1.1 欧几里德算法的输入大小为 $\log_2 a$, $\log_2 b$,分别将算法中进行的求余操作和赋值操作的次数表示成 $\log_2 a, \log_2 b$ 的函数，并由此得出欧几里德算法的渐近复杂性。

1.2 理解下面的插入排序算法，并完成后面的分析。

插入排序算法 InsertSort

输入：数组 $A[1:n]$

输出：排序后的数组 $A[1:n]$

```
For  i←2  To  n  Do
    key←A[i];
    j←i-1;
    While  j>0 且 A[j]>key  Do
        A[j+1] ← A[j];
        j←j-1;
    A[j+1] ←key;
```

(a)证明算法必然停止；

(a)利用循环不变量方法，证明算法的正确性。

(b)分别分析最坏情况下、最好情况下、平均情况下算法执行的比较操作次数和赋值操作次数，将分析结果表示成 n 的函数。分析平均复杂度是，假设所有输入服从均匀分布。

1.4 考虑如下的素数判定算法，将整除判定操作视为基本操作。

插入排序算法 IsPrime

输入：输入正整数 n

输出：n 是否为素数

```
For  i←2  To  n^(1/2)  Do
    If  i 整除 n  then  返回“no”;
返回“Yes”
```

指出算法的输入规模，将基本操作个数表达成输入规模的函数，指出这个算法是多项式时间算法还是指数时间算法。

1.5 考虑如下的计算斐波那契数列第 n 项的算法，将加法操作视为基本操作。

斐波那契 DP 算法

输入：正整数 n

输出：斐波那契数列第 n 项

```
If  n=0 或 1  Then 返回 1；
For  i←2  To  n
    F[i] ← F[i-1] + F[i-2];
返回 F[n]
```

指出算法的输入规模，将基本操作个数表达成输入规模的函数，指出这个算法是多项式时间算法还是指数时间算法。