乍一看很难做，而且长得一脸要转化的样子，不妨考虑简单情形，来贪心找性质。

如果 $k=2$，可以先把左边一个元素单独成段，逐渐调整。

一个自然的想法是两边尽量均分，考虑一个位置，若 $s_1+k>s_2$，则显然 $k>s_2-s_1$，所以逐渐移动到这个位置即可。

k 段也可以如上调整，使得每一刀都符合要求，所以解一定存在(后面有证明)。

然后就是魔幻转化，均分实际上意味着，对于一个实数满足 $k>1$，都应该有 $\sum(s_i)^k$ 最小，其中 s_i 表示第 i 段和，考虑取 $k=2$。

于是问题变成了，将序列划分为 k 个连续段，最小化每一段平方和的和。

> 简单证明：
>
> 若如上划分非法，则存在 $s_1\ge s_2, s_1-s_2>\max(m_1,m_2)$。
>
> 考虑把 s_1 划分一个给 s_2，此时平方和变为 $(s_1-k)^2+(s_2+k)^2$，暴力展开发现平方和变小，与假设矛盾，故若划分非法，则一定可以调整为平方和更小的方案，所以平方和最小时一定合法。

容易想到设 $f(i,j)$ 表示前 i 个数划分为 j 段的最小平方和，转移点容易记录，不难斜率优化，可以做到 $\mathcal{O}(nk)$，并过不去。

然而注意到，随着段数的增加，平方和是一个下凸函数，于是可以 wqs 二分来做，答案容易求，难点在于构造方案，三点共线非常的影响我们构造方案。

考虑三点共线我们是如何求权值的？在最大化权值的情况下，最大化或者最小化选的物品个数，若最后没有切到 k，说明此时的斜率和 k 共线，用此时的斜率来计算 k 即可。

总之我们应该先跑出斜率 k，然后跑出 `L[i]`，`R[i]` 表示极化权值的情况下，最少/最多能划分多少段。

从后往前倒推即可，上一个位置理应是可以转移到我的，而且划分段应当包含 $x-1$ 的，递归去做即可。