A

从小到大枚举最小值X，使用并查集将所有小于X的位置都并起来。

对于X，利用并查集往左往右找到前两个大于X的位置，讨论一下可以计算出以X为第三大的区间的情况。

时间复杂度$O(n\log n)$

B

发现机器人相遇的次数只和一开始它们的相对位置有关（和绝对位置无关）。我们先假设一开始第一个机器人的 x 坐标和第二个机器人的 x 坐标的差为 D。

对于某个时刻，如果第一个机器人的速度为 v，第二个机器人的速度为 w，那么 D 就会增加 $v-w$。并且，总相遇次数就是 D 为 0 的时刻数。

思考发现，对于所有不同的 D，相遇次数的最大值可以变成下面这个问题：

- 0 时刻有变量 $k=0$。
- 每个 ≥ 1 的时刻，k 都会增加对应的 $v-w$。
- 每个时刻，位置 k 的计数器增加 1。
- 答案为所有位置上计数器的最大值。

于是，我们可以把时刻分成 $O(n)$ 段（假设 n,m 同阶），对于每一段，D 增加的值都是固定的常数 c。由于 $c\in[-2,2]$，我们可以按位置的奇偶性分开考虑。对于每一段时刻，都相当于给一个线段加了 1，所以用扫描线即可。时间复杂度 $O(n\log n)$。

C

考虑根号分治

定义一个值x出现次数为$size[x]$，如果没有修改操作，则预处理所有$size$大于$\sqrt{n}$的值到所有其他值的答案，以及每个值出现位置的一个排序后的数组。 查询的时候如果x和y中有一个是$size$大于$\sqrt{n}$的，则可以直接查询预处理的信息，否则进行归并 $O(n\sqrt{n}+m\sqrt{n})$。 有修改操作即在这个做法上改良，因为发现这个做法的根号平衡并没有卡满，所以有改良余地 假设把所有x变成y 由于可以用一些技巧使得x变成y等价于y变成x，所以不妨设$size[x]<size[y]$ 定义$size$大于$\sqrt{n}$为大，$size$小于等于$\sqrt{n}$为小 定义$ans[x][y]$为x到y去除附属集合的部分的答案（附属集合是什么下面有说） x取遍所有值，y只有所有大的值，总$O(\sqrt{n})$个

修改：

如果x和y均为大，则可以暴力重构，$O(n)$处理出y这个值对每个其他值的答案，因为这样的重构最多进行$O(\sqrt{n})$次，所以这部分复杂度为$O(n\sqrt{n})$ 如果x和y均为小，则可以直接归并两个值的位置的排序后的数组，单次$O(\sqrt{n})$，所以这部分复杂度为$O(m\sqrt{n})$ 如果x为小，y为大，发现小合并进大的均摊位置数为$O(n)$，对这个再次进行根号平衡 对于每个大，维护一个附属的位置集合，这个位置集合的大小不超过$\sqrt{n}$ 每次把小的x合并入大的y的时候，即合并入这个附属集合，并且用x到所有大的值的答案更新y到所有大的值的答案，这样 如果合并后附属集合大小超过$\sqrt{n}$，则$O(n)$重构y到所有值的答案，这个过程最多进行$O(\sqrt{n})$次，均摊复杂度$O(n\sqrt{n})$ 由于大的值个数$<=\sqrt{n}$, 附属集合大小$<=\sqrt{n}$，这一部分是单次$O(\sqrt{n})$的，所以这部分复杂度为$O(m\sqrt{n})$

查询：

如果x和y均为大，则在维护的答案中查询$min(ans[x][y],ans[y][x])$，然后将x的附属集合和y的附属集合进行归并，这一部分是单次$O(\sqrt{n})$的，所以这部分复杂度为$O(m\sqrt{n})$ 如果x和y均为小，则进行一次归并即可，所以这部分复杂度为$O(m\sqrt{n})$ 如果x为小，y为大，则在维护的答案中查询$ans[x][y]$，然后将x和y的附属集合进行归并，这一部分是单次$O(\sqrt{n})$的，所以这部分复杂度为$O(m\sqrt{n})$

总复杂度$O(n\sqrt{n}+m\sqrt{n})$ 由于维护了所有可能的贡献，而且更新是取min，正确性也得到了保障