瘟疫（ill）

根据期望的线性性，答案 $E(t) = P_2 + P_3 + ... + P_n + 1$，其中 P_i 是第 i 个小区 在 1 号小区之前被选中的概率。

考虑 1 号和 i 号小区，可以发现其他小区都不会影响它们，因此相当于只要考虑只有这两个小区的情况，因此概率即为 $\frac{a_i}{a_1+a_i}$ 。因此答案即为 $\sum_{i=2}^{n} \frac{a_i}{a_1+a_i} + 1$，直接计算即可。时间复杂度：$O(n)$。

打怪（monster）

如果 x 固定，答案可以通过二分很容易的在 $O(n \log V)$ 的时间复杂度内算出。

设对于一个 x，对应的最优解为 $F(x)$。直接对于所有 x 都暴力计算即可得到 $O(nm \log n)$ 的复杂度，可以通过子任务 $1, 2$。

如果我们以随机顺序遍历所有 x，则在期望情况下，新的 $F(x)$ 比当前所有 F 的 值都要小的 x 个数的期望应该是 $\sum \frac{1}{i}$，这是调和级数，为 $O(\ln n)$。 于是我们只对于这些 x 二分即可：我们令答案为 $ans - 1$ 并贪心，即可判断当前 的 x 是否满足条件。

时间复杂度：$O(nm + n \ln n \log n)$。

比赛（tournament）

不知道有没有更好的解法，这里讲一个垃圾构造。

可以证明一定有解，对于任意输入，用如下方法构造：

首先比赛可以抽象成一个图。

对于一个序列 $a_1, a_2, ..., a_n$，递归下去构造 $a_n - a_{n-1}, a_n - a_{n-2}, ..., a_n - a_1$，得到一个 $a_n - a_1 + 1$ 个点的图，然后向里面加进去 a_1 个点，再取补图，就得到了一组构造。

递归到 $n = 1$ 时构造完全图即可。

稍微看一下就知道刚刚的构造可以在每一次递归时都算出合法的解：递归下去得到的图里的点度数取补图再加上 a_1 恰好就是我们要的新图里的一个度数。

时间复杂度 $O(n^2)$。

数据很水，随机应该表现很好吧（大概吧），欢迎来乱搞过题（

位运算（bit）

考虑我们如何比较两个数的大小，可以发现是按照从高位到低位的顺序依次比较。直到第一个不同的位出现，那么两个数的大小关系就是这个不同的位的大小关系。

值得一提的是，这里我们认为较短的数字已经在前面补前导 0 使得两个数字长度相同。 那么对于本题，在二进制下比较方式同样不变，因此我们可以按照从高位到低位 的顺序逐位确定，这样就可以得到最大值了。

按位与

我们按照从高位到低位的顺序进行贪心，如果当前位有至少两个数字是 1，那么我们可以让这一位是 1，因此最终答案的这一位一定是 1。

那么如果我们确定了某一位是 1，我们最后选择的两个数的这一位都得是 1，所以我们可以去掉所有这一位为 0 的数字，然后继续向下贪心即可。最后的方案数就是剩余的数字中选出两个的方案数。

令 $V = \max a_i$，时间复杂度：$O(n \log V)$。

按位异或

同样地从高到低贪心，逐位确定。但是直接确定下来某一位是 1 并不能和按位与那样减少候选范围。

考虑先确定最后选择的一个数字 a_i，考虑如何求出另一个 a_j 使得 $a_i \oplus a_j$ 最大。考虑将所有数字建出一个 trie 树，然后同样从高到低进行贪心：如果当前位可以取得和 a_i 这一位不一样，那么就取不一样的，否则取一样的。方案数可以在 trie 树 的每个节点上存储对应数字个数得到。

时间复杂度：$O(n \log V)$。

按位或

同样地在确定了一个数之后从高到低贪心。

考虑贪心到第 i 位，如何确定该位是否可以填 1：设 A_i 为 a_i 是 1 的位置的集合，S 为已经确定的高位必须是 1 的集合再并上第 i 位，再设 $B_i = \{x | x \in S \& x \notin A_i\}$，如果 $\exists (i, j), B_i \subseteq A_j$，则第 i 位可以填 1。那么我们只要能求出 $f_S = \{i | S \subseteq A_i\}$，暴力枚举所有的 B_i 即可判断了。

这个 f 其实就是**高维后缀和**。令一开始的 f_S 数组为桶，然后枚举每一位分别做后缀和即可。具体而言，就是枚举每一个 i，对于所有 $S \& (1 << i) = 0$ 的 S，令 $f_S \leftarrow f_{S | (1 << i)}$。

时间复杂度：$O((n + V) \log V)$。