

双指针算法

双指针算法是一种通过设置**两个指针**不断进行**单向移动**来解决问题的算法。

双指针算法最核心的用途就是**优化时间复杂度**。

【核心思想】：

原本两个指针是有 n^2 种组合，因此时间复杂度是 $O(n^2)$ 。

而双指针算法就是运用单调性使得指针只能单向移动，因此总的时间复杂度只有 $O(2n)$ ，也就是 $O(n)$ 。

之所以双指针可以实现 $O(n)$ 的时间复杂度是因为指针只能单向移动，没有指针的回溯，而且每一步都会有指针移动。

而朴素的 $O(n^2)$ 算法的问题就在于指针经常**回溯到之前的位置**。

双指针算法的模板一般都可以写成下面的形式(模板)：

```
1  for (int i = 0, j = 0; i < n; i++) {  
2  
3      while (j < i && check(i, j)) j++;  
4  
5      // 每道题目的具体逻辑  
6  }
```

因为双指针算法是一种优化时间复杂度的方法，所以我们可以首先写出最朴素的**两层循环**的写法。

然后考虑题目中是否具有**单调性**。

即当其中一个指针 i 向后移动时，在希望得到答案的情况下，另一个指针 j 是不是只能向着一个方向移动。

如果是，说明题目具有单调性，可以通过**双指针算法优化**。

快乐刷题

- [P323 最长连续不重复子序列](#)
- [P324 数组元素的目标和](#)
- [P325 判断子序列](#)