台

我们先差分一下，然后每次操作花费 a_i 代价给 i 上的数加一，给 $i+1$ 上的数减一，或者反过来。

我们对 a 求前缀和，然后 i 位置上的数我们把它移到 sa_i 的位置上，这样就变成了我们可以花费两个数距离的代价把一个 1 从一个数移到另一个数上。

当 $h_i > d$ 时，我们在 i 放置 $h_i - d$ 个必须送入传送门的箱子与 $2d$ 个没有要求的箱子。

当 $-d \le h_i \le d$ 时，我们在 i 放置 $h_i + d$ 个没有要求的箱子与 $d - h_i$ 个没有要求的传送门。

当 $h_i < -d$ 时，我们在 i 放置 $-d - h_i$ 个必须接收箱子的传送门与 $2d$ 个没有要求的传送门。

这样就变成了把箱子匹配传送门的最小代价。

我们考虑怎么计算，我们从左到右扫描箱子与传送门，对于必须匹配东西的箱子与传送门，我们先在负无穷远处添加无穷个传送门与箱子与之配对。

我们维护两个堆，当扫到一个箱子时，我们把它与传送门堆中代价最小的传送门匹配，我们要把距离贡献分开处理，假设堆顶代价为 v ，则匹配代价为 $s_i + v$。

考虑到后续可能有一个传送门要来抢夺这个箱子，我们在箱子堆中添加一个代价为 $-v - 2s_i$ 的箱子。

考虑到后续可能有一个箱子要来抢夺这个传送门，在这个箱子不是必须被匹配的情况下，我们在传送门堆中添加一个代价为 $-s_i$ 的传送门。

对于传送门的情况是完全对称的。

我们发现放入箱子与传送门堆中的权值有很多是相同的，于是我们可以把它们合并。

关于复杂度，我们发现我们加入一批箱子时，那么这一批箱子会分裂为代价不等的 x 堆箱子扔进箱子堆，相应的，也会有 $O(x)$ 个代价不等的传送门会被合并成一个等代价的传送门，对于传送门同理。

由于传送门总数是不多的，所以分裂出代价不等的传送门是很少的。对于一个箱子，我们取每取一次堆顶，传送门代价种类会少 1 ，而每次加入的代价不同的传送门种类数至多有 $2d+10$ 个，所以复杂度均摊 $O(nd\log(n))$。