NOIP2022 模拟题

A. 报数

num.cpp/.in/.out 1S 1G

题目描述

作为 NOIP 的第一题，需要一道签到题。

有 n 个数 $a_1, a_2, \ldots, a_n$，你需要找到一个集合 S，使得 S 中**严格**大于 S 的平均数的数字个数尽量多。输出最多的个数。

注意：这里的集合是可重集，数字可以重复，但每个数字只能选一次。

输入格式

第一行一个整数 n。

接下来一行 n 个整数 $a_1, a_2, \ldots, a_n$。

输出格式

一个整数，表示答案。

样例输入1

```
5
1 2 3 4 5
```

样例输出1

```
2
```

样例输入输出2

见下发文件。

数据范围

一共 10 个测试点。

对于测试点 $1, 2$，保证 $n \le 20$。

对于测试点 $3, 4, 5$，保证 $n \le 2000$。

对于所有测试点，保证 $n \le 10^6, 1 \le a_i \le 10^9$。

B. 数列

tree.cpp/.in/.out 2S 128M

题目描述

作为 NOIP 的第二题，需要一道简单的dp题。

你有个 n 个点 m 条边的无向图，每条边都有红蓝两种颜色中的一种，保证红色的边形成了这个图的一个生成树。

你希望给这些边赋上边权，保证边权是 $1 \sim m$ 的排列，使得红色的边是最小生成树。

希望这些边权形成的序列字典序最小，也就是先比较第一条边的边权，再比较第二条边的边权，依次类推。

提示：注意内存限制。

输入格式

第一行两个个整数 n, m。

接下来 m 行，每行三个整数 u_i, v_i, c_i，c_i 表示颜色，其中 1 表示红色，0 表示蓝色。

保证没有自环，可能有重边，保证 $c_i = 1$ 形成了生成树。

输出格式

一共 m 个整数，依次表示每条边的边权。

样例输入1

```
4 5
1 2 0
2 3 1
3 4 1
2 4 0
1 3 1
```

样例输出1

```
3 1 4 5 2
```

样例输入输出2

见下发文件。

数据范围

一共 10 个测试点。

对于测试点 1，保证 $n,m\le 10$。

对于测试点 2，保证 $n,m\le 20$。

对于测试点 $3,4,5$，保证 $n,m\le 1000$。

对于测试点 $6,7$，保证 $n,m\le 10^5$。

对于所有数据，保证 $n,m\le 5\times 10^5,m\ge n-1$。

提示：注意内存限制。

C. 方差

interval.cpp/.in/.out 1S 1G

题目描述

作为 NOIP 的第三题，需要一个模拟退火题。

现在有 n 个区间 $[l_i, r_i]$，每个区间有个权值 w_i。我们把这 n 个区间当成 n 个点，如果两个区间它们之间有交（包括端点），那么我们就在这两个区间之间连边，形成了一个区间图。

现在希望你删除一些区间，使得每个连通块大小不超过 k。输出删除区间最小的权值和。

输入格式

第一行两个整数 n, k。

接下来 n 行，每行三个整数 l_i, r_i, w_i。

输出格式

一个整数，表示答案。

样例输入1

```
5 2
1 4 1
3 6 2
5 8 5
7 10 2
9 12 1
```

样例输出1

```
3
```

数据范围

一共 10 个测试点。

对于测试点 $1, 2$，保证 $n \le 20$。

对于测试点 $3, 4$，保证 $n \le 100$。

对于测试点 $5, 6$，保证 $n \le 500$。

对于所有测试点，保证 $1 \le k \le n \le 2500, 1 \le l_i \le r_i \le 10^9, 1 \le w_i \le 10^9$。

D. 棋局

rectangle.cpp/.in/.out 4S 1G

题目描述

作为 NOIP 的第四题，需要一个纯纯的屑题。

你有个 n 个矩形，每个矩形四条边平行于坐标轴。对于一个矩形，它左下角坐标为 (x_1, y_1)，右上角坐标为 (x_2, y_2)，包含了所有满足 $x_1 \le x \le x_2, y_1 \le y \le y_2$ 的点 (x, y)。

接下来你要对这些矩形进行 m 次移动操作。每次移动会选择一个矩形，具体的移动可以用方向和距离表示。方向分别为，上、下、左、右、左上、左下、右上、右下，具体会在输入格式中按方向向量给出。距离为一个正整数 d。假设矩形的左下角坐标为 (a, b)，方向向量为 (dx, dy)，那么会移动到$(a + d \cdot dx, b + d \cdot dy)$位置。同时，这个矩形在移动的过程中，会把中间过程都保留下来。也就是说，矩形移动到$(a + d \cdot dx, b + d \cdot dy)$位置，那么会产生 d 个新的矩形，左下角为 $(a + i \cdot dx, b + i \cdot dy)$ 其中 $i = 0 \ldots d - 1$，大小和原矩形相同。这些矩形生成之后就不会再移动了。

m 次移动操作之后，有 q 个询问，每次给你一个点 (px, py)，问有多少个矩形会包含这个点。

输入格式

第一行三个整数 n, m, q。

接下来 n 行，每行四个整数 x_1, y_1, x_2, y_2，表示矩形的坐标。

接下来 m 行，每行三个整数 $v_i (0 \le v_i \le 7), r_i, d_i$，表示方向，矩形的编号（从 1 开始）和移动的距离。

其中方向从 0 到 7 分别表示：$(1, 0), (1, 1), (0, 1), (-1, 1), (-1, 0), (-1, -1), (0, -1), (1, -1)$。

接下来 q 行，每行两个整数 px, py 表示查询点的坐标。

输出格式

一共 q 行，表示答案。

样例输入1

```
1 8 3
2 1 2 1
0 1 1
1 1 1
2 1 1
3 1 1
4 1 1
5 1 1
6 1 1
7 1 1
1 1
2 1
4 2
```

样例输入1

```
0
2
1
```

样例输入输出2

见下发文件，这个点与测试点 $2, 3$ 的限制相同。

样例输入输出3

见下发文件，这个点与测试点 $5, 6$ 的限制相同。

数据范围

一共 10 个测试点。

对于测试点 1，保证 $n, m, q \leq 100$，坐标范围 1 到 100 之间。

对于测试点 $2, 3$，保证 $n, m, q \leq 1000$。

对于测试点 4，保证 $m = 0$。

对于测试点 $5, 6$，保证 $v_i \in \{0, 2, 4, 6\}$。

对于测试点 $7, 8$，保证坐标范围 1 到 1000 之间。

对于所有测试点，保证 $n, m, q \leq 250000$，坐标范围 1 到 250000 之间。

这里的坐标范围指矩形的在**所有**时间的坐标都在这个范围内，并且查询的点也在这个范围内。