日期	科目	班级	姓名	学号
2022 年 10 月 3 日	算法设计与分析	强基数学 002	吴天阳	2204210460

第一次作业

题目 1. (1) 假设某算法在输入规模为 n 时的计算时间为 $T(n) = 3\times 2^n$. 在某台计算机上实现并完成该算法的时间为 t 秒. 现有另一台计算机, 其运行速度为第一台的 64 倍, 那么在这台新机器上用同一算法在 t 秒内能解输入规模为多大的问题?

(2) 若上述算法的计算时间改进为 $T(n) = n^2$, 其余条件不变, 则在新机器上用 t 秒时间能解输入规模为多大的问题?

(3) 若上述算法的计算时间进一步改进为 $T(n) = 8$, 其余条件不变, 那么在新机器上用 t 秒时间能解输入规模为多大的问题?

解答. (1) $t = 3\times 2^n$，则 $64t = 64\times 3\times 2^n = 3\times 2^{n+5}$，所以新机器可解决 $n+5$ 规模的数据.

(2) $t = n^2$，则 $64t = (5n)^5$，所以新机器可解决 $5n$ 规模的数据.

(3) $t = 8n$，则 $64t = 8\times(64n)$，所以新机器可解决 $64n$ 规模的数据.

题目 2. 证明：如果一个算法在平均情况下的计算时间复杂性为 $\theta(f(n))$，则该算法在最坏情况下所需的计算时间为 $\Omega(f(n))$。

证明. 设 D_N 为规模为 N 的数据集. $T_{arg}(N) = \sum\limits_{I\in D} P(I)f(I) \leqslant \max\limits_{I\in D_N} f(I) \sum\limits_{I\in D_N} P(I) = \max\limits_{I\in D_N} f(I)$，所以该算法计算时间的上界为 $\Omega(f(n))$. □

题目 3. 已知计算函数 $F(n)$（n 为非负整数）的算法如下：

```c
int F(int n){
    if (n==0) return 1;
    int s=0;
    for (int i=0;i<n;i++) s=s+F(i);
    return s+1;
}
```

上述算法在计算 $F(n)$ 的过程中，调用执行 $F(0)$ 的次数是多少？给出函数 $F(n)$ 的非递归算术表达式；分析上述算法的时间复杂度（给出复杂度的递归表达式，并求解）。

解答. $F(n)$ 调用 $F(0)$ 的次数为 2^{n-1}. 非递归算术表达式为 $F(n) = 2^n$. 时间复杂度的递归表达式为 $T(n) = \sum\limits_{k=0}^{n-1} T(k)$，令 $T(0) = 1$，则通过数学归纳法可知，$T(n) = 2^{n-1}$, $n \geqslant 1$. 下面使用归纳法进行证明，$T(1) = T(0) = 1$，假设 $n-1$ 时原命题成立，由命题假设可知

$$F(n) = \sum_{k=1}^{n-1} 2^{k-1} + 1 = \frac{1-2^{n-1}}{1-2} + 1 = 2^{n-1}.$$

故原命题得证.