

维护秩序

(monotone.cpp/c/pas)

时空限制：1s/256M，测试数据共 10 组

【问题描述】

“封闭系统内，总会越来越混乱” --熵增定律

世界上的很多场景在没有人为管理的情况下都会不由自主的变乱，要维护世界的秩序是需要付出代价的。现在有一个序列包含 n 个整数，第 i 个数字为 $x[i]$ ，这个序列可能是乱序的。现在你需要把 x 序列改造成另一个包含 n 个整数的序列 y ，其中第 i 个数字从 $x[i]$ 改为 $y[i]$ ，修改的代价为两者的差值 $|x[i]-y[i]|$ 。要求 y 序列必须全部不上升或者全部不下降，请问最小的修改总代价为多少？

【输入格式】输入文件 monotone.in

输入第一行为正整数 n ， $n \leq 2000$ 。接下去 n 行为 n 个正整数，代表 $x[i]$ ，均不超过 1000000000。

【输出格式】输出文件 monotone.out

输出一个整数。

【输入输出样例 1】

monotone.in	monotone.out
3 5 7 6	1

【输入输出样例 2】

monotone.in	monotone.out
4 5 1 3 2	2

【数据规模与约定】

1,2 号数据: 保证 $n \leq 3$ 。

3,4,5,6 号数据: 保证 $n \leq 100$ 。

所有数据: 保证 $n \leq 2000$ ， $x[i] \leq 1000000000$