

## 03 时间复杂度与空间复杂度分析

更新时间：2019-08-02 10:05:19



“我们活着不能与草木同腐，不能醉生梦死，枉度人生，要有所作为。

——方志敏”

在各种类型的刷题网站中，大部分题目要求答案代码执行时间都是1秒钟。所以我们在做题之前需要优先考虑我们即将要实现的算法能否在1秒钟之内完成，这就要求我们了解1秒钟大概能做什么量级的运算。在特定的数据范围内我们的算法会不会超时，所以对算法的时间复杂度预估就非常重要。

### 时间复杂度

说到排序，我们的脑子里总会出现几种耳熟能详的排序算法，如插入排序、冒泡排序，快速排序和归并排序。其中插入排序、冒泡排序的时间复杂度是 $O(n^2)$ ，快速排序、归并排序的时间复杂度是 $O(n \log n)$ 。那么这个时间复杂度是什么意思呢？

大家都知道，计算机执行程序是通过二进制指令来执行的，二进制数据最基本的操作有——与（&）、或（|）、非（~）、异或（^），基本四则运算（+、-、\*、/），逻辑运算（>、<、=）是转换成有限次二进制指令来执行，次数不跟参与计算的数据大小有关，我们称之为 $O(1)$ 复杂度。

大家请看如下代码：

```
int sum = 0;
for (int i = 0; i < N; i++) {
    for (int j = 0; j < N; j++) {
        sum++;
    }
}
```

这段代码中，`int sum = 0` 执行了一次，`int i = 0` 执行了一次，`i < N` 执行了N次，`i++` 执行了 N 次，`int j = 0` 执行了N次，`j < N`、`j++` 和 `sum++` 分别执行了 $N^2$ 次，我们假设赋值、比较、`++`，都是一个 `cpu` 指令（事实上不是，可以拆分成更细的操作），那么这段代码执行的指令数量是 $3*N^2+3*N+2$

时间复杂度关注的是影响速度的因素。上述式子，当 N 增大， $N^2$  比 N 大得多，在  $N^2$  面前，N 的影响几乎可以忽略，因此我们忽略掉 N 的部分，将上述代码的时间复杂度简写为  $O(N^2)$ 。

## 1秒钟

1秒钟能执行千万级别的运算，千万级别是什么概念呢？

- n取值在百万-千万左右， $O(n)$ 的算法能在1秒钟内实现
- n取值在十万左右， $O(n\log n)$ 的算法能在1秒钟内实现
- n取值在一千左右， $O(n^2)$ 的算法能在1秒钟内实现
- .....

上面我们看了一个简单小案例的时间复杂度分析，和各种时间复杂度下1 秒钟能够完成多大量级的运算。如果你还是觉得不明白也没关系，在后面的专栏中也会慢慢教你如何分析时间复杂度。下面我们来看下空间复杂度。

## 空间复杂度

内存中基本单位是字节，1个字节为8位，总共可以表示 $2^8=256$ 个值。

我们在设定变量时，一旦变量赋值成功，该变量就在内存中有一片指向的地址，这片地址是操作系统分配的。该变量定义的时候需声明需要多少空间，操作系统会给分配多少空间。一旦分配，这个变量在内存中就固定下来了。

我们在代码中需要声明每个变量的类型，比如`int`，为4个字节的整型；`double`，为8个字节的浮点型；`char`，为一个字节的字符等等。`python`由于可以不用声明类型，我们很少关注每个变量所占用的内存空间。`python`底层仍然是存在基本类型，只是封装了一层动态识别。

基本类型	占用空间	表示数的个数
char	1个字节	256
byte	1个字节	256
bool	1个字节	256(实际上只用了两个值)
short	2个字节	65536
int	4个字节	4294967296
double	8个字节	11位整数，52位尾数，1位符号

空间复杂度也有跟时间复杂度一样的表示方式。

一个n个元素的数组，是 $O(n)$ 的空间复杂度；一个 $n*n$ 的二维数组，是 $O(n^2)$ 的空间复杂度。这里的 $O(n)$ 表示，存在一个跟n无关的常数k，使得申请的内存空间为 `kn`。

例子：（`C++`）

```
int a[N]; // 空间复杂度O(N)，申请的内存空间为4N个字节
```

例子：(`python`)

```
a = [0] * N; # 空间复杂度O(N)，但申请的内存空间不止4N个字节，python为数组封装了一层，里面还会记录一些常用的元数据以便快速查询，比如size
```

## 写在最后

不管是刷题，还是在平时工作中写代码，我们对自己写出来的代码的时间复杂度和空间复杂度应当有所理解，对代码消耗的内存和花费的时间有个底。时间复杂度和空间复杂度都需要根据场景变化而变化。比如响应速度快的，通常用空间换时间的原则；内存放不下，就适当放宽时间复杂度来换内存的优化。设计代码就是对时间、空间、开发成本各种要素的平衡，尽可能取到最优化的策略。

}