

为什么要学习 复杂度的分析方法

胡船长

初航我带你，远航靠自己

本期内容

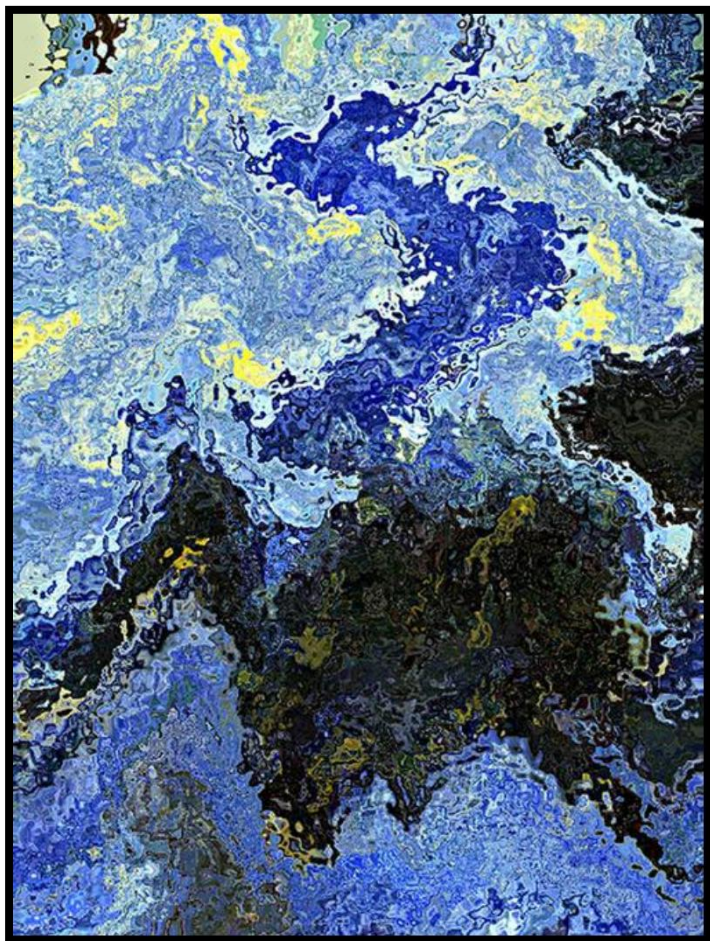
- 一. 复杂度对于算法学习的作用
- 二. 时间复杂度与空间复杂度
- 三. 学会这几种复杂度，解决90%的问题

一. 复杂度对于算法学习的作用

一. 复杂度对于算法学习的作用

怎么区分『**内行**』和『**外行**』？

一. 复杂度对于算法学习的作用



一. 复杂度对于算法学习的作用



一. 复杂度对于算法学习的作用

『**内行**』：能够说清楚一个行业的审美标准

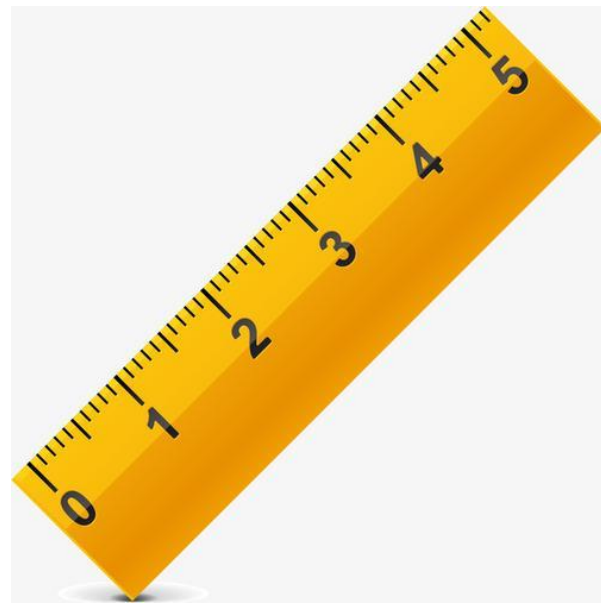
一. 复杂度对于算法学习的作用

『**内行**』：能够说清楚一个行业的审美标准

『**复杂度**』：是程序中最重要的一项审美标准

一. 复杂度对于算法学习的作用

明是非、辨善恶、识好歹、知良莠



二. 时间复杂度与空间复杂度

二. 时间复杂度与空间复杂度

时间复杂度：跑的快不快

空间复杂度：吃的多不多



二. 时间复杂度与空间复杂度

时间换空间，空间换时间

二. 大 O 表示法

大 O 符号：是用于描述函数渐进行为的数学符号。

1. 用常数1取代运行时间中的所有加法常数
2. 在修改后的运行次数函数中，只保留最高阶项
3. 如果最高阶项存在且不是1，则去除与这个项目相乘的常数

二. 大 O 表示法

1. 用常数1取代运行时间中的所有加法常数
2. 在修改后的运行次数函数中，只保留最高阶项
3. 如果最高阶项存在且不是1，则去除与这个项目相乘的常数

```
20 int main() {  
21     int sum = 0;  
22     for (int i = 1; i <= 100; i++) {  
23         sum += i;  
24     }  
25     cout << sum << endl;  
26     return 0;  
27 }
```


二. 大 O 表示法

1. 用常数1取代运行时间中的所有加法常数
2. 在修改后的运行次数函数中，只保留最高阶项
3. 如果最高阶项存在且不是1，则去除与这个项目相乘的常数

```
20 int main() {  
21     int a1, n, d, sum = 0;  
22     cin >> a1 >> n >> d;  
23     for (int i = a1, j = 0; j < n; i += d, j++) {  
24         sum += i;  
25     }  
26     cout << sum << endl;  
27     return 0;  
28 }
```

二. 大 O 表示法

1. 用常数1取代运行时间中的所有加法常数
2. 在修改后的运行次数函数中，只保留最高阶项
3. 如果最高阶项存在且不是1，则去除与这个项目相乘的常数

```
20 int main() {  
21     int a1, n, d, sum = 0;  
22     cin >> a1 >> n >> d;  
23     for (int i = a1, j = 0; j < n; i += d, j++) {  
24         sum += i;  
25     }  
26     cout << sum << endl;  
27     return 0;  
28 }
```

时间复杂度： $O(n)$

空间复杂度： $O(1)$

二. 大 O 表示法

1. 用常数1取代运行时间中的所有加法常数
2. 在修改后的运行次数函数中，只保留最高阶项
3. 如果最高阶项存在且不是1，则去除与这个项目相乘的常数

```
20 int main() {  
21     int sum = (1 + 100) * 100 / 2;  
22     cout << sum << endl;  
23     return 0;  
24 }
```

二. 大 O 表示法

1. 用常数1取代运行时间中的所有加法常数
2. 在修改后的运行次数函数中，只保留最高阶项
3. 如果最高阶项存在且不是1，则去除与这个项目相乘的常数

```
20 int main() {  
21     int a1, n, d, sum = 0;  
22     cin >> a1 >> n >> d;  
23     sum = (a1 + a1 + (n - 1) * d) * n / 2;  
24     cout << sum << endl;  
25     return 0;  
26 }
```

二. 大 O 表示法

1. 用常数1取代运行时间中的所有加法常数
2. 在修改后的运行次数函数中，只保留最高阶项
3. 如果最高阶项存在且不是1，则去除与这个项目相乘的常数

```
20 int main() {  
21     int a1, n, d, sum = 0;  
22     cin >> a1 >> n >> d;  
23     sum = (a1 + a1 + (n - 1) * d) * n / 2;  
24     cout << sum << endl;  
25     return 0;  
26 }
```

时间复杂度： $O(1)$

空间复杂度： $O(1)$

二.大 O 表示法

系数归一、常数归一
保留最高次项

三. 学会这几种复杂度， 解决90%的问题

三. $O(1)$ 时间复杂度

```
20 int main() {  
21     int a1, n, d, sum = 0;  
22     cin >> a1 >> n >> d;  
23     sum = (a1 + a1 + (n - 1) * d) * n / 2;  
24     cout << sum << endl;  
25     return 0;  
26 }
```

三. $O(n)$ 时间复杂度

```
20 int main() {  
21     int a1, n, d, sum = 0;  
22     cin >> a1 >> n >> d;  
23     for (int i = a1, j = 0; j < n; i += d, j++) {  
24         sum += i;  
25     }  
26     cout << sum << endl;  
27     return 0;  
28 }
```

三. $O(n)$ 时间复杂度

```
20 int main() {  
21     int n;  
22     cin >> n;  
23     for (int i = 0; i < n; i += 2) {  
24         cout << i << endl;  
25     }  
26     return 0;  
27 }
```

三. $O(n^2)$ 时间复杂度

```
20 int main() {  
21     int n;  
22     cin >> n;  
23     for (int i = 0; i < n; i++) {  
24         for (int j = 0; j < n; j++) {  
25             // TODO  
26         }  
27     }  
28     return 0;  
29 }
```

三. $O(\log n)$ 时间复杂度

```
20 int main() {  
21     int n;  
22     cin >> n;  
23     for (int i = 1; i <= n; i *= 2) {  
24         cout << i << endl;  
25     }  
26     return 0;  
27 }
```


三. $O(\log n)$ 时间复杂度

```
20 int main() {  
21     int n;  
22     cin >> n;  
23     for (int i = 1; i <= n; i *= 3) {  
24         cout << i << endl;  
25     }  
26     return 0;  
27 }
```

三. $O(nm)$ 时间复杂度

```
20 int main() {  
21     int n, m;  
22     cin >> n >> m;  
23     for (int i = 0; i < n; i++) {  
24         for (int j = 0; j < m; j++) {  
25             cout << i << " " << j << endl;  
26         }  
27     }  
28     return 0;  
29 }
```

三. $O(n+m)$ 时间复杂度

```
20 int main() {  
21     int n, m;  
22     cin >> n >> m;  
23     for (int i = 0; i < n; i++) {  
24         cout << i << endl;  
25     }  
26     for (int i = 0; i < m; i++) {  
27         cout << i << endl;  
28     }  
29     return 0;  
30 }
```

三. 随堂练习1

```
20 int main() {
21     int n, m;
22     cin >> n >> m;
23     for (int i = 0; i < n; i += 4) {
24         for (int j = 0; j < m; j++) {
25             cout << i << " " << j << endl;
26         }
27     }
28     return 0;
29 }
```

三. 随堂练习2

```
20 int main() {  
21     int n, m;  
22     cin >> n >> m;  
23     for (int i = 0; i * i < n; i++) {  
24         cout << i << endl;  
25     }  
26     return 0;  
27 }
```

不要考虑太多，坚持看完，
你就已经超过了95%的人。

5. 整型数据类型

 | 3.58万次播放

54. 主函数参数

 | 2892次播放