



棋局

河南省实验中学
信息技术组

注意事项

题目背景

题目描述

输入格式

输出格式

输入输出样例

数据范围

算法

子任务 1

子任务 2

子任务 3

子任务 4

小结

致谢

棋局

NOIP 2021 Task4

河南省实验中学信息技术组

2022 年 01 月 26 日



注意事项

棋局

河南省实验中学
信息技术组

注意事项

题目背景

题目描述

输入格式

输出格式

输入输出样例

数据范围

算法

子任务 1

子任务 2

子任务 3

子任务 4

小结

致谢

- 输入输出文件: `chess.in/out`
- 时间限制: `4s`
- 内存限制: `1024MiB`
- 编译命令: `g++ -O2 -lm chess.cpp -o chess`



河南省得分情况

棋局

河南省实验中学
信息技术组

注意事项

题目背景

题目描述

输入格式

输出格式

输入输出样例

数据范围

算法

子任务 1

子任务 2

子任务 3

子任务 4

小结

致谢

- 100 分：0 人 (0.00%)
- 44 分：1 人 (0.59%)
- 32 分：2 人 (1.19%)
- 24 分：4 人 (2.38%)
- 16 分：2 人 (1.19%)
- 4 分以上：10 人 (5.95%)



题目背景

棋局

河南省实验中学
信息技术组

注意事项

题目背景

题目描述

输入格式

输出格式

输入输出样例

数据范围

算法

子任务 1

子任务 2

子任务 3

子任务 4

小结

致谢

在输了一晚上的麻将之后，小 z 和小 c 卸掉了手机上的所有牌类游戏。不过这怎么可能阻挡得了他们上课颓废的决心呢？现在他们的目光盯在了棋类游戏中，但他们两个除了天天下飞行器以外，几乎所有棋类游戏都只懂个大概规则。

“既然我们都会玩但只能玩一点点，不如我们自己搞个缝合怪出来吧！”

于是，在他们的精心脑洞之下，一个融合了围棋、象棋与军棋的奇妙游戏诞生了……



题目描述

棋局

河南省实验中学
信息技术组

注意事项

题目背景

题目描述

输入格式

输出格式

输入输出样例

数据范围

算法

子任务 1

子任务 2

子任务 3

子任务 4

小结

致谢

游戏在一张长 n 行宽 m 列的网格形棋盘上进行，棋子落在网格的交叉点上。我们不妨记左上角的交叉点的坐标为 $(1, 1)$ ，右下角的交叉点坐标为 (n, m) 。

棋子分为黑白两色，对局双方各执一方棋子。

每个棋子除了颜色以外还有等级，不妨设 col_i 为棋子 i 的颜色， lv_i 为棋子 i 的等级。另外，棋盘上的网格线共有 4 种状态，对于第 i 条网格线，设其状态为 opt_i 。



题目描述

棋局

河南省实验中学
信息技术组

注意事项

题目背景

题目描述

输入格式

输出格式

输入输出样例

数据范围

算法

子任务 1

子任务 2

子任务 3

子任务 4

小结

致谢

轮到每方下棋时，他可以选择棋盘上的一个己方棋子沿网格线进行移动到另一个交叉点，称为走子。形式化定义走子的过程如下：选择一个坐标序列 $(x_0, y_0), (x_1, y_1), \dots, (x_k, y_k)$ ，其中 k 是任意选定的正整数， (x_0, y_0) 是棋子初始的位置， (x_k, y_k) 是棋子最终走到的位置，需要满足：

- 对于任意 $i = 0, 1, \dots, k-1$ ，坐标 (x_i, y_i) 和 (x_{i+1}, y_{i+1}) 之间必须有网格线直接相连，也就是说走子必须沿着网格线走；
- 对于任意 $i \neq j$ ，必须有 $(x_i, y_i) \neq (x_j, y_j)$ ，也就是说走子过程中不能经过重复位置，特别地 $(x_0, y_0) \neq (x_k, y_k)$ ，也就是说不能原地不动(或走回原地)。
- 对于任意 $i = 1, \dots, k-1$ ，坐标 (x_i, y_i) 上必须没有棋子，也就是说走子时不能越过已有的棋子。
- 若 (x_k, y_k) 上没有棋子，称为普通走子，否则称为吃子。在吃子过程中，设正在走的棋子颜色为 col_1 ，等级为 lv_1 ，被吃的棋子颜色为 col_2 ，等级为 lv_2 ，则必须满足 $col_1 \neq col_2, lv_1 \geq lv_2$ ，换句话说只能吃与自己颜色不同，且等级不高于自己等级的棋子。

需要注意的是，由上述定义可以得出，不允许棋子在吃子后继续向前走。



题目描述

棋局

河南省实验中学
信息技术组

注意事项

题目背景

题目描述

输入格式

输出格式

输入输出样例

数据范围

算法

子任务 1

子任务 2

子任务 3

子任务 4

小结

致谢

网格线的状态含义如下所述：

- 如果 $opt_i = 0$ ，代表此路不通，走子时不能经过这条网格线。
- 如果 $opt_i = 1$ ，代表这条网格线是一条“普通道路”，每次走子时棋子最多只能经过 1 条普通道路。
- 如果 $opt_i = 2$ ，代表这条网格线是一条“直行道路”，每次走子时棋子可以经过任意条直行道路，但只能一直沿横向或一直沿纵向走，不能转弯。如沿直行道路从 (1,1) 经过 (1,2) 走到 (1,3) 是可以的，但是从 (1,1) 经过 (1,2) 走到 (2,2) 不行。
- 如果 $opt_i = 3$ ，代表这条网格线是一条“互通道路”，每次走子时棋子可以经过任意条互通道路，且中途可任意转弯。

同时规定在一次走子过程中，棋子经过的网格线的状态必须全部相同，比如从 (1,1) 经过直行道路走到 (1,2) 再经过互通道路走到 (1,3) 是不允许的。

至于如何判断胜负等其他细节，与本题无关，故略去。



题目描述

棋局

河南省实验中学
信息技术组

注意事项

题目背景

题目描述

输入格式

输出格式

输入输出样例

数据范围

算法

子任务 1

子任务 2

子任务 3

子任务 4

小结

致谢

小 z 和小 c 开发出这款棋类游戏后，为了提升水平，想了一个训练的策略：一开始棋盘是空的，然后小 c 会每次往棋盘的某个空交叉点上放一枚棋子，小 z 需要快速计算出：若选择这枚新放上的棋子进行一次走子，棋盘上一共有多少个位置是能被走到的？注意，因为这只是思维训练，他们并不会真的走这枚棋子。

可怜的小 z 发现他的计算力不足以算出这个问题，只好向你求助。



输入格式

棋局

河南省实验中学
信息技术组

注意事项

题目背景

题目描述

输入格式

输出格式

输入输出样例

数据范围

算法

子任务 1

子任务 2

子任务 3

子任务 4

小结

致谢

每个测试点由多组数据组成。

第一行：一个正整数 T 表示数据组数。

对于每组数据：

第一行：3 个正整数 n, m, q ，分别表示棋盘的行数、列数和游戏的轮数。

接下来 n 行，每行为一个长 $m-1$ 的字符串，每个字符为 0 、 1 、 2 、 3 中的一个，第 i 行第 j 个字符表示交叉点 (i, j) 连向交叉点 $(i, j+1)$ 的网格线状态。

接下来 $n-1$ 行，每行为一个长 m 的字符串，每个字符为 0 、 1 、 2 、 3 中的一个，第 i 行第 j 个字符表示交叉点 (i, j) 连向交叉点 $(i+1, j)$ 的网格线状态。

接下来 q 行，每行 4 个非负整数 col_i, lv_i, x_i, y_i ，表示在第 i 轮有一枚颜色为 col_i ，等级为 lv_i 的棋子放在了交叉点 (x_i, y_i) 上。其中 $col_i = 0$ 表示黑子， $col_i = 1$ 表示白子。保证之前交叉点 (x_i, y_i) 上没有棋子。



输出格式

棋局

河南省实验中学
信息技术组

对于每组数据输出 q 行，每行一个非负整数，表示第 i 枚棋子放置后能走到的交叉点数量。

注意事项

题目背景

题目描述

输入格式

输出格式

输入输出样例

数据范围

算法

子任务 1

子任务 2

子任务 3

子任务 4

小结

致谢



输入输出样例 1

棋局

河南省实验中学
信息技术组

注意事项

题目背景

题目描述

输入格式

输出格式

输入输出样例

数据范围

算法

子任务 1

子任务 2

子任务 3

子任务 4

小结

致谢

【样例输入 1】

```
1
3 3 5
13
22
23
010
233
0 1 2 3
1 2 2 1
1 3 1 2
0 2 3 2
1 3 2 2
```

【样例输出 1】

```
4
3
3
3
2
```

【样例 1 解释】

放置棋子 1 后，它能走到的位置为
(2, 1), (2, 2), (3, 2), (3, 3)。

放置棋子 2 后，它能走到的位置为 (2, 2), (2, 3), (3, 1)。

放置棋子 3 后，它能走到的位置为 (1, 1), (1, 3), (2, 2)。

放置棋子 4 后，它能走到的位置为 (2, 2), (3, 1), (3, 3)。

放置棋子 5 后，它能走到的位置为 (2, 3), (3, 2)。



输入输出样例 1

棋局

河南省实验中学
信息技术组

注意事项

题目背景

题目描述

输入格式

输出格式

输入输出样例

数据范围

算法

子任务 1

子任务 2

子任务 3

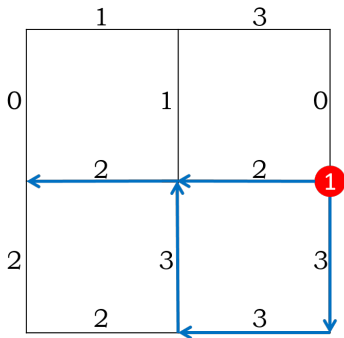
子任务 4

小结

致谢

【样例输入 1】

```
1
3 3 5
13
22
23
010
233
0 1 2 3
1 2 2 1
1 3 1 2
0 2 3 2
1 3 2 2
```



图：在 (2,3) 放下一枚等级为 1 的白棋



输入输出样例 1

棋局

河南省实验中学
信息技术组

注意事项

题目背景

题目描述

输入格式

输出格式

输入输出样例

数据范围

算法

子任务 1

子任务 2

子任务 3

子任务 4

小结

致谢

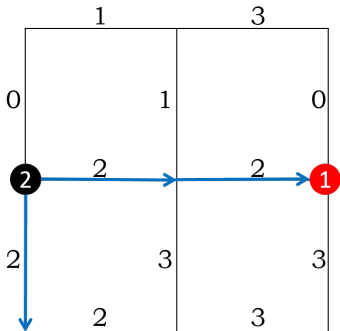


图: 在 (2,1) 放下一枚等级为 2 的黑棋

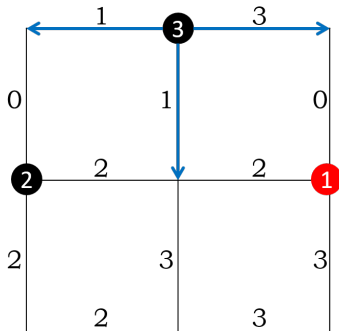


图: 在 (1,2) 放下一枚等级为 3 的黑棋



输入输出样例 1

棋局

河南省实验中学
信息技术组

注意事项

题目背景

题目描述

输入格式

输出格式

输入输出样例

数据范围

算法

子任务 1

子任务 2

子任务 3

子任务 4

小结

致谢

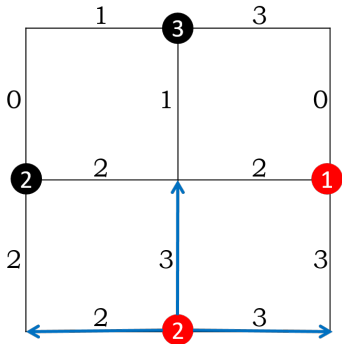


图: 在 (3,2) 放下一枚等级为 2 的白棋

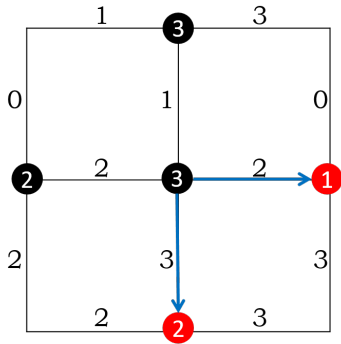


图: 在 (2,3) 放下一枚等级为 3 的黑棋



输入输出样例 2

棋局

河南省实验中学
信息技术组

注意事项

题目背景

题目描述

输入格式

输出格式

输入输出样例

数据范围

算法

子任务 1

子任务 2

子任务 3

子任务 4

小结

致谢

【样例输入 2】

```
2
2 3 4
22
33
123
0 2 1 2
0 1 2 1
1 2 1 3
0 3 2 2
3 2 3
3
1
3
32
32
0 2 1 2
1 2 3 2
0 1 2 2
```

【样例输出 2】

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



输入输出样例 2

棋局

河南省实验中学
信息技术组

注意事项

题目背景

题目描述

输入格式

输出格式

输入输出样例

数据范围

算法

子任务 1

子任务 2

子任务 3

子任务 4

小结

致谢

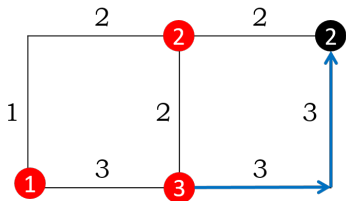
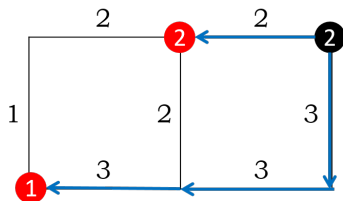
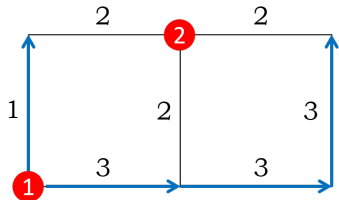
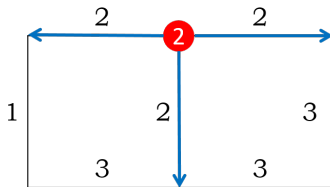


图: 样例 2 测试点 1



输入输出样例 2

棋局

河南省实验中学
信息技术组

注意事项

题目背景

题目描述

输入格式

输出格式

输入输出样例

数据范围

算法

子任务 1

子任务 2

子任务 3

子任务 4

小结

致谢

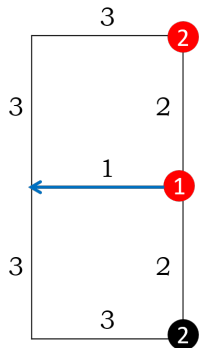
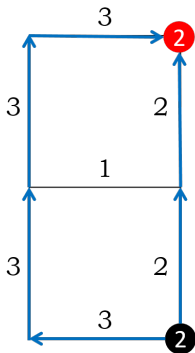
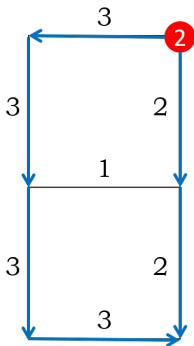


图: 样例 2 测试点 2



输入输出样例 3、4

棋局

河南省实验中学
信息技术组

注意事项

题目背景

题目描述

输入格式

输出格式

输入输出样例

数据范围

算法

子任务 1

子任务 2

子任务 3

子任务 4

小结

致谢

【输入输出样例 3】

见选手目录下的 chess/chess.in 与 chess/chess.ans。

【输入输出样例 4】

见选手目录下的 chess/chess.in 与 chess/chess.ans。



数据范围

棋局

河南省实验中学
信息技术组

注意事项

题目背景

题目描述

输入格式

输出格式

输入输出样例

数据范围

算法

子任务 1

子任务 2

子任务 3

子任务 4

小结

致谢

测试点编号	$n \times m \leq$	$q \leq$	特殊性质
1 ~ 2	100	50	无
3 ~ 6	5000	2000	
7 ~ 8	2×10^5	10^5	不存在“直行道路”与“互通道路”
9 ~ 11			不存在“互通道路”
12 ~ 14			不存在“直行道路”
15 ~ 16			$lv_i = i$
17 ~ 18			$lv_i = q - i + 1$
19 ~ 21		2000	$n, m \leq 1000$
22 ~ 25		10^5	无

对于 100% 的数据, $1 \leq T \leq 5, 2 \leq n, m \leq 10^5, 4 \leq n \times m \leq 2 \times 10^5, 1 \leq q \leq \min\{10^5, n \times m\}, 1 \leq lv_i \leq q, 1 \leq x_i \leq n, 1 \leq y_i \leq m, col_i \in \{0, 1\}$ 。

注: 由于本题输入输出规模较大, 建议使用较为快速的输入输出方式。



算法

棋局

河南省实验中学
信息技术组

注意事项

题目背景

题目描述

输入格式

输出格式

输入输出样例

数据范围

算法

子任务 1

子任务 2

子任务 3

子任务 4

小结

致谢

- 根据题意，显然需要构造一个以交叉点为点、以网格线为边的图。
- 由于 $n, m \leq 10^5$ 而 $n \times m \leq 2 \times 10^5$ 可以得出：不应当存储和使用原棋盘，而是将其转换为普通带权图。
- 本题解提供一种转换方式：对于交叉点 (i, j) 将其转换为普通图的顶点 $x = (i - 1) \times n + j$ ，这样每个交叉点对应 $1 \sim n \times m$ 之间的一个整数。
- 构图时间复杂度： $O(NM)$ 。



子任务 1: 测试点 1 ~ 6

棋局

河南省实验中学
信息技术组

注意事项

题目背景

题目描述

输入格式

输出格式

输入输出样例

数据范围

算法

子任务 1

子任务 2

子任务 3

子任务 4

小结

致谢

- 对于测试点 1 ~ 6, 数据范围小, 按照题意模拟。
- 对于 $opt_i = 1$ (普通道路), 直接判断上下左右四个位置即可。
- 对于 $opt_i = 2$ (直行道路), 需要判断上下左右四个方向。
- 对于 $opt_i = 3$ (互通道路), 按照普通图的遍历方法遍历每个点即可。
- 注意: 上述方法能到达的顶点可能存在重复, 注意去重。
- 查询时间复杂度: $O(NMq)$ 。
- 预计得分: 24 分。



子任务 2: 测试点 7 ~ 8

棋局

河南省实验中学
信息技术组

注意事项

题目背景

题目描述

输入格式

输出格式

输入输出样例

数据范围

算法

子任务 1

子任务 2

子任务 3

子任务 4

小结

致谢

- 对于测试点 7 ~ 8, 只存在 $opt_i = 1$ (普通道路)。
- 对于 $opt_i = 1$ (普通道路), 只能走一步, 也即对于一个棋子最多只能走上下左右四个位置。
- 那么对于每个询问, 直接判断上下左右四个位置即可。
- 查询时间复杂度: $O(q)$ 。
- 预计得分: 8 分。



子任务 3: 测试点 9 ~ 11

棋局

河南省实验中学
信息技术组

注意事项

题目背景

题目描述

输入格式

输出格式

输入输出样例

数据范围

算法

子任务 1

子任务 2

子任务 3

子任务 4

小结

致谢

- 对于测试点 9 ~ 11, 不存在 $opt_i = 3$ (互通道路), 那么主要处理 $opt_i = 1$ (普通道路) 和 $opt_i = 2$ (直行道路) 即可。
- 如果对于每个棋子进行模拟, 显然时间复杂度为 $O(q(N + M))$, 不可行。
- 注意到我们只需要知道棋子可以到达连通块大小而不需要知道到达的具体位置, 因此可以用并查集来处理。
- 而添加棋子等于将集合拆分, 此过程使用并查集不易维护, 因此将操作离线化, 首先将所有棋子放在棋盘上, 然后每次倒序删除一个棋子, 删除前查询对应的答案。
- 为了避免集合相交, 我们只用并查集维护空白位置构成的连通块, 对于吃子的情况再做额外处理。
- 建立维护行、列空白区域的并查集, 对于每个查询:
 - 查询四个方向对应空白区域大小和最远端吃子情况。
 - 删除棋子, 将行、列空白区域合并。
- 查询时间复杂度: $O(q)$ 。
- 预计得分: 12 分。



子任务 3: 测试点 9 ~ 11

棋局

河南省实验中学
信息技术组

注意事项

题目背景

题目描述

输入格式

输出格式

输入输出样例

数据范围

算法

子任务 1

子任务 2

子任务 3

子任务 4

小结

致谢

- 对于棋子上方，可以到达的空白位置总数为 3，棋子不能吃掉最上棋子。
- 对于棋子下方，可以到达的空白位置总数为 0，棋子可以吃掉最下棋子。
- 对于棋子左方，可以到达的空白位置总数为 5，棋子可以吃掉最左棋子。
- 对于棋子右方，可以到达的空白位置总数为 2，棋子不能吃掉最右棋子。
- 对于本次询问，答案为 $3 + 0 + 0 + 1 + 5 + 1 + 2 + 0 = 12$ 。

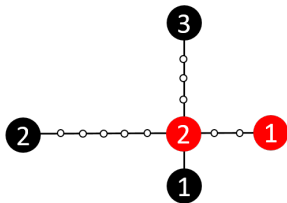


图: 案例



子任务 3: 测试点 9 ~ 11

棋局

河南省实验中学
信息技术组

注意事项

题目背景

题目描述

输入格式

输出格式

输入输出样例

数据范围

算法

子任务 1

子任务 2

子任务 3

子任务 4

小结

致谢

- 为了能直接获取最远端棋子，我们需要在并查集内维护集合内的最小位置和最大位置。
- 那么最远端棋子 (如果存在且可以通过 2 类道路到达) 一定是最小位置或最大位置的相邻位置。

```
1 struct UFSET
2 {
3     int fa;
4     int lid; // 最小值
5     int rid; // 最大值
6     int sz;
7 } row_set[NM], col_set[NM]; // 行列对应的并查集
```



子任务 3: 测试点 9 ~ 11

棋局

河南省实验中学
信息技术组

注意事项

题目背景

题目描述

输入格式

输出格式

输入输出样例

数据范围

算法

子任务 1

子任务 2

子任务 3

子任务 4

小结

致谢

- 在将所有棋子放置在棋盘后，需要先查找对应行列的空白区域，进行并查集合并。

```
1 // 并查集初始化
2 void ufset_init()
3 {
4     for(int i = 1; i <= n * m; ++i) row_set[i] = col_set[i] = { i, i, i, 1};
5     for(int i = 1; i <= n * m - 1; ++i) // 行上合并
6     {
7         if(col[i] == -1 && g[i][R] == 2 && col[i + 1] == -1)
8         {
9             merge(row_set, i, i + 1);
10        }
11    }
12    for(int i = 1; i <= (n - 1) * m; ++i) // 列上合并
13    {
14        if(col[i] == -1 && g[i][D] == 2 && col[i + m] == -1)
15        {
16            merge(col_set, i, i + m);
17        }
18    }
19 }
```



子任务 3: 测试点 9 ~ 11

棋局

河南省实验中学
信息技术组

注意事项

题目背景

题目描述

输入格式

输出格式

输入输出样例

数据范围

算法

子任务 1

子任务 2

子任务 3

子任务 4

小结

致谢

```
1 // 2 类道路
2 void direct(int x, int dir)
3 {
4     int y = v[x][dir];
5     if(col[y] >= 0 && eat(x, y)) ++cnt; // x 的相邻位置有棋子 则只需要检查能否吃掉
6     else if(col[y] == -1)                // x 的相邻位置没有棋子 需要检查并查集
7     {
8         if(dir == U || dir == D)          // 上下
9         {
10             int root = get(col_set, y);
11             cnt += col_set[root].sz;      // 答案加上集合大小
12             int z = 0;                    // 集合最边缘的位置
13             if(dir == U) z = col_set[root].lid; // 最上方位置
14             else if(dir == D) z = col_set[root].rid; // 最下方位置
15             if(g[z][dir] == 2 && eat(x, v[z][dir])) ++cnt; // x 能否吃最上 or 最下棋子
16             merge(col_set, root, x);      // 棋子撤回 集合合并
17         }
18         if(dir == L || dir == R)          // 左右
19         {
20             // 情况类似 注意此时需要处理行并查集 row_set
21         }
22     }
23 }
```



子任务 4

棋局

河南省实验中学
信息技术组

注意事项

题目背景

题目描述

输入格式

输出格式

输入输出样例

数据范围

算法

子任务 1

子任务 2

子任务 3

子任务 4

小结

致谢



小结

棋局

河南省实验中学
信息技术组

注意事项

题目背景

题目描述

输入格式

输出格式

输入输出样例

数据范围

算法

子任务 1

子任务 2

子任务 3

子任务 4

小结

致谢

- 本题考察了选手代码水平。
- 本题要考虑卡常问题。
- 本题着重考察了选手数据结构的应用能力。



致谢

棋局

河南省实验中学
信息技术组

注意事项

题目背景

题目描述

输入格式

输出格式

输入输出样例

数据范围

算法

子任务 1

子任务 2

子任务 3

子任务 4

小结

致谢

谢谢观看！