Lun

简单枚举即可

$Konj$

实质就是图的联通，把线段抽象成点就是典型的图的联通问题，每次联通进行绘图即可。一条线段的两个端点如何建图呢？想想网格图的并查集是如何转化为节点的?

$Simfonija$

"贪心的想法，肯定是把K个使用在两头，因为两头的差值大嘛(构造差值并排序)，换句话来说就是中间留下连续的 $N-K$ 个，假设起点为 L，那么终点就是 $L+N-K-1$，对于这段区间 $[L, L+N-K-1]$ 里的每个数字 $+X$，最后套上绝对值求解最小值而已，表达式如下：
$\min(|X+A[L]-B[L]|+|X+A[L+1]-B[L+1]|+...+|X+A[L+N-K-1]-B[L+N-K-1]|)$

就是典型的数学中位数。

EGZ

能从 i 走到所有跑道相当于能从 i 走到 1 和 n，边反向后就相当于能从 1 和 n 走到 i;
为了方便叙述,把 1 到 n 叫做 x 坐标， 1 到 $(m+1)$ 叫做 y 坐标。
然后我们将图上下翻转(y 坐标)后，能从 1 走到 i 的话一定会经过前 $i-1$ 条向右的边，且这些边的 y 坐标不下降（从左往右看）。
那么我们设 $F[0][i]$ 表示从 1 走到 i 最少加边数量，那么有 $F[0][i]=i-1-LIS$;
这里的 LIS 是 x 坐标在 1 到 $(i-1)$的边的 y 坐标形成的最长不下降子序列长度。
LIS可以 $O(nlogn)$ 预处理出来，同时我们反着处理一遍向左的边就能处理出从 n 走到 i 的最小加边数量 $F[1][i]$
其实 $F[0][i]$ 的值从左往右看是单调不降的，$F[1][i]$ 的值从右往左看也依然是单调不降的
最后一趟 $O(n)$ 的枚举即可，因为 $F[1][i]$ 单调不降(从后往前看)，所以再上一次的基础之上就OK。