

# 01-C-1 Big-O

#数据结构邓神

Mathematics is more in need  
of good notations than  
of new theorems

- Alan Turing

对于数学而言，最最急需的  
与其说是一个又一个新的定理  
不如说是一个又一个好的记号

好读书不求甚解  
每有会意，便欣然忘食

- 陶渊明

要抓住主要的方面

## 随着问题规模的上升，计算成本如何增长？

注意我们更多的关心的是足够大的问题，注重考察成本的增长趋势（而不是确切的值）

渐进分析：当问题规模足够大后，计算成本如何增长

Asymptotic analysis：当  $n \gg 2$  后 对于规模为  $n$  的输入，算法

需要的基本操作次数  $T(n) = ?$

需要占用的存储单元个数为  $S(n) = ?$  // 通常可以不考虑

## 大O记号 ( Big-O notation )

$T(n) = O(f(n))$

if 存在  $c > 0$  当  $n \gg 2$  时候 有  $T(n) \leq c * f(n)$

与  $T(n)$  相比 ,  $f(n)$  更加简介, 但依然反映前者的增长趋势

常数项可以忽略, 低次项可以忽略

可以认为 Big-0 是对 时间复杂度悲观的估计 (当  $n$  足够大时)