

{% note info %}

摘要

Title: 102. 最佳牛围栏

Tag: 浮点二分、二分、平均数、前缀和

Memory Limit: 64 MB

Time Limit: 1000 ms

{% endnote %}

Powered by: NEFU AB-IN

[Link](#)

[@TOC](#)

102. 最佳牛围栏

• 题意

农夫约翰的农场由 N 块田地组成，每块地里都有一定数量的牛，其数量不会少于 1 头，也不会超过 2000 头。

约翰希望用围栏将一部分连续的田地围起来，并使得围起来的区域内每块地包含的牛的数量的平均值达到最大。

围起区域内至少需要包含 F 块地，其中 F 会在输入中给出。

在给定条件下，计算围起区域内每块地包含的牛的数量的平均值可能的最大值是多少。

• 思路

首先

- 当看到求最大值和最小值时想到二分，所以此题为浮点二分平均值
- 平均值具有单调性，因为一个区间如果平均值 $\geq x$ ，那么比 x 小的也一定满足

其次

- 看到一个区间的平均值是否大于 mid ，就要想到让这个区间都减去 mid ，从而判断这个区间的和是否为正
- 然后

- 求长度大于等于 x 的区间的最大值时，可采用双指针+前缀和，即遍历右端点 j 的区间为 $[x, n]$ ，更新左端点 $[0, j]$ ，使 $sum[j] - sum[i]$ 最大 (sum 数组为前缀和数组)
- 如何使 $sum[j] - sum[i]$ 最大？就让 $sum[i]$ 最小，即每次更新右端点时，更新 $sum[i]$ 的最小值
 - 即 $minv = \min(minv, sum[j - x])$ ，其实就是求出 $sum[0], sum[1], sum[2] \dots$ 的最小值，即最小前缀

那么这个题解法就出来了，即浮点二分平均值，用区间的数都减去 mid 判断这个区间是否可行，可行就向右遍历

• 代码

```
'''
Author: NEFU AB-IN
Date: 2022-02-20 15:10:20
FilePath: \ACM\Acwing\102.py
LastEditTime: 2022-02-20 16:46:18
'''
N = int(1e5 + 100)

a = [0]
b = [0] * N

def check(x):
    for i in range(1, n + 1):
        b[i] = b[i - 1] + a[i] - x
    minv = 0
    for i in range(f, n + 1):
        minv = min(minv, b[i - f])
        if b[i] - minv >= 0:
            return True
    return False

n, f = map(int, input().split())

l, r = 0.0, 0.0

for i in range(n):
    x = int(input())
    a.append(x)
    r = max(r, float(x))

while r - l > 1e-6:
```

```
mid = (l + r) / 2
if check(mid):
    l = mid
else:
    r = mid

print(int(r * 1000))
```