

# 单调队列题目讨论

# ➤ 问题1：修剪草坪

## H 修剪草坪

时间限制：10000 MS 空间限制：65536 KB

### 问题描述

在一年前赢得了小镇的最佳草坪比赛后，FJ变得很懒，再也没有修剪过草坪。现在，新一轮的最佳草坪比赛又开始了，FJ希望能够再次夺冠。然而，FJ的草坪非常脏乱，因此，FJ只能让他的奶牛来完成这项工作。FJ有 $N$  ( $1 \leq N \leq 100,000$ )只排成一排的奶牛，编号为 $1 \cdots N$ 。每只奶牛的效率是不同的，奶牛 $i$ 的效率为 $E_i$  ( $0 \leq E_i \leq 1,000,000,000$ )。靠近的奶牛们很熟悉，因此，如果FJ安排超过 $K$ 只连续的奶牛，那么，这些奶牛就会罢工去开派对:)。因此，现在FJ需要你的帮助，计算FJ可以得到的最大效率，并且该方案中没有连续的超过 $K$ 只奶牛。

### 输入格式

第一行：空格隔开的两个整数 $N$ 和 $K$   
第二到 $N+1$ 行：第 $i+1$ 行有一个整数 $E_i$

### 输出格式

第一行：一个值，表示FJ可以得到的最大的效率值

## ➤ 问题1：修建草坪

给你n个数字的数组，至多连续取k个数字，求取出的最大和。

预处理前缀和sum[i]之后， $dp[i] = \max(dp[j] + \text{sum}[i] - \text{sum}[j + 1])$   $j \in [i-k-1, i-2]$ ;

那么我们滑动的是j, 把跟j无关项提出去，就是再求窗口内 $dp[j] - \text{sum}[j+1]$ 的最大值。

显然单调队列优化。

## ➤ 问题1：修建草坪

参考代码：

```
int n, k, a[N], q[N], l = 1, r;
ll dp[N], sum[N];
inline ll S(int i) { return dp[i] - sum[i + 1]; }
int main() {
    cin >> n >> k;
    for (int i = 1; i <= n; i++) {
        scanf("%d", &a[i]);
        sum[i] = sum[i - 1] + a[i];
    }
    if (k >= n) {
        cout << sum[N];
        return 0;
    }
    if (!k) {
        cout << 0;
        return 0;
    }
    for (int i = 1; i <= k; i++) {
        dp[i] = sum[i];
        while (l <= r && S(i - 1) >= S(q[r])) r--;
        q[++r] = i - 1;
    }
    for (int i = k + 1; i <= n; i++) {
        while (l <= r && q[l] + 1 < i - k) l++;
        dp[i] = S(q[l]) + sum[i];
        while (l <= r && S(i - 1) >= S(q[r])) r--;
        q[++r] = i - 1;
    }
    printf("%lld", max(dp[n], dp[n - 1]));
    return 0;
}
```

## ➤ 问题2：生日礼物

### I 【SCOI2009】生日礼物

时间限制：10000 MS 空间限制：165536 KB

#### 问题描述

小西有一条很长的彩带，彩带上挂着各式各样的彩珠。已知彩珠有 $N$ 个，分为 $K$ 种。简单的说，可以将彩带考虑为 $x$ 轴，每一个彩珠有一个对应的坐标(即位置)。某些坐标上可以没有彩珠，但多个彩珠也可以出现在同一个位置上。

小布生日快到了，于是小西打算剪一段彩带送给小布。为了让礼物彩带足够漂亮，小西希望这一段彩带中能包含所有种类的彩珠。同时，为了方便，小西希望这段彩带尽可能短，你能帮助小西计算这个最短的长度么？彩带的长度即为彩带开始位置到结束位置的位置差。

#### 输入格式

第一行包含两个整数 $N, K$ ，分别表示彩珠的总数以及种类数。

接下来 $K$ 行，每行第一个数为 $T_i$ ，表示第 $i$ 种彩珠的数目。接下来按升序给出 $T_i$ 个非负整数，为这 $T_i$ 个彩珠分别出现的位置。

#### 输出格式

一行，为最短彩带长度。

## ➤ 问题2：生日礼物

找出最短的一段区间包含所有种类的彩珠。

先对所有的点按x坐标进行排序

我们维护两个指针l,r，每一次令r不断右移直到所有颜色齐全，再不断右移l直到颜色数不足，那么此时[l-1,r]这个区间里的彩珠肯定能构成所有颜色，且不难发现不会有包含这一区间的方案比它更优

这种思想严格来说应该称为two-pointer(尺取法)，不过还是可以用双端队列优雅实现(：

## ➤ 问题2：生日礼物

参考代码：

```
long long l, r, ans = 2147483649, x;
int b[99999], n, k, i, q, j, cnt;
struct node {
    long long v, x;
} a[1000005];
bool cmp(node a, node b) { return a.x < b.x; }
int main() {
    scanf("%d%d", &n, &k);
    for (i = 1; i <= k; i++) {
        scanf("%d", &q);
        for (j = 1; j <= q; j++) {
            scanf("%lld", &x);
            a[++cnt].v = i;
            a[cnt].x = x;
        }
    }
    sort(a + 1, a + 1 + n, cmp);
    l = 1;
    cnt = 0;
    for (r = 1; r <= n; r++) {
        if (b[a[r].v] == 0)
            ++cnt;
        b[a[r].v]++;
        while (b[a[l].v] > 1) b[a[l].v]--, l++;
        if (cnt == k)
            ans = min(ans, a[r].x - a[l].x);
    }
    printf("%lld", ans);
}
```



## ➤ 问题3：划区灌溉

### J 划区灌溉

时间限制：10000 MS 空间限制：65536 KB

#### 问题描述

约翰的奶牛们发现山脊上的草特别美味。为了维持草的生长，约翰打算安装若干喷灌器。为简化问题，山脊可以看成一维的数轴，长为 $L(1 \leq L \leq 10^6)$ ，而且 $L$ 一定是一个偶数。每个喷灌器可以双向喷灌，并有确定的射程，该射程不短于 $A$ ，不长于 $B$ ， $A, B(1 \leq A \leq B \leq 10^3)$ 都是给出的正整数。它所在位置的两边射程内，都属它的灌溉区域。注意，一个喷灌器往左右两边喷射的距离是一样的，比如往左喷的距离是 $x$ ，那么往右也是 $x(A \leq x \leq B)$ 。

现要求山脊的每一个区域都被灌溉到，**喷灌器不能将水喷到山脊以外的区域**，而且喷灌器的灌溉区域不允许重叠。约翰有 $N(1 \leq N \leq 10^3)$ 只奶牛，每一只都有特别喜爱的草区，第 $i$ 奶牛喜爱的草区是 $[S_i, E_i]$ ，不同奶牛的草区可以重叠( $E_i - S_i \leq 2 * B$ )。现要求，每只奶牛的草区仅被一个喷灌器灌溉。寻找最少需要的喷灌器数目。

#### 输入格式

第一行，两个整数 $N$ 和 $L$

第二行，两个整数 $A$ 和 $B$

接下来 $N$ 行，每行两个整数 $S$ 和 $E(0 \leq S < E \leq L)$ ，表示每头奶牛喜欢的草区的起止位置

#### 输出格式

一行，一个整数，表示最少需要的喷灌器的数量，若无解，输出-1



## ➤ 问题3：划区灌溉

DP[i] 表示当覆盖区间为[1,i] 时用的最少的喷灌器的数量。

$$DP[i] = \min(DP[j]) + 1, A \leq (i - j) / 2 \leq B$$

转移过程中dp[i], dp[j]满足单调性，显然可以用单调队列记录值和编号优化。

还要判是否有解。

## ➤ 问题3：划区灌溉

---

参考代码：略。

## ➤ 问题3：布丁

K

### 【单调队列】布丁

时间限制：10000 MS 空间限制：65536 KB

#### 问题描述

FJ建立了一个布丁工厂。在接下来的 $N$ 个星期里，原料牛奶和劳动力的价格会有很大波动。FJ希望能够在满足消费者需求的前提下尽量减小花费。

FJ预计接下来每个星期会需要 $C_i$ 元钱来生产一单位布丁，且消费者会需要 $P_i$ 单位布丁。

FJ每星期即可以生产布丁，也可以储存布丁供以后使用。它的仓库存储一星期一单位布丁需要 $S$ 元钱。但是由于布丁有保质期，所以最多只能保存 $T$ 星期。也就是说 $x$ 星期生产的布丁可以在 $(x+T)$ 星期销售，但是不能在 $(x+T+1)$ 星期销售。

帮助FJ安排生产与存储的方案使得在满足消费者需求的前提下尽量减小花费。

#### 输入格式

第一行为 $N, S$ 与 $T$ 。

第二行到第 $(N+1)$ 行：每一行两个数，即 $C_i$ 与 $P_i$ 。

#### 输出格式

仅一个数，即满足顾客需求前提下的最小花费。

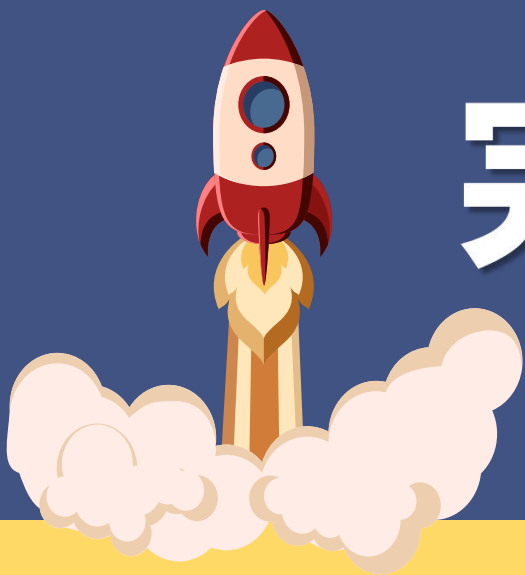
## ➤ 问题3：布丁

问题是求满足每周需求量最小的费用，显然每一周的最小费用是相互独立的。  
又因为最小费用 = 单价 \* 需求量，需求量是已经确定了的。

设单价为 $V[i]$

则 $V[i] = \min( C[j] + S * (i - j) )$  ( $0 < j \leq i, j \geq i - T$ )。

利用单调队列优化求解即可。



# 完！

以梦为码 心之所往

