绪论

算法

算法运行时间

时间频度：基本语句执行次数，可以表示为处理的数据规模 n 的函数 T(n)。

- 循环语句：需要计算实际运行的次数
- 分支语句：按照执行语句多的分支计算

时间复杂度：如果当 n 趋近于无穷大时， T(n) / f(n) 的极限值为不等于零的常数，称 O(f(n)) 为算法的时间复杂度。

- 时间复杂度由时间频度函数中的最高次项决定，不带系数
- 计算方法：找算法中和数据规模n有关的循环语句，计算循环体的执行次数获得时间频度函数。观察时间频度函数中关于n的最高次项，去掉其系数，即是时间复杂度的大O表示。
- 特殊地，如果算法中无执行次数和数据规模有关的语句，时间复杂度为 O(1)。
- 常见时间复杂度：O(1) < O(logn) < O(n) < O(nlogn) < O(n^2) < O(n^3) < O(2^n) < O(n!) < O(n^n)
- 计算时间复杂度的两个定理：
    - 求和定理：若 T1(n) = O(f(n)), T2(n) = O(g(n)), 则 T1(n) + T2(n) = O(max(f(n), g(n)))
    - 乘积定理：若 T1(n) = O(f(n)), T2(n) = O(g(n)), 则 T1(n) * T2(n) = O(f(n) * g(n))
- 通常用最坏时间复杂度来表示算法的时间复杂度。

注意事项

1. 如何写出一个完整的程序：
    1. 参数检查：检查输入参数是否合法
    2. 空间是否足够
    3. 核心操作
    4. 对其他属性的影响
    5. 正确返回