# K - 分序列

##### Time Limit: 3000/1000MS (Java/Others)     Memory Limit: 65535/65535KB (Java/Others)

Submit Status

最近，你的老师给了你一个序列，让你把这个序列划分成若干段小的序列。注意，这个操作不能改变每个数在原来序列中的位置。

对于分开的每个小的序列，都有一个对应的权值，计算方法如下：

假设这个小序列中的数是:c1,c2⋯ckc1,c2⋯ck，则权值为(∑k1ci)2+m(∑1kci)2+m，其中，mm为常数。

现在你要找到一种分法，使得所有小序列的总权值之和最小。

## Input

题目有多组数据。

对于每一组数据，第一行是两个数n,mn,m，分别表示的是序列中的数的个数，和公式中出现的常数mm。

接下来有n个数，其中第ii个数表示的是序列中的第ii个数cici。

0<n≤5000000<n≤500000

0≤ci0≤ci

## Output

对于每一组数据，输出一行，包含一个数，表示最小的权值和。

## Sample input and output

| **Sample Input** | **Sample Output** |
| --- | --- |
| 5 5  5 9 5 7 5 | 230 |

## Hint

保证答案不超过long long 的范围。

题意：

如题。

题解：

斜率优化DP。

首先，我们根据题意，可以很容易想到时间复杂度为O(n^2)的算法：

dp[i]=min(dp[j]+(sum[i]-sum[j])^2+m)(1<=j<i)

其中，sum[i]表示数列的前i项和。题目中，n最大可达5\*10^6，我们需要想办法对这个式子进行优化。

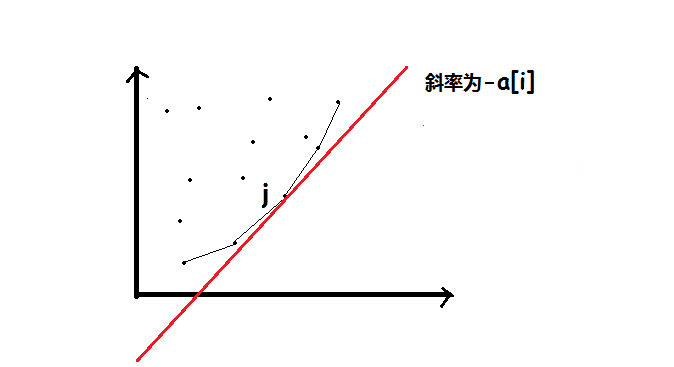
我们可以假设k<j<i且在j点决策比在k点好,则得到不等式:

dp[j]+(sum[i]-sum[j])^2+m> dp[k]+(sum[i]-sum[k])^2+m

化简可得

令Xi=2\*sum[i],Yi=dp[i]+sqr(sum[i]),则我们可以将不等式左侧化为(Xj,Yj),(Xk,Yk)两点斜率。我们发现，斜率具有单调性。

如果我们从图形上去理解，可能更为直观：



之后我们按照如上图所示的方法，维护一个凹函数(利用单调队列)，将不在凹函数上的点从队列删除(它们必然不是最优决策点)，每次dp只需要取队首元素进行决策即可。这样，时间复杂度被降为O(n).