滑动窗口：

1. 简介：利用双指针在某种数据结构（↓一般是 array）中虚构出了一个窗口。窗口按照规则向右移动，最终找到全局解。

2. 目的：将暴力解的 $O(N^2)$ 中的重复计算去掉 sp
   s[i = i+k]，每次移动时改变的只是窗口两端的元素

```
window_sum = sum([elem for elem in n[:k]])
res = window_sum

for i in range(k, len(n) - k + 1):
    window_sum = window_sum - n[i] + n[i + k]
    res = max(res, window_sum)

return res
```

$O(N^2)$
↓
$O(N)$

主要解决没有利用好前置状态计算好的信息而带来的复杂度增加的问题

3. 基本步骤：
   - i. 向窗口增加元素 （↙ 就窗口.）
   - ii. 向窗口删减元素
   - iii. 更新信息

4. 总结：
   - i. 核心：解决没有利用前置状态计算好的信息而带来的复杂度增加的问题
     → 每次移动窗口变化的只有头尾元素，中间的内容是不变的
   - ii. 窗口大小：
     - 可变滑动窗口
     - 固定滑动窗口