第 23 届全国青少年信息学奥林匹克联赛模拟题解

CCF-NOIP-2018

提高组(复赛)第一试

竞赛时间：2018 年 11 月 4 日 8:00-11:30

题目名称	人生赢家	虐暴全场	锋芒毕露
题目类型	传统型	传统型	传统
目录	winner	beatall	everytime
可执行文件名	winner	beatall	everytime
输入文件名	winner.in	beatall.in	everytime.in
输出文件名	winner.out	beatall.out	everytime.out
每个测试点时限	1 秒	1.5 秒	2.5 秒
内存限制	128MB	128MB	512MB
测试点数目	10	10	10
每个测试点分值	10	10	10

提交源程序文件名

对于 pascal 语言	winner.pas	beatall.pas	everytime.pas
对于 C 语言	winner.c	beatall.c	everytime.c
对于 C++语言	winner.cpp	beatall.cpp	everytime.cpp

编译选项

对于 C 语言	-lm	-lm	-lm
对于 C++语言	-lm -O2	-lm -O2	-lm -O2

注意事项：

1、 文件名（程序名和输入输出文件名）必须使用小写。
2、 除非特殊说明，结果比较方式均为忽略行末空格及文末回车的全文比较。
3、 C/C++中函数 main()的返回值类型必须是 int，程序正常结束时的返回值必须是 0。
4、 全国统一评测时采用的机器配置为：CPU AMD Athlon(tm)II x2 240 processor，2.8GHz，内存 4G，上述时限以此配置为准。
5、 只提供 Linux 格式附加样例文件。
6、 评测在 NOI Linux 下进行。
7、 编译时不打开任何优化选项。

1. 人生赢家

(winner.pas/c/cpp)

Solution

签到题，大家肯定都秒了，显然地，可以发现，输入的a_i并没有什么用，
设第一个随机产生的数为 i，则有：

$$\mathrm{E}=\frac{1}{n}\sum_{i=0}^{n-1}(i+z[i]*E)$$

z[i]表示当 i 这个数在a_i中出现时为 0，不出现为 1，
化简：

设 $S=\frac{n*(n-1)}{2}$

$$E=\frac{S}{n}+\frac{n-m}{n}*E$$

E= $E=\frac{S}{m}$

$$=\frac{n*(n-1)}{2*m}$$

2. 虐暴全场

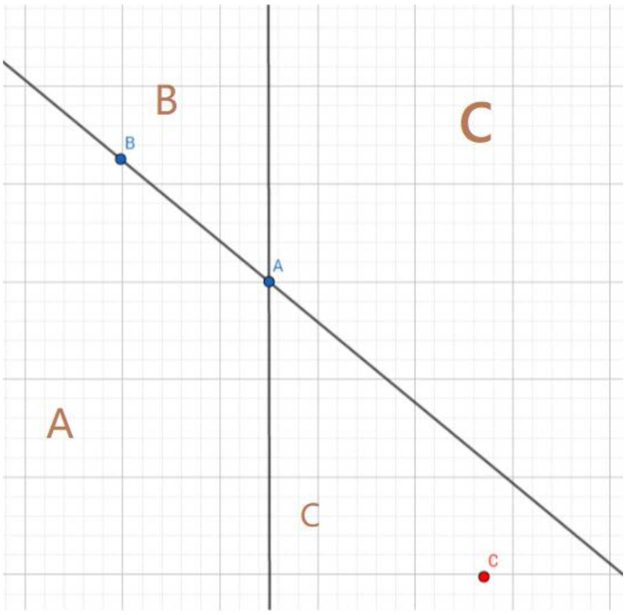
(beatall.pas/c/cpp)

考虑对于每个点的权值被哪些点影响，对于点 i，显然，它的权值只会被他前面比它高的点所影响，

如果你多试了几组数据且足够细心，你应该能发现，题目的输出序列本质上就是每个点往它左边看能看到的最高点，（连线斜率最小的点）如果它左边没有比它大的则输出它自己

证明：

设当前做到点 C，它最后被点 A 产生的直线覆盖，A 点指向 B，C 点之前的点符合上面的结论；（见下图）



把图像分成 A,B,C 三块区域进行讨论，A 区为在点 A 左侧且在直线下方区域，B 区为在点 A 左侧且在直线上方区域，C 区为点 A 右侧区域，

用反证法，显然的，如果存在点 X，他到点 C 的斜率比 A 到点 C 的斜率小，那么点 X 一定在 B 区或 C 区，

如果 X 点在 B 区，那么 A 点就一定会指向 X，所以不存在；

如果 X 点在 C 区，那么肯定是点 X 与它指向的点 X’构成直线的斜率过小，才没能覆盖到点 C，同样的 X’指向的点 X’’构成直线也应的斜率过小，没能覆盖到点 C，这样一直迭代下去，所以这种情况不存在。

所以最终的做法是用单调栈维护一些斜率递增的直线

3. 锋芒毕露

(everytime.pas/c/cpp)

算法 1：两个圆一共有四个点，暴力枚举这四个点再判是否合法，时间复杂度 $O(n^4)$，期望得分 20。

算法 2：枚举一个圆，然后求有多少个圆比它前与其相交，先枚举左端点，然后从左往右扫找可以与之匹配的右端点，在扫的同时顺便记录跨过左端点的点的不同颜色的圆的数量，如此这般便能统计出答案了。时间复杂度 $O(n^2)$，期望得分 50。

算法 3：
考虑容斥，题目要求我们求有多少对颜色不同的圆相交，我们可以先求出有多少对颜色不同的圆不相交，再用总数减去。
我们将相同颜色的点用同一个字母表示，那么便只有 $AABB$ 和 $ABBA$ 两种情况。

情况 1: $A_1A_2B_1B_2$，我们求出对于每个B_1前面有多少对A_1A_2可以与之匹配，再求出对于每个B_2在其前面可以与之匹配的B_1的贡献，时间复杂度 $O(n)$。

情况 2 : $ABBA$
我们设置一个阈值 K，将出现次数大于 K 的颜色记为 L，小于等于 K 的颜色记为 P。
那么现在又有四种情况 $LPPL$, $PLLP$, $P_1P_2P_2P_1$, $L_1L_2L_2L_1$。

- $LPPL$ 的情况，对于每个 L，枚举每个 P，记第 i 个 P 前面出现了b_i个 L，后面出现了c_i个 L，那么第 i 个 P 对答案的贡献显然为$\sum_{j=1}^{i-1} b_j \times c_i$，这个式子显然可以 $O(1)$计算，总的时间复杂度 $O(\frac{n^2}{K})$。
- $*LL*$的情况，对于每个 L，枚举每个$*$，记第 i 个$*$前面出现了a_i个 L，那么第第 i 个$*$的贡献显然为$\sum_{j=1}^{i-1}\binom{a_i-a_j}{2}$，这个东西只需维护$\sum_{j=1}^{i-1} a_j^2$和$\sum_{j=1}^{i-1} a_j$就可以 $O(1)$计算，总的时间复杂度 $O(\frac{n^2}{K})$。
- $P_1P_2P_2P_1$的情况，现在我们需要计算对于每一对$P_1(i,j)$，有多少对$P_2(l,r)$，满足 $i < l < r < j$，现在从左往右枚举 j，接着 i，统计有多少对合法的(l,r)就是在统计有多少个 l 满足 $i < l$，这个可以用树状数组快速求，统计完答案后在树状数组 i 位置$+1$ 即可。时间复杂度 $O(n\,K\log n)$。

不难发现，当 $K = \sqrt{\frac{n}{\log n}}$，总的复杂度最小，为 $O(n\sqrt{n\log n})$，鉴于树状数组常数小，K 可以取大一点，期望得分 100.