

NOI 2022 模拟赛

NOI 2022 Simulation

cdw

2022 年 5 月 13 日

题目名称	卡牌游戏	最小圆覆盖	计算几何
题目类型	交互型	传统型	传统型
提交源程序文件名	card.cpp	circle.cpp	geometry.cpp
输入文件名		circle.in	geometry.in
输出文件名		circle.out	geometry.out
每个测试点时限	1.0 秒	4.0 秒	2.0 秒
内存限制	256 MiB	256 MiB	256 MiB
子任务数目	9	20	20
子任务是否等分	否	是	是

【注意事项（请仔细阅读）】

1. 选手提交的源程序请**直接放在个人目录下**，无需建立子文件夹；
2. 文件名（包括程序名和输入输出文件名）必须使用英文小写。
3. C++ 中函数 `main()` 的返回值类型必须是 `int`，值必须为 0。
4. **对于因未遵守以上规则对成绩造成的影响，相关申诉不予受理。**
5. 若无特殊说明，结果比较方式为**忽略行末空格、文末回车后的全文比较**。
6. 程序可使用的栈空间大小与该题内存空间限制一致。
7. 若无特殊说明，每道题的**代码大小限制为 100KB**。
8. 若无特殊说明，输入与输出中同一行的相邻整数、字符串等均使用一个空格分隔。
9. 直接复制 PDF 题面中的跨页样例，数据将带有页眉页脚，建议选手直接使用对应目录下的样例文件进行测试。
10. 使用 `std::deque` 等 STL 容器时，请注意其内存空间消耗。
11. 评测时使用的系统配置为 Intel(R) Core(TM) i7-9700F CPU @ 3.00GHz，内存 16.0 GB，操作系统 64 位 Windows 10。上述时空限制以此配置为准。
12. 评测时使用的编译参数为 **`-lm -O2 -std=c++14`**，保证你的程序在本机能够通过编译。此外**不允许在程序中手动开启其他编译选项**，一经发现，本题成绩以 0 分处理。

卡牌游戏 (card)

【题目描述】

这是一道交互题。

给定 $2n$ 张卡牌，编号为 $1 \sim 2n$ ，卡牌的类型有 n 种，每种分别有 2 张。

维护一个初始为空的卡牌集合 S ，每次询问时，你可以在 S 中加入或删除任意一张卡牌，交互库会告诉你 S 中有多少种类型的卡牌。

你至多可以询问 10^6 次，要求给相同类型的卡牌配对。

【实现细节】

你需要提交一个程序 `card.cpp`，将头文件 `card.h` 的内容复制到开头，并实现以下函数：

```
void Solve(int N);
```

在每个测试点中，该函数会在程序运行时被调用恰好一次。 N 表示题目描述中的 n 。

你的程序可以调用以下函数：

```
int Query(int x);
```

你至多能调用该函数 10^6 次。你需要保证 $1 \leq x \leq 2n$ 。若 $x \in S$ 则将卡牌 x 从 S 中删去，否则在 S 中加入卡牌 x ，然后交互库返回 S 中的卡牌类型数。

```
void Answer(int a, int b);
```

表示给出一对相同类型的卡牌 a 和 b 。

你需要保证每张卡牌都被给出恰好一次，从而恰好调用此函数 n 次。

【样例交互库】

下发文件中包含了样例交互库 `grader.cpp`，该交互库可以帮助你理解这道题目的题意并测试你的程序。在最终评测时，所使用的交互库与该样例交互库有所不同，你的程序不应依赖该样例交互库的实现。

你可以使用以下命令编译出可执行文件 `card`：

```
g++ grader.cpp card.cpp -o card -lm -O2 -std=c++14
```

可执行文件 `card` 将从标准输入中读入以下格式的输入数据：

- 第一行，一个正整数 n 。
- 之后 n 行，每行两个正整数 x_i, y_i ，表示卡牌 x_i 与 y_i 的类型相同。

如果你给出的答案正确，可执行文件会输出 **Accepted**：之后一个非负整数表示询问次数。

【测试点约束】

对于所有数据， $1 \leq n \leq 43\,000$ ， $1 \leq x_i, y_i \leq 2n$ ， x_i, y_i 两两不同。

子任务编号	分值	$n \leq$	特殊性质
1	6	100	A
2	25	15 000	
3	9		
4	30	38 000	
5	5	39 000	
6	5	40 000	
7	5	41 000	
8	5	42 000	
9	10		

特殊性质 A: $1 \leq x_i \leq n$ ， $n + 1 \leq y_i \leq 2n$ 。

最小圆覆盖 (circle)

【题目描述】

已知二维平面上 n 个整点 $A_i(x_i, y_i)$ ，这些点均由下述代码中调用 `init` 函数之后的 `gen` 函数生成，其中参数 n 和 sd 将由输入数据给出：

```
#include<bits/stdc++.h>
const int N = 50003, V = 2e8;
std::mt19937 rng;
void init(unsigned sd){rng.seed(sd);}
int x[N], y[N];
void gen(int n){
    for(int i = 1; i <= n; ++ i){
        x[i] = rng() % V; y[i] = rng() % V;
    }
}
```

给定正整数 l, k ，求最小的实数 r ，使得存在长为 l 的子段 A_i, \dots, A_{i+l-1} 和一个半径为 r 的圆，满足这个圆覆盖这个子段中至少 k 个点。

你需要在一个测试点中处理多组数据的询问。第 i 组数据由第 i 次调用 `gen` 函数生成。

【输入格式】

从文件 `circle.in` 中读入数据。

第一行，一个正整数 T 表示数据组数，以及一个非负整数 sd 。

之后对于每组数据：一行，三个正整数 n, l, k 。

【输出格式】

输出到文件 `circle.out` 中。

对于每组数据，输出一行，一个非负实数，表示答案。

当你的答案与标准答案的绝对误差或相对误差不超过 10^{-9} 时即算正确，保证标准答案与正确答案的绝对误差或相对误差不超过 10^{-15} 。

【样例输入】

```
6 1145141919
30 25 20
100 50 30
500 111 30
5000 998 30
15000 5555 30
29370 1111 30
```

【样例输出】

```
87210430.941034281422617
73372921.223518104998220
39377120.594243174436997
12405795.624578265473247
4872004.657048527696134
10517085.666085580529398
```

【测试点约束】

对于所有数据, $T \leq 10^3$, $\sum n \leq 50\,000$, $k \leq 30$, $2 \leq k \leq l \leq n$, sd 在 $[0, 2^{32} - 1]$ 中独立均匀随机生成。

测试点编号	特殊性质
1 ~ 3	$T \leq 5, n \leq 30$
4 ~ 6	$T \leq 3, n \leq 100$
7 ~ 9	$T \leq 3, n \leq 500$
10 ~ 12	$\sum n \leq 5\,000, k \leq 20$
13 ~ 16	$\sum n \leq 20\,000, k \leq 20$
17 ~ 20	

计算几何 (geometry)

【题目描述】

这场模拟赛有两道计算几何题，怎么会是呢？



emu 酱认为平面上的一棵树是**可爱的**当且仅当：

- 每个节点有一个坐标 $A_i(x_i, y_i)$ ，两两不同且没有三点共线。
- 对于两条边 (u_1, v_1) 和 (u_2, v_2) ，若 $u_1 \neq u_2$ 、 $u_1 \neq v_2$ 、 $v_1 \neq u_2$ 且 $v_1 \neq v_2$ ，则线段 $A_{u_1}A_{v_1}$ 与线段 $A_{u_2}A_{v_2}$ 不相交。

平面上的一棵树是**可怕的**当且仅当凸包顶点**恰好**为所有叶子，叶子是指度数 ≤ 1 的节点。特别地，节点数 ≤ 2 的树都是可怕的。

给定一棵 n 个点的可爱树，emu 酱想知道有多少种将其划分为连通子树的方案，使得每个连通子树都是可怕的。两种方案不同当且仅当存在两个节点，使得其在一种方案中被划分到同一子树，而在另一种方案中被划分到不同子树。

由于答案可能很大，你只需求答案对 998 244 353 取模的结果。

【输入格式】

从文件 *geometry.in* 中读入数据。

第一行，一个正整数 n 。

之后 n 行，每行两个整数 x_i, y_i ，表示第 i 个点的坐标。

之后 $n - 1$ 行，每行两个正整数 u, v ，表示树的一条边 (u, v) 。

【输出格式】

输出到文件 *geometry.out* 中。

一行，一个非负整数，表示答案。

【样例 1 输入】

```
4
0 0
0 1
-1 -1
1 -1
1 2
1 3
1 4
```

【样例 1 输出】

```
5
```

【样例 2 输入】

```
6
4 -5
0 1
-2 8
3 -10
0 -1
-4 -5
2 5
3 2
1 2
4 6
4 2
```

【样例 2 输出】

```
13
```

【样例 3】

见选手目录下的 *geometry3.in* 和 *geometry3.ans*。

【测试点约束】

对于所有数据， $n \leq 300$ ，直径长度 ≤ 14 ， $|x_i|, |y_i| \leq 2^{31} - 1$ 。

测试点编号	$n =$	测试点编号	$n =$
1	10	9	70
2	15	10	80
3	20	11	100
4	25	12	120
5	30	13 ~ 14	150
6	40	15 ~ 16	200
7	50	17 ~ 18	250
8	60	19 ~ 20	300