问题1：长方体（cube）

思路

由于要统计的整点需要至少 $n-1$ 个长方体都覆盖到，所以可以考虑枚举不需要覆盖到的长方体，然后求剩下长方体的交，最后再求这 n 个长方体交的并即可。

首先考虑快速的求交，可以利用前缀计算的思路，先求出所有前 i 个长方体和后 j 个长方体的交，这样每一个交都可以由一个前缀交和一个后缀交再求交得到。

然后考虑求并，用扫描的方式效率也不高，仔细分析这些交的特点，考虑到它们本身就已经是 $n-1$ 个长方体的交了，所以对于任意两个交，它们重叠的部分只可能是所有 n 个长方体的交，所以只要减去 n 个长方体的交，就是只被这 $n-1$ 个长方体覆盖的部分，这样就可以做到每个交的部分只刚好统计一次了。

复杂度

时间

计算前缀、后缀交 $O(n)$ 。

枚举忽略的立方方体，求交并统计答案 $O(n)$ 。

总时间复杂度为 $O(n)$ 。

空间

记录前缀交、后缀交 $O(N)$ 。

问题2：三角形（triangle）

思路

对于每个询问，可以暴力枚举边并判断是否能够组成三角形，但这样做效率较低。

能够构成三角形的场景比较多，不妨思考不能构成三角形的场景，此时最小的两条边相加应当小于等于最大边。那么对于一个边集，可以先进行排序，这样任意相邻两条边相加都小于等于下一条边，所以只需要枚举任意三条相邻边并判断即可。

但区间的范围可能较大，这样枚举效率还是不够高，再仔细思考判断的过程，可以发现，如果一直无法组成三角形，那么边长会呈指数级增长，增长速度至少为斐波那契数列，计算可知只需要不足 100 项就会超出题目规定的取值范围，所以如果区间长度超过了 100 ，必定能够组成三角形。因此，只需要对长度不超过 100 的区间进行判定即可。

复杂度

令数的取值范围为 v 。

时间

读入序列 $O(n)$ 。

查询数量 $O(q)$ 。

- 排序 $O(log\ v\ log\ log\ v)$ 。
- 校验$O(log\ v)$ 。

总时间复杂度为 $O(n + q\ log\ v\ log\ log\ v)$ 。

空间

记录序列 $O(n)$ 。

问题3：区间（section）

思路

对任意区间进行翻转可以通过 Splay 来实现，但这样做没有利用本题中两个关键的性质：

1. 翻转操作的区间左边界不减。
2. 区间长度是固定的。

考虑上述两个条件，我们可以将问题看成是一个长度为 m 的定长区间由左向右进行滑动，并某些位置执行翻转操作，所以问题的关键就是处理滑动的过程。

因此，我们需要对当前区间内的数进行维护，如果要操作的区间在右边，则先滑动过去再操作，操作可以不用实际翻转整个区间，而是用一个翻转状态进行标记。滑动时，如果区间是未翻转状态，则头出尾进，如果是翻转状态，则尾出头进，这可以用双端队列来实现。进来的数就是原序列中下个位置的数，出去的数就回到原序列中。

查询时也需要分情况讨论，如果查询的位置在当前区间中，则根据区间是否翻转来寻找对应位置，否则直接在原序列中找到对应位置即可。

复杂度

时间

读入序列 $O(n)$ 。

查询数量 $O(q)$ ，处理查询 $O(1)$ ，总共 $O(q)$ 。

处理查询时区间滑动总共 $O(n)$ 。

总时间复杂度为 $O(n + q)$ 。

空间

记录序列 $O(n)$ 。

问题4：图（graph）

思路

首先可以假定初始时集合中有一个点 0 ，权值也为 0 ，这样实际上就只有一种操作，每次选择加入集合的点时都必须在集合中选择一个接入点。

那么不难发现，最终操作的结果集合可以用一棵树来表示， 0 点是树根，每个加入集合的点都需要选择一个点作为父亲。

所以可以尝试往生成树的方向考虑，由于每次操作都会累加父亲的权值，而每个节点都有父亲（0 点可以忽略），为了正确的累加权值，我们可以先将所有点的权值减去，并将任意两点之间边的权值设置成两点权值的总和。这样每个点在接入树的时候，它本身被减去的权值会被抵消，而父亲的权值会额外累加一次。

最后求这张图的最大生成树即可，考虑到可能是一个稠密图，建议使用 Prim 算法求解。

复杂度

时间

Prim 算法 $O(N^2)$ 。

空间

记录点 $O(N)$ 。