

CSP2022模拟赛

8:00--12:00

提交CWOI

无需文件操作

A. 最后一课

题目描述

姬子正要去找 Kiana，但在这之前，她还需要去一个地方。

在平面直角坐标系上，有一条直线 $y = k$ ，还有两点 $P(x_1, y_1), Q(x_2, y_2)$ 。姬子在点 P 处，Kiana 在点 Q 处。姬子希望先走到直线 $y = k$ 上，然后再去找 Kiana。求姬子走到 Kiana 处所需要走的最小距离的平方。

换句话说，你需要在这条直线上找到一点 R ，使得 $|PR| + |QR|$ 尽可能小，也就是最小化从 P 到 R 再到 Q 的距离。为了方便检验答案，你只需要输出这个值的平方，可以证明这个值平方后总是整数。

输入格式

一行五个整数 k, x_1, y_1, x_2, y_2 ，含义如题面所述。

输出格式

一行一个自然数，表示平方后的答案。

样例输入 1

```
2 1 1 3 3
```

样例输出 1

```
8
```

样例输入 2

```
0 1 1 3 3
```

样例输出 2

```
20
```

样例输入 3

```
0 -1000000000 -1000000000 -1000000000 1000000000
```

样例输出 3

```
4000000000000000000
```

样例输入 4

```
2 1 2 3 2
```

样例输出 4

```
4
```

更多样例

没有。

数据范围

- 对于 30% 的数据 $k = y_1 = y_2$
- 对于额外 30% 的数据 $|k|, |x_1|, |y_1|, |x_2|, |y_2| \leq 20$
- 对于 100% 的数据 $|k|, |x_1|, |y_1|, |x_2|, |y_2| \leq 5 \times 10^8$

B. 日常管理

题目描述

Kiana 正在基地里打靶。在一条长度为 m 的线段上，有 n 个靶子，第 i 个靶子的覆盖了 $[l_i, r_i]$ 这一段区间，且靶子之间不存在交（注意交为一个点也算有交）。对于第 i 个靶子，其中的 $[x_i, y_i]$ 这一段区间被标成了红色。

接下来，Kiana 进行了 k 次射击，会发射一发子弹打在线段的某个位置。对于每次射击，基地的人工智能--你需要输出对应的结果。具体的，假如射到了已经射过的靶子，输出 `Again`，假如没有射在靶子上，输出 `Failed`。假如不符合以上两条，且打到了红色区域，则输出 `Perfect`，否则输出 `Normal`。

输入格式

- 第一行三个自然数 n, m, k ，含义如题面所述。
- 接下来 n 行,每行四个自然数 l_i, x_i, y_i, r_i ，描述了靶子的信息。
- 接下来一行 k 个自然数，表示每次射击的坐标。

输出格式

对于每个询问，输出一行一个对应的结果。

样例输入 1

```
2 14 4
1 2 3 4
10 11 12 14
11 4 5 14
```

样例输出 1

```
Perfect
Normal
Failed
Again
```

更多样例

见下发文件。

数据范围

- 对于 30% 的数据， $n, k \leq 1000$
- 对于额外 20% 的数据， $m \leq 10^6, \sum r_i - l_i \leq 10^6$
- 对于额外 30% 的数据， $l_i = x_i, y_i = r_i$
- 对于 100% 的数据， $1 \leq n, k \leq 10^5, 1 \leq l_i \leq x_i \leq y_i \leq r_i \leq m \leq 10^9$ 。

C. 渡尘

题目描述

识之律者修改了你的意识，让你认为这场题目非常毒瘤。

在面对识之律者的战斗中，Kiana 和 Bronya 的意识都遭到了干扰，受到了识之律者制造的幻象影响。为了排除识之律者的干扰，她们决定采用这样的方式来排除识之律者的干扰：将她们的所感受到的场景抽象为一个正整数序列，然后每次取出一个子区间，并求出这个子区间内的**绝对值最大子段和**的值，并将这个值与符华所看到的值进行比对。但非常不幸的是，Kiana 的智商只有草履虫级别，于是她就把这个问题交给了你。

简化题意：你有一个长度为 n 的整数序列 A 。你需要支持以下询问：

查询 $[l, r]$ 中满足 $l \leq l_1 \leq r_1 \leq r$ 的最大的 $f(l_1, r_1)$ 的值。其中 $f(l_1, r_1) = |\sum_{i=l_1}^{r_1} A_i|$ 。

输入格式

第一行两个自然数 n, m 。

接下来一行 n 个整数，表示序列的初始值。

接下来 m 行，每行两个自然数，描述了一个查询。

本题的读入/输出量较大，请使用较快的读入/输出方式！

输出格式

对于每个询问，输出一行一个自然数，表示答案。

样例输入 1

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

样例输出 1

```
2
5
4
```

样例解释 1

三问选择的区间分别是 $[2, 2], [2, 3], [4, 4]$ 。

更多样例

见下发文件。

数据范围

对于 10% 的数据 $n, m \leq 50$

对于 20% 的数据 $n, m \leq 500$

对于 40% 的数据 $n, m \leq 5000$

对于 70% 的数据 $n, m \leq 10^5$

对于 100% 的数据 $n \leq 2 \times 10^5, m \leq 3 \times 10^6, |A_i| \leq 10^9$ 。

快读板子

下面的快读板子来自 [dysyn1314](#)，关注 dysyn1314，顿顿解馋！

用法：把这段代码复制到你的代码前面，然后用cin/cout正常写，就会变成快读、快输了。

```
/* ----- fast io ----- */ // begin
namespace Fread {
const int SIZE = 1 << 21;
char buf[SIZE], *S, *T;
inline char getchar() {
    if (S == T) {
        T = (S = buf) + fread(buf, 1, SIZE, stdin);
        if (S == T) return '\n';
    }
    return *S++;
}
} // namespace Fread
namespace Fwrite {
const int SIZE = 1 << 21;
char buf[SIZE], *S = buf, *T = buf + SIZE;
inline void flush() {
    fwrite(buf, 1, S - buf, stdout);
    S = buf;
}
inline void putchar(char c) {
    *S++ = c;
    if (S == T) flush();
}
struct NTR {
    ~ NTR() { flush(); }
} ztr;
} // namespace Fwrite
#ifdef ONLINE_JUDGE
#define getchar Fread :: getchar
#define putchar Fwrite :: putchar
#endif
namespace Fastio {
struct Reader {
    template<typename T>
    Reader& operator >> (T& x) {
        char c = getchar();
        T f = 1;
        while (c < '0' || c > '9') {
            if (c == '-') f = -1;
            c = getchar();
        }
        x = 0;
        while (c >= '0' && c <= '9') {
            x = x * 10 + (c - '0');
        }
    }
};
}
```

```

        c = getchar();
    }
    x *= f;
    return *this;
}

Reader& operator >> (char& c) {
    c = getchar();
    while (c == ' ' || c == '\n') c = getchar();
    return *this;
}

Reader& operator >> (char* str) {
    int len = 0;
    char c = getchar();
    while (c == ' ' || c == '\n') c = getchar();
    while (c != ' ' && c != '\n' && c != '\r') { // \r\n in windows
        str[len++] = c;
        c = getchar();
    }
    str[len] = '\0';
    return *this;
}

Reader(){}
} cin;

const char endl = '\n';

struct Writer {
    template<typename T>
    Writer& operator << (T x) {
        if (x == 0) { putchar('0'); return *this; }
        if (x < 0) { putchar('-'); x = -x; }
        static int sta[45];
        int top = 0;
        while (x) { sta[++top] = x % 10; x /= 10; }
        while (top) { putchar(sta[top] + '0'); --top; }
        return *this;
    }

    Writer& operator << (char c) {
        putchar(c);
        return *this;
    }

    Writer& operator << (char* str) {
        int cur = 0;
        while (str[cur]) putchar(str[cur++]);
        return *this;
    }

    Writer& operator << (const char* str) {
        int cur = 0;
        while (str[cur]) putchar(str[cur++]);
        return *this;
    }
}

```

```

    Writer(){}
} cout;
} // namespace Fastio
#define cin Fastio :: cin
#define cout Fastio :: cout
#define endl Fastio :: endl
/* ----- fast io ----- */ // end

```

D. 罪人挽歌

题目描述

Kiana 正在面对芽衣。

Kiana 有 n 个技能，每个技能都有两个参数 (A_i, B_i) 。她希望以某个顺序释放这些技能，来最大化对芽衣的效果。具体的，为了技能之间的连贯性，她希望对于任意连续的两个技能，要么 A_i 相同，要么 B_i 相同。然而对手是芽衣，最熟悉她的人。因此，假如她释放的连续三个技能的 A 或 B 都相同，她的攻击就会被看穿，而她也会落入下风。而为了尽可能的提升自己技能的效果，她还希望自己释放的技能是字典序最小的（在输入中先出现的二元组字典序更小）。你需要判断，Kiana 是否有一种合法的释放技能的顺序。如果有，你还需要给出字典序最小的一种。

简化题面：你有 n 个二元组，第 i 个二元组的值为 (a, b) 。任意两个二元组都不相同。

求是否存在一个二元组的排列，使得这些二元组满足对于任意 $1 \leq i < n$ ，有 $A_i = A_{i+1}$ 或 $B_i = B_{i+1}$ 。且对于任意 $2 \leq i < n$ ， $A_{i-1} = A_i = A_{i+1}$ 和 $B_{i-1} = B_i = B_{i+1}$ 都不成立。如果有解，你还需要找到字典序最小的解。我们认为一个二元组的字典序更小，当且仅当这个二元组在输入中更早出现。

输入格式

第一行一个自然数 n 。

接下来 n 行，每行两个自然数 a, b

输出格式

第一行一个字符串 `Yes` 或 `No`，表示是否存在方案。

如果存在方案，接下来一行 n 个自然数，第 i 个自然数表示第 i 次所释放的技能。假如你的方案的字典序不是最小，但是符合条件，你可以获得 50% 的分数。

样例输入 1

```

5
1 2
1 3
2 4
2 3
3 4

```

样例输出 1

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

样例输入 2

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

样例输出 2

```
No
```

更多样例

见下发文件。

数据范围

本题采用捆绑测试，你在一个子任务得到的分数是这个子任务内所有测试点的最小值。

subtask1(20pts) : $n \leq 10$ 。

subtask2(30pts) : $n \leq 1000$ 。

subtask3(30pts) : 保证 $A_i \leq 10$ 。

subtask4(20pts) : 无特殊限制。

对于 100% 的数据， $1 \leq a, b \leq n, 2 \leq n \leq 5 \times 10^5$ ，保证不存在两张完全一样的二元组。