

本文内容是对空间复杂度的梳理和总结，本文内容包括：

1 空间复杂度

简单的说就是程序运行所需要的存储空间。

一个算法的空间复杂度，也常用大 O 记法表示。

要知道每一个算法所编写的程序，运行过程中都需要占用大小不等的存储空间，算法的存储量包括：

1. 程序本身所占空间
2. 输入数据所占空间
3. 辅助变量所占空间

首先，程序自身所占用的存储空间取决于其包含的代码量，如果要压缩这部分存储空间，就要求我们在实现功能的同时，尽可能编写足够短的代码。

其次，程序运行过程中输入输出的数据，往往由要解决的问题而定，即便所用算法不同，程序输入输出所占用的存储空间也是相近的。

事实上，对算法的空间复杂度影响最大的，往往是程序运行过程中所申请的临时存储空间。不同的算法所编写出的程序，其运行时申请的临时存储空间通常会有较大不同。

举个例子：

```
1 | Scanner scanner = new Scanner(System.in)
2 | int n;
3 | n = scanner.nextInt();
4 | int a[10];
```

通过分析不难看出，这段程序在运行时所申请的临时空间，并不随 n 的值而变化。而如果将第 4 行代码改为：

```
1 | int a[n];
```

此时，程序运行所申请的临时空间，和 n 值有直接的关联。

- 如果程序所占用的存储空间和输入值无关，则该程序的空间复杂度就为 $O(1)$
- 如果随着输入值 n 的增大，程序申请的临时空间成线性增长，则程序的空间复杂度用 $O(n)$ 表示
- 如果随着输入值 n 的增大，程序申请的临时空间成 n^2 关系增长，则程序的空间复杂度用 $O(n^2)$ 表示
- 如果随着输入值 n 的增大，程序申请的临时空间成 n^3 关系增长，则程序的空间复杂度用 $O(n^3)$ 表示

写代码我们可以用时间换空间，也可以用空间换时间。加大空间消耗来换取运行时间的缩短，加大时间的消耗换取空间，我们一般选择空间换时间。一般说复杂度是指时间复杂度。