opt

令 $x_1, x_2, \cdots, x_k$ 表示选择的数位，x_1 是最高位，x_k 是最低位，则概率是 $\sum(9-x_i)10^{-i}$。

这是一个字典序的形式，不断贪心的取最靠前的后缀最大值就行，

tree

一棵树上，给定一个节点 u 和一个点集 S，你需要选择 S 内的一个点 v，最大化 $dist(u,v)$。则一定存在一个 v ，满足它是点集 S 构成的虚树上的任意一条直径的两个端点之一。

根据这个事情，我们也可以实现在加点的过程中维护直径。

按 A_i 排序后一边加点一边维护直径就行。

复杂度 $O(n\log n)$。

rev

令 $B_i = |A_i - A_{i+1}|$，令 $C_i = |A_i + A_{i+1}|$。

我们可以看成有 $N-1$ 张卡牌，每张卡牌正面是 B_i，背面是 C_i，一次操作可以选择两张卡牌翻转。

考虑建图，对于每张卡牌，连有向边 $C_i \to B_i$，一次操作可以看成翻转两条边，你需要让每个点的入度 ≤ 1。

观察每个联通块，边数必须小于等于点数，否则根据鸽笼原理无解。

如果一个联通块是基环树，环上要么是顺时针要么是逆时针，确定环边的方向后剩下的树边的方向就确定了，只有两种方案。

如果一个联通块是树，枚举哪个节点没有入度，然后所有边的方向就确定了，写一个换根 dp 或者稍微用一点数据结构维护下就行。

这样，对于每个联通块，你可以求出在所有方案中，翻转最少的奇数/偶数条边，然后再做一个 dp 求出翻转最少的偶数条边就行。

不计离散化复杂度，精细实现能做到 $O(n)$。

min

分治，令当前分治区间是 $[l,r]$，$mid = \lfloor (l+r)/2 \rfloor$。考虑跨越两侧的区间对答案的贡献。

令:

- $C_i = \min\{A_i, \cdots, A_{mid}\}$
- $D_i = \min\{B_i, \cdots, B_{mid}\}$
- $E_i = \min\{A_{mid+1}, \cdots, A_i\}$
- $F_i = \min\{B_{mid+1}, \cdots, B_i\}$

则 $w(l,r) = |\min\{C_l, E_r\} - \min\{D_l, F_r\}|$。

因为 C, D 是递增的，E, F 是递减的，所以固定 $k = r-l+1$ 后，在一段前缀中，$\min\{C_l, E_r\} = C_l$，一段后缀中，$\min\{C_l, E_r\} = E_r$。

同理，在一段前缀中，$\min\{D_l, F_r\} = D_l$，一段后缀中，$\min\{D_l, F_r\} = F_r$。

分四种情况讨论：

- $w(l,r)=|C_l-D_l|$，直接求区间 $\min$ 就行。
- $w(l,r)=|E_r-F_r|$，还是直接求区间 $\min$ 就行。
- $w(l,r)=|C_l-F_r|$，C 是递增的 F 是递减的，你只需要二分出交点就行。
- $w(l,r)=|E_r-D_l|$，E 是递减的 D 是递增的，还是只需要二分出交点就行。

复杂度 $O(n\log^2 n)$。