代码

```
import time
def fib(n):
    if n in (1,2):
        return 1
    else:
        return fib(n-1) + fib(n-2)

for i in range(1,35):
    start = time.time()
    res = fib(i)
    print('Time used: %.6f'%(time.time()-start),'seconds,','Fib(%d) ='%i,res)
```

运行结果

```
Time used: 0.000000 seconds, Fib(1) = 1
Time used: 0.000000 seconds, Fib(2) = 1
Time used: 0.000000 seconds, Fib(3) = 2
Time used: 0.000000 seconds, Fib(4) = 3
Time used: 0.000000 seconds, Fib(5) = 5
Time used: 0.000000 seconds, Fib(6) = 8
Time used: 0.000000 seconds, Fib(7) = 13
Time used: 0.000000 seconds, Fib(8) = 21
Time used: 0.000000 seconds, Fib(9) = 34
Time used: 0.000000 seconds, Fib(10) = 55
Time used: 0.000000 seconds, Fib(11) = 89
Time used: 0.000000 seconds, Fib(12) = 144
Time used: 0.000000 seconds, Fib(13) = 233
Time used: 0.001000 seconds, Fib(14) = 377
Time used: 0.000000 seconds, Fib(15) = 610
Time used: 0.000000 seconds, Fib(16) = 987
Time used: 0.001014 seconds, Fib(17) = 1597
Time used: 0.000982 seconds, Fib(18) = 2584
Time used: 0.001998 seconds, Fib(19) = 4181
Time used: 0.002016 seconds, Fib(20) = 6765
Time used: 0.005013 seconds, Fib(21) = 10946
Time used: 0.007012 seconds, Fib(22) = 17711
Time used: 0.010013 seconds, Fib(23) = 28657
Time used: 0.017995 seconds, Fib(24) = 46368
Time used: 0.028983 seconds, Fib(25) = 75025
Time used: 0.046958 seconds, Fib(26) = 121393
Time used: 0.076956 seconds, Fib(27) = 196418
Time used: 0.121932 seconds, Fib(28) = 317811
Time used: 0.211859 seconds, Fib(29) = 514229
Time used: 0.346112 seconds, Fib(30) = 832040
Time used: 0.521769 seconds, Fib(31) = 1346269
Time used: 0.850668 seconds, Fib(32) = 2178309
Time used: 1.373353 seconds, Fib(33) = 3524578
Time used: 2.209270 seconds, Fib(34) = 5702887
```

运行时间变化规律

每次调用 fib(n)，都需要执行 fib(n-1)+fib(n-2)

可类比于二叉树得知，每次调用 fib 都要增加 2 倍的复杂度，故时间复杂度为 O(2^n)