

# 时间复杂度和空间复杂度

前置知识：选择排序、冒泡排序、插入排序、等差数列、等比数列

建议：不要跳过

1，常数操作，固定时间的操作，执行时间和数据量无关

2，时间复杂度，一个和数据量有关、只要高阶项、不要低阶项、不要常数项的操作次数表达式

举例：选择、冒泡、插入

3，严格固定流程的算法，一定强调最差情况！比如插入排序

4，算法流程上利用随机行为作为重要部分的，要看平均或者期望的时间复杂度，因为最差的时间复杂度无意义  
用生成相邻值不同的数组来说明

5，算法流程上利用随机行为作为重要部分的，还有随机快速排序（【必备】课）、跳表（【扩展】课）

也只在乎平均或者期望的时间复杂度，因为最差的时间复杂度无意义

6，时间复杂度的内涵：描述算法运行时间和数据量大小的关系，而且当数据量很大很大时，这种关系相当的本质，并且排除了低阶项、常数时间的干扰

7，空间复杂度，强调额外空间；常数项时间，放弃理论分析、选择用实验来确定，因为不同常数操作的时间不同

8，什么叫最优解，先满足时间复杂度最优，然后尽量少用空间的解

# 时间复杂度和空间复杂度

前置知识：选择排序、冒泡排序、插入排序、等差数列

建议：不要跳过

9，时间复杂度的均摊，用动态数组的扩容来说明（等比数列、均摊的意义）

并查集、单调队列、单调栈、哈希表等结构，均有这个概念。这些内容【必备】课都会讲

10，不要用代码结构来判断时间复杂度，比如只有一个`while`循环的冒泡排序，其实时间复杂度 $O(N^2)$

11，不要用代码结构来判断时间复杂度，比如： $N/1 + N/2 + N/3 + \dots + N/N$ ，这个流程的时间复杂度是 $O(N * \log N)$ ，著名的调和级数

12，时间复杂度只能是对算法流程充分理解才能分析出来，而不是简单的看代码结构！这是一个常见的错误！

甚至有些算法的实现用了多层循环嵌套，但时间复杂度是 $O(N)$ 的。在【必备】课程里会经常见到

13，常见复杂度一览：

$O(1)$   $O(\log N)$   $O(N)$   $O(N * \log N)$   $O(N^2)$  ...  $O(N^k)$   $O(2^N)$  ...  $O(k^N)$  ...  $O(N!)$

14，时间复杂度非常重要，可以直接判断某个方法能不能通过一个题目，根据数据量猜解法，【必备】课都会讲

15，整套课会讲很多算法和数据结构，也会见到很多的时间复杂度的表达，持续看课即可

## 等差数列求和公式

$$S = n / 2 * ( 2 * a_1 + (n - 1) * d )$$

其中，S 是等差数列的和；n 是项数；a<sub>1</sub> 是首项；d 是公差。

也可以认为任何等差数列的都符合：

$a * n^2 + b * n + c$ ，其中a、b、c都是常数