

memseto's Notebook

方知蓦然回首之时
那人却已不在灯火阑珊处

关于我
友情链接
文章聚合

Theme **Ringo** by **memseto**
Proudly powered by **Typecho**

CF538G Berserk Robot

2019-03-02 | 题解

本题中我们可以对机器人进行四种操作：

- U , 即向量 $(0, 1)$
- D , 即向量 $(0, -1)$
- L , 即向量 $(1, 0)$
- R , 即向量 $(0, -1)$

考虑正解，容易发现我们需要分开考虑 x 和 y 轴的变化，然而上述四种操作对 x 或 y 轴而言有 $1 / 0 / -1$ 三种情况，较难考虑，且一边为 1 或 -1 时，另一边只能为 0 ，较难判断。

于是有一个经典的 trick 就是旋转坐标轴。

考虑以 $(\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2})$ 和 $(-\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2})$ 为基底，考虑原来的四种操作：

- U 变为向量 $(1, 1)$
- D 变为向量 $(-1, -1)$
- L 变为向量 $(-1, 1)$
- R 变为向量 $(1, -1)$

那么就可以分开考虑 x 和 y 轴的 ± 1 变化。



memset0's Notebook

方知蓦然回首之时
那人却已不在灯火阑珊处

关于我
友情链接
文章聚合

Theme **Ringo** by **memset0**
Proudly powered by **Typecho**

假设最后所有向量的和为 \vec{S} ，则题目给出的坐标在时间模 L 后即可转换为一个关于 \vec{S} 的一次函数。列出不等式并判断下奇偶性即可。

代码：

```
// =====
//   author: memset0
//   date: 2019.03.01 15:42:34
//   website: https://memset0.cn/
// =====
#include <bits/stdc++.h>
#define ll long long
#define int long long
namespace ringo {
template <class T> inline void read(T &x) {
    x = 0; register char c = getchar(); register bool f = 0;
    while (!isdigit(c)) f ^= c == '-', c = getchar();
    while (isdigit(c)) x = x * 10 + c - '0', c = getchar();
    if (f) x = -x;
}
template <class T> inline void print(T x) {
    if (x < 0) putchar('-'), x = -x;
    if (x > 9) print(x / 10);
    putchar('0' + x % 10);
}
template <class T> inline void print(T x, char c) { print(x), putchar(c); }

const int N = 2e5 + 10, M = 2e6 + 10;
int n, L, sum, det_k, det_b, det_t, flag_1, flag_2, flag_sum, max_sum, min_su
int ans[2][M << 1];

struct info {
    int k, t, x, y, b;
    inline bool operator < (const info &other) const {
        return t < other.t;
    }
};
```



memseto's Notebook

方知蓦然回首之时
那人却已不在灯火阑珊处

关于我
友情链接
文章聚合

Theme [Ringo](#) by [memseto](#)
Proudly powered by [Typecho](#)

```

    }
} a[N];

void no_answer(int flag=0) { puts("NO"), exit(0); }
inline int get_flag(int x) { x %= 2; return x == -1 ? 1 : x; }

void main() {
    read(n), read(L);
    for (int i = 1, t, x, y; i <= n; i++) {
        read(t), read(x), read(y);
        a[i] = (info){0, t, x + y, y - x};
    }
    for (int i = 1; i <= n; i++) a[i].k = -a[i].t / L, a[i].t %= L;
    a[++n] = (info){0, 0, 0, 0}, a[++n] = (info){1, L, 0, 0}, std::sort(a + 1
    for (int cho = 0; cho < 2; cho++) {
        flag_sum = -1, min_sum = -L - 1, max_sum = L + 1;
        for (int i = 1; i <= n; i++) a[i].b = cho ? a[i].y : a[i].x;
        for (int i = 1; i < n; i++) {
            det_k = a[i + 1].k - a[i].k;
            det_b = a[i + 1].b - a[i].b;
            det_t = a[i + 1].t - a[i].t;
// 判断奇偶性
            flag_1 = get_flag(det_t - det_b);
            flag_2 = get_flag(det_k);
            if (flag_1 && flag_2) {
                if (!flag_sum) no_answer();
                flag_sum = 1;
            } else if (!flag_1 && flag_2) {
                if (flag_sum == 1) no_answer();
                flag_sum = 0;
            } else if (flag_1 && !flag_2) {
                no_answer();
            }
        }
// 约束答案
        if (det_k > 0) {
            max_sum = std::min(max_sum, (int)floor((det_t - det_b) / (lon
            min_sum = std::max(min_sum, (int)ceil((-det_t - det_b) / (lon

```



memseto's Notebook

方知蓦然回首之时
那人却已不在灯火阑珊处

关于我
友情链接
文章聚合

Theme Ringo by memseto
Proudly powered by Typecho

CF538G Berserk Robot - memseto 的博客

```

} else if (det_k < 0) {
    max_sum = std::min(max_sum, (int)floor((det_t + det_b) / (lon
    min_sum = std::max(min_sum, (int)ceil((-det_t + det_b) / (lon
} else if (!(-det_t <= det_b && det_b <= det_t)) {
    no_answer();
}
if (min_sum > max_sum) no_answer(1);
}
sum = min_sum; if (~flag_sum && get_flag(sum) != flag_sum) ++sum;
if (sum > max_sum) no_answer();
for (int i = 1; i <= n; i++) a[i].b = a[i].b + a[i].k * sum;
for (int i = 1; i < n; i++) {
    det_b = a[i + 1].b - a[i].b;
    det_t = a[i + 1].t - a[i].t;
    if (det_b < 0) {
        for (int j = a[i].t; j < a[i].t - det_b; j++)
            ans[cho][j] = -1;
    } else if (det_b > 0) {
        for (int j = a[i].t; j < a[i].t + det_b; j++)
            ans[cho][j] = 1;
    }
    for (int j = a[i].t + abs(det_b); j < a[i + 1].t; j++)
        ans[cho][j] = j & 1 ? 1 : -1;
}
}
for (int i = 0; i < L; i++) {
    if (ans[0][i] == 1 && ans[1][i] == 1) putchar('U');
    if (ans[0][i] == -1 && ans[1][i] == -1) putchar('D');
    if (ans[0][i] == 1 && ans[1][i] == -1) putchar('R');
    if (ans[0][i] == -1 && ans[1][i] == 1) putchar('L');
} putchar('\n');
}

} signed main() { return ringo::main(), 0; }

```



memset0's Notebook

方知蓦然回首之时
那人却已不在灯火阑珊处

关于我
友情链接
文章聚合

Theme **Ringo** by **memset0**
Proudly powered by **Typecho**

巧妙的思路

已有 5 条评论



Itst

March 14th, 2019 at 07:58 pm

这个题484 CF538G啊QAQ

回复



memset0

March 14th, 2019 at 08:02 pm

4滴啊 QAQ

回复



memset0

March 14th, 2019 at 08:04 pm

woc 貌似交 CF 上挂了 QAQ

回复



CMXRYNP

March 4th, 2019 at 05:29 pm

老虎大蒜保护协会

回复



memset0's Notebook

方知蓦然回首之时
那人却已不在灯火阑珊处

[关于我](#)
[友情链接](#)
[文章聚合](#)

Theme [Ringo](#) by [memset0](#)
Proudly powered by [Typecho](#)



memset0

March 11th, 2019 at 03:17 pm

您大蒜保护协会

[回复](#)

用户名
邮箱
网址（选填）
可以在这里写评论哦 ~
提交评论

[如何在家中访问内网 OJ ?](#)
上一篇 «

[洛谷2496 \[SDOI2012\]体育课](#)
» 下一篇



memseto's Notebook

方知蓦然回首之时
那人却已不在灯火阑珊处

© 2017 - 2019 memset0 的博客.

浙ICP备19006255号-1

97714 visits · 24757 visitors · 74.48 W words

在这里输入关键字哦 ~ (回车搜索)

关于我

友情链接

文章聚合

Theme **Ringo** by memseto

Proudly powered by **Typecho**

