

转载请注明链接：[https://blog.csdn.net/feather\\_wch/article/details/50477042](https://blog.csdn.net/feather_wch/article/details/50477042)

归纳总结数据结构和算法中 复杂度分析 这一部分的内容。将知识点以提问的形式总结出来，希望有所帮助。

极客群: 779213122。只招募热爱技术的极客和希望进阶的小伙伴。只分享干货，不水群。非诚勿扰。

# 复杂度分析

版本号：2018/09/29-1(23:30)

- 复杂度分析
  - 为什么需要复杂度分析(13)
  - 分析复杂度的三种方法(4)
  - 时间复杂度(19)
    - $O(1)$
    - $O(\log n)$   $O(n \log n)$
    - $O(m + n)$   $O(m * n)$
    - 最好时间复杂度
    - 最坏时间复杂度
    - 平均时间复杂度
    - 均摊时间复杂度
  - 空间复杂度(5)
  - 知识扩展(2)
  - 实例练习
  - 问题汇总(34)
  - 参考资料

**交流平台：极客窝-QQ群:779213122**

有任何问题，欢迎交流。

每周末有技术交流会，只招募有分享精神和极客精神的小伙伴。

非诚勿扰！非诚勿扰！非诚勿扰！

[点我加入极客窝！](#)

## 为什么需要复杂度分析(13)

1、什么是事后统计法?该方法的局限性是什么?

1. 代码运行后，通过统计、监控，就能得到算法运行的时间和占用的内存大小
2. 但是这种方法有非常大的局限性：
  1. 测试结果非常依赖测试环境---在两种不同设备上运行速度完全不同，甚至出现设备A中算法a快于算法b，但是在设备B中算法b快于算法a的情况。
  2. 测试结果受数据规模的影响很大---例如较小规模的数据排序，插入排序可能比快速排序更快。

## 2、为什么需要复杂度分析？

1. 能够不需要具体的测试数据
2. 可以粗略地估计算法的执行效率

## 3、算法的执行效率是什么？

算法代码的执行时间

## 4、如何可以在不运行代码的情况下，直接得到一段代码的运行时间呢？

时间复杂度分析进行粗略的估计

## 5、分析下列代码的执行时间？

```
int cal(int n){
    int sum = 0;
    int i = 1;
    for(; i <= n; i++){
        sum = sum + i;
    }
    return sum;
}
```

1. 估算每行代码执行时间为  $\text{unit\_time} = T$
2. 总体执行时间： $2 * T + 2 * n * T$
3. 代码执行时间 $T(n)$ 和代码执行次数成正比

## 6、为什么每行代码的执行时间只能粗略估计为unit\_time？

1. 每行代码对应的CPU指令执行的个数是不同的
2. 会包括读取数据、运算、写数据
3. 这些CPU指令执行的时间也是不同的。
4. 只能估算

## 7、分析下列代码的执行时间

```

int cal(int n){
    int sum = 0;
    int i = 1;
    int j = 1;
    for(; i <= n; i++){
        j = 1;
        for(; j <= n; j++){
            sum = sum + i * j;
        }
    }
    return sum;
}

```

总体执行时间:  $3 * T + (2 * n + 2) * n * T = (2n^2 + 2n + 3) * T$

8、 $T(n) = O(f(n))$ 的各个部分是什么意思？

1.  $T(n)$ : 代码执行的时间
2.  $f(n)$ : 每行代码执行的次数总和
3.  $O$ : 表示 $T(n)$  和  $f(n)$  成正比关系

9、 $T(n) = (2*n + 2) * T$  用大O如何表示？

1.  $O(2n + 2)$
2. 忽略低阶、常量和系数, 只记录最大量级:  $O(n)$

10、 $T(n) = (2n^2 + 2n + 3) * T$  用大O如何表示？

1.  $O(2n^2 + 2n + 3)$
2. 忽略低阶、常量和系数, 只记录最大量级:  $O(n^2)$

11、大O时间复杂度是什么？

1. 表示 $T(n)$  和  $f(n)$  成正比关系
2. 大O时间复杂度实际上并不具体表示代码真正的执行时间
3. 代表 代码执行时间- $T(n)$  随 数据规模- $f(n)$  增长的变化趋势
4. 也叫做 渐进时间复杂度-asymptotic time complexity
5. 简称时间复杂度

12、时间复杂度本质是代表了什么？

1. 同义词: 渐进时间复杂度、大O时间复杂度
2. 代表 代码执行时间- $T(n)$  随 数据规模- $f(n)$  增长的变化趋势

13、“时间复杂度可以忽略低阶、常量和系数，只需要记录一个最大量级”的说法是否正确？

1. 因为只要数据规模足够大( $n$ 很大)
2. 低阶、常量和系数并不会明显影响增长趋势，可以忽略

## 分析复杂度的三种方法(4)

### 1、如何分析一段代码的时间复杂度？

有三种实用方法进行分析：

1. 只需要关注循环执行次数最多的一段代码
2. 加法法则
3. 乘法法则

### 2、什么是只需要关注循环执行次数最多的一段代码？下面的实例的时间复杂度是多少？

1. 就是指不需要去关注常量级的代码，只去关注循环执行的代码。
2. 实例的时间复杂度就是  $O(n)$ ：

```
1 int cal(int n) {  
2     int sum = 0;  
3     int i = 1;  
4     for (; i <= n; ++i) {  
5         sum = sum + i;  
6     }  
7     return sum;  
8 }
```

1. 2、3行代码都是常量级的执行时间，与n的大小无关，对复杂度并没有影响，可以直接省略。
2. 循环执行次数最多的是第 4、5 行代码，这两行代码执行了 n 次。

### 3、加法法则是什么？下面实例的时间复杂度是多少？

1. 加法法则是：总复杂度等于量级最大的那段代码的复杂度
2. 实例的时间复杂度 =  $O(1) + O(n) + O(n^2) = O(n^2)$

```

int cal(int n) {
// O(1)
    int sum_1 = 0;
    int p = 1;
    for (; p < 100; ++p) {
        sum_1 = sum_1 + p;
    }
// O(n)
    int sum_2 = 0;
    int q = 1;
    for (; q < n; ++q) {
        sum_2 = sum_2 + q;
    }
// O(n^2)
    int sum_3 = 0;
    int i = 1;
    int j = 1;
    for (; i <= n; ++i) {
        j = 1;
        for (; j <= n; ++j) {
            sum_3 = sum_3 + i * j;
        }
    }

    return sum_1 + sum_2 + sum_3;
}

```

#### 4、乘法法则是什么?下面实例的时间复杂度是多少?

1. 乘法法则是: 嵌套代码的复杂度等于嵌套内代码复杂度和嵌套外代码复杂度的乘积
2. 实例的复杂度 =  $O(n) * O(n) = O(n^2)$

```

1 int cal(int n) {
2     int ret = 0;
3     int i = 1;
4     // O(n)
5     for (; i < n; ++i) {
6         ret = ret + f(i);
7     }
8
9 int f(int n) {
10    int sum = 0;
11    int i = 1;
12    // O(n)
13    for (; i < n; ++i) {
14        sum = sum + i;
15    }
16    return sum;
17 }

```

# 时间复杂度(19)

1、所有的复杂度量级，有哪几种？

7种

## 复杂度量级 (按数量级递增)

- 常量阶  $O(1)$
- 对数阶  $O(\log n)$
- 线性阶  $O(n)$
- 线性对数阶  $O(n \log n)$
- 平方阶  $O(n^2)$ 、立方阶  $O(n^3)$  ...  $k$ 次方阶  $O(n^k)$
- 指数阶  $O(2^n)$
- 阶乘阶  $O(n!)$

2、复杂度量级分为哪两类？

1. 非多项式量级：只有两个- $O(2^n)$ 和  $O(n!)$ 。作为NP问题，属于及其低效的算法。
2. 多项式量级

3、多项式是指什么？

4、复杂度量级中的NP问题是什么？

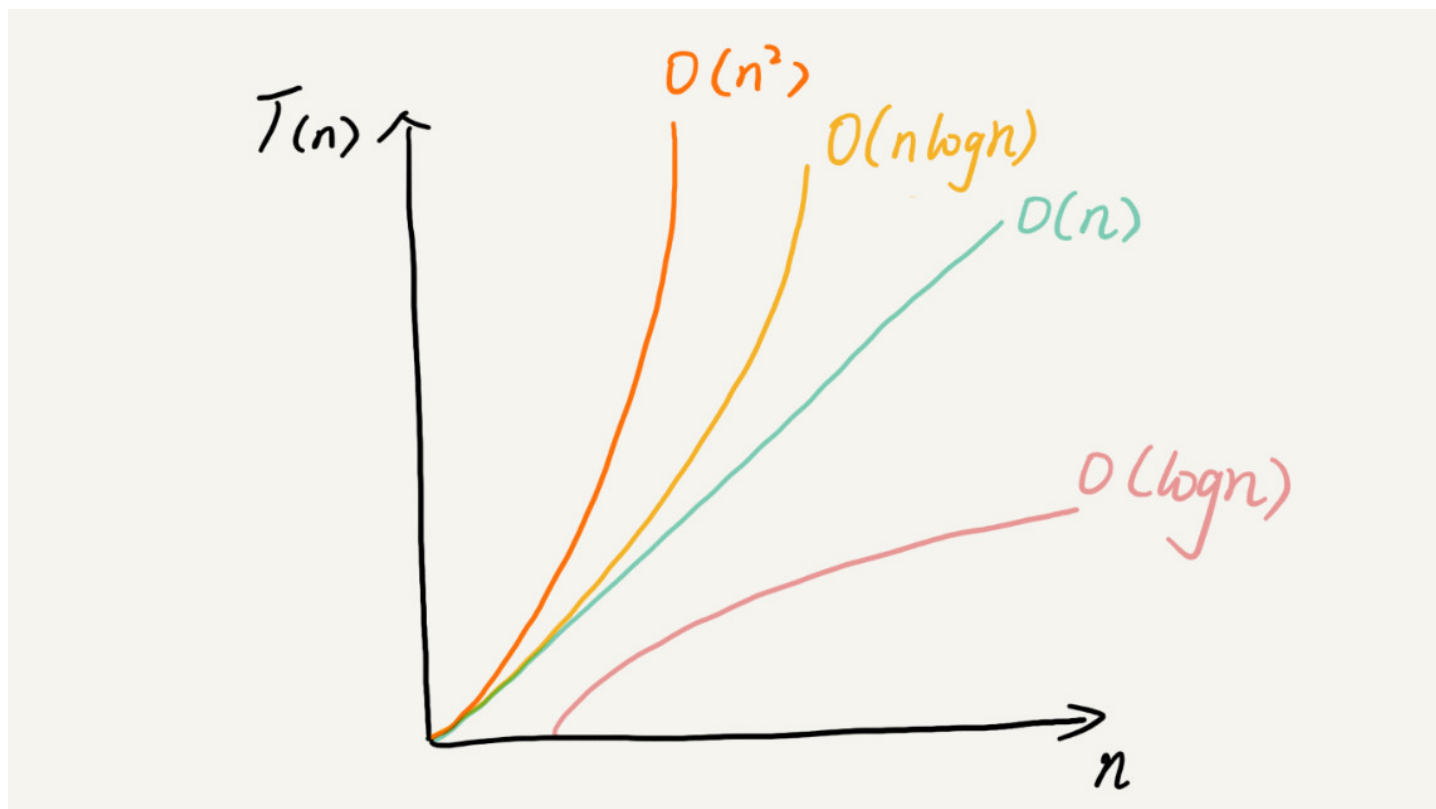
1. 时间复杂度为非多项式量级的算法问题叫作NP问题。
2. NP-Non-Deterministic Polynomial, 非确定多项式
3. 当数据规模  $n$  越来越大时，非多项式量级算法的执行时间会急剧增加，求解问题的执行时间会无限增长。
4. 所以，非多项式时间复杂度的算法是非常低效的算法。

5、最常见的几种复杂度以 $n$ 为 $x$ 轴， $T(n)$ 为 $y$ 轴绘制出关系图？

效率从高到低:

1.  $O(1)$
2.  $O(\log n)$
3.  $O(n)$
4.  $O(n \log n)$

## 5. $O(n^2)$



## $O(1)$

6、 $O(1)$ 是指什么？

1.  $O(1)$ 是常量级时间复杂度的一种表示方法
2. 并不是指只执行了一行代码。
3. 总结为: 只要代码的执行时间不随  $n$  的增大而增长, 这样代码的时间复杂度都是  $O(1)$

7、下面实例的时间复杂度是多少?时间复杂度是 $O(3)$ 是否正确?

```
1 int i = 8;  
2 int j = 6;  
3 int sum = i + j;
```

1.  $O(1)$
2. 不管有多少行, 复杂度都应该是 $O(1)$

8、某算法中不存在循环语句、递归语句, 但是有上千行普通代码, 因此不能单纯用 $O(1)$ 来作为该算法的时间复杂度?

只要代码的执行时间不随  $n$  的增大而增长, 这样代码的时间复杂度都是  $O(1)$

## $O(\log n)$ $O(n \log n)$

9、下面实例的时间复杂度是多少?

```

1 i=1;
2 while (i <= n) {
3     i = i * 2;
4 }

```

1.  $2^k = n$ , 因此执行的次数为

$$\log_2 n$$

2. 时间复杂度为  $O(\log n)$

10、下面实例的时间复杂度是多少？

```

1 i=1;
2 while (i <= n) {
3     i = i * 3;
4 }

```

1. 执行次数为

$$\log_3 n$$

2. 时间复杂度为  $O(\log n)$

11、执行次数为

$$\log_2 n$$

和

$$\log_3 n$$

的算法时间复杂度一样吗？为什么？

1. 一样, 都是  $O(\log n)$
2. 因为根据, 对数之间是可以互相转换的,

$$\log_3 n$$

就等于

$$\log_3 2 * \log_2 n$$

, 所以也可以都转换为以10为底的对数  $\log n$ , 时间复杂度都为  $O(\log n)$

12、 $O(n \log n)$  是什么? 什么情况下会出现?

1. 根据乘法法则, 将时间复杂度为  $O(\log n)$  的算法执行  $n$  次, 就是  $O(n \log n)$



13、哪些算法的时间复杂度是 $O(n\log n)$ ?

1. 归并排序
2. 快速排序

## $O(m + n)$ $O(m * n)$

14、复杂度 $O(m + n)$ 是什么意思?什么情况下会出现?

1. 时间复杂度由两个数据的规模来决定
2. 无法去简单判断哪个数据规模的量级最大
3. 实例的时间复杂度为:  $O(m + n)$

```
int cal(int m, int n) {  
    int sum_1 = 0;  
    int i = 1;  
    for (; i < m; ++i) {  
        sum_1 = sum_1 + i;  
    }  
  
    int sum_2 = 0;  
    int j = 1;  
    for (; j < n; ++j) {  
        sum_2 = sum_2 + j;  
    }  
  
    return sum_1 + sum_2;  
}
```

15、复杂度 $O(m * n)$ 是什么意思?什么情况下会出现?

1. 也是由两个数据的规模来决定
2.  $O(m) * O(n) = O(m * n)$

## 最好时间复杂度

16、最好时间复杂度是什么?

最好情况下的时间复杂度

## 最坏时间复杂度

17、最坏时间复杂度是什么?

最坏情况下的时间复杂度

## 平均时间复杂度

18、平均时间复杂度是什么？

- 1. 求加权平均值，也成为期望值。
- 2. 获取期望值之后，用大O表示法来表达
- 3. 平均时间复杂度一般用于特殊情况

均摊时间复杂度

19、均摊时间复杂度是什么？

- 1. 对均摊复杂度的分析，成为均摊分析\平摊分析
- 2. 相比于平均复杂度，均摊复杂度是更为特殊的情况。是一种特殊的平均时间复杂度
- 3. 一般是n个O(1)复杂度的情况，然后一次O(n)复杂度，然后继续是n个O(1)复杂度。这样可以将O(n)的复杂度均摊到n个O(1)的情况中。就可以得到均摊复杂度。

空间复杂度(5)

1、空间复杂度是什么？

- 1. 渐进空间复杂度（asymptotic space complexity）
- 2. 表示算法的存储空间与数据规模之间的增长关系。

2、时间复杂度和空间复杂度的区别？

| 复杂度   | 全称      | 含义                     |
|-------|---------|------------------------|
| 时间复杂度 | 渐进时间复杂度 | 表示算法的执行时间与数据规模之间的增长关系  |
| 空间复杂度 | 渐进空间复杂度 | 表示算法的存储空间与数据规模之间的增长关系。 |

3、下面实例的空间复杂度是多少？

$O(n) = O(1) + O(n)$

```
1 void print(int n) {  
  // 常量阶的空间：i  
2  int i = 0;  
  // 申请了n个int类型数据的空间  
3  int[] a = new int[n];  
4  for (i; i < n; ++i) {  
5    a[i] = i * i;  
6  }  
7  for (i = n-1; i >= 0; --i) {  
8    print out a[i]  
9  }  
10}
```

## 4、常见的空间复杂度

$O(1)$ 、 $O(n)$ 、 $O(n)$

## 5、 $O(\log n)$ 、 $O(n \log n)$ 这样对数阶的复杂度一般用不上

# 知识扩展(2)

## 1、时间复杂度越高，执行效率就越低？

1. 不是
2. 大O表示法表示的时间复杂度只是一种粗略的估计。在实际的数据规模中，会出现 $O(1)$ 效率低于 $O(n)$ 的可能。
3. 比如 $O(1)$ 的算法中执行的代码成千上万行，而 $O(n)$ 算法中数据规模 $n$ 比较小，所以执行的代码就很少。

## 2、大O是什么？是什么英文的缩写？

1. 大O用于表示增长的关系：如渐变时间复杂度、渐变空间复杂度
2. 大O本身就是一个数学符号

# 实例练习

## 1、下面的算法中的最好、最坏、平均时间复杂度是多少？

```
// 全局变量，大小为 10 的数组 array，长度 len，下标 i。
int array[] = new int[10];
int len = 10;
int i = 0;

// 往数组中添加一个元素
void add(int element) {
    if (i >= len) { // 数组空间不够了
        // 重新申请一个 2 倍大小的数组空间
        int new_array[] = new int[len*2];
        // 把原来 array 数组中的数据依次 copy 到 new_array
        for (int j = 0; j < len; ++j) {
            new_array[j] = array[j];
        }
        // new_array 复制给 array，array 现在大小就是 2 倍 len 了
        array = new_array;
        len = 2 * len;
    }
    // 将 element 放到下标为 i 的位置，下标 i 加一
    array[i] = element;
    ++i;
}
```

1. 最好是 $O(1)$
2. 最坏是 $O(n)$ -具有扩容操作
3. 平均是 $O(1)$

## 问题汇总(34)

本文归纳总结的34个问题。用于自我检测。

1. 什么是事后统计法?该方法的局限性是什么?
2. 为什么需要复杂度分析?
3. 算法的执行效率是什么?
4. 如何可以在不运行代码的情况下, 直接得到一段代码的运行时间呢?
5. 分析下列代码的执行时间(设 $T$ 为一行代码的执行时间)?

```
int cal(int n){
    int sum = 0;
    int i = 1;
    for(; i <= n; i++){
        sum = sum + i;
    }
    return sum;
}
```

6. 为什么每行代码的执行时间只能粗略估计为一个单位时间(unit\_time)?
7.  $T(n) = O(f(n))$ 的各个部分是什么意思?
8.  $T(n) = (2*n + 2) * T$ 用大 $O$ 如何表示?
9.  $T(n) = (2n^2 + 2n + 3) * T$ 用大 $O$ 如何表示?
10. 大 $O$ 时间复杂度是什么?
11. 时间复杂度本质是代表了什么?
12. “时间复杂度可以忽略低阶、常量和系数, 只需要记录一个最大量级”的说法是否正确?
13. 如何分析一段代码的时间复杂度?有哪几种方法?
14. 什么是只需要关注循环执行次数最多的一段代码?
15. 加法法则是什么?
16. 乘法法则是什么?
17. 所有的复杂度量级, 有哪几种?
18. 复杂度量级分为哪两类?
19. 多项式是指什么?
20. 复杂度量级中的NP问题是什么?
21. 最常见的几种复杂度以 $n$ 为 $x$ 轴,  $T(n)$ 为 $y$ 轴绘制出关系图?
22.  $O(1)$ 是指什么?
23. 某算法中不存在循环语句、递归语句, 但是有上千行普通代码, 因此不能单纯用 $O(1)$ 来作为该算法的时间复杂度?
24. 执行次数为 $\log_2$ 底 $n$  $\log_3$ 底 $n$ 的算法时间复杂度一样吗? 为什么?
25.  $O(n\log n)$ 是什么时间复杂度?什么情况下会出现?

26. 哪些算法的时间复杂度是 $O(n \log n)$ ?
27. 复杂度 $O(m + n)$ 是什么意思?什么情况下会出现?
28. 复杂度 $O(m * n)$ 是什么意思?什么情况下会出现?
29. 最好时间复杂度是什么?
30. 最坏时间复杂度是什么?
31. 平均时间复杂度是什么?
32. 均摊时间复杂度是什么?
33. 空间复杂度是什么?
34. 时间复杂度和空间复杂度的区别?
35. 常见的空间复杂度
36.  $O(\log n)$ 、 $O(n \log n)$ 这样对数阶的复杂度一般用不上?
37. 时间复杂度越高, 执行效率就越低?
38. 大O是什么? 是什么英文的缩写?

## 参考资料

1. [03 | 复杂度分析（上）：如何分析、统计算法的执行效率和资源消耗？](#)

## 交流平台：极客窝-QQ群:779213122

有任何问题, 欢迎交流。

每周末有技术交流会, 只招募有分享精神和极客精神的小伙伴。

非诚勿扰! 非诚勿扰! 非诚勿扰!

[点我加入极客窝!](#)