 内层循环：j从i-1开始，减1，最多到 j=0 结束

外层循环：i从2开始，增1，到 i=n 时结束.

所以循环必定终止

(a2).

循环不变量（性质）：

在每次循环开始前，枚举i时，$A[1,\cdots,i-1]$ 来自输入的 $A[1,\cdots,i-1]$ 且有序

初始化：当 $i=2$ 时，$A[1,\cdots,i-1]$ 即 $A[1]$. 有序

保持：循环开始时，$A[1,\cdots,i-1]$ 有序，循环中，将大于 $A[i]$ 的元素后移，再把 $A[i]$ 插到 $A[1,\cdots,i-1]$ 中.

使得 $A[1,\cdots,i]$ 有序，$A[1,\cdots,i]$ 来自输入的 $A[1,\cdots,i]$.

终止：$i \leftarrow n+1$，$A[1,\cdots,n]$ 来自输入且有序.

(b) 最坏情况：反向排序

比较次数：$\sum_{i=2}^{n} i = \frac{(n+2)(n-1)}{2}$

赋值次数：[1, 2, 3, 5, 6, 7] 都在赋值，分别讨论相加即可！

$4(n-1) + 2\sum_{i=2}^{n} i$

最好情况：正向排序.

比较次数：$n-1$

赋值次数：$4(n-1)$.

平均情况：

复杂度 $O(n^2)$

1.4 素数判定算法描述

IsPrime
输入：正整数 n
输出：n 是否为素数
1. For $i \leftarrow 2$ To $n^{\frac{1}{2}}$ Do
2. If i 整除 n then 返回 "no"；
3. 返回 "Yes"

输入规模：$k = \log_2 n$.（整数 n 在计算机内部以二进制方式存储）

操作次数大致为 $n^{\frac{1}{2}} = (2^k)^{\frac{1}{2}} = 2^{k-1}$.

算法复杂度为 $\Theta(n^{\frac{1}{2}})$，为多项式时间复杂度.

1.5. 斐波那契算法 描述

DP

输入：正整数 n

输出：斐波那契的第 n 项.

1. If $n=0$ 或 1 Then 返回 1.
2. For $i \leftarrow 2$ To n.
3. $F[i] \leftarrow F[i-1]+F[i-2]$;
4. 返回 $F[n]$.

输入规模：$k=\log_2 n$.

操作个数：$n=2^k$.

算法复杂度 $O(n)$，多项式时间算法