

2015元旦集训题

2014 年 12 月 25 日

题目名称	方块	下棋	树
可执行文件名	cube	chess	tree
输入文件名	cube.in	chess.in	tree.in
输出文件名	cube.out	chess.out	tree.out
内存限制	64MB	64MB	256MB
每个测试点时限	0.5s	0.5s	2.5s
测试点数目	10	10	10
每个测试点分值	10	10	10
是否有部分分	无	无	无
题目类型	传统	传统	传统
备注	无	无	开O2

提交文件名需加后缀:

对于pascal语言	cube.pas	chess.pas	tree.pas
对于C语言	cube.c	chess.c	tree.c
对于C++语言	cube.cpp	chess.cpp	tree.cpp

注意:

- 第三题除了O2别的优化什么都没开。
- 实际时间限制由评测机的运行速度决定（大概比标程多10%）。
- 请不要在考场大喊“我要爆零了”，“好难怎么做”等话语，影响他人考试。
- 可以喊“这题太水我要AK了”。

方块

(cube / cube.in / cube.out)

小明走在路上，被一个不明的硬物砸中了脑袋，~~并~~

~~那么问题来了，~~

~~求太阳的质量，~~

回到正题。小明爬起来后，发现砸中他的是一个边长为 N 的方块。

比较奇特的是，这个方块是由 $1 \times 1 \times 1$ 的小方块组装而成的。不过，因为这个方块是从高处摔落下来的，其中的一些小方块已经被弹飞而消失了。

更好玩的是，这些小方块的面都被几种不同颜色的染料上过色。但可以保证的是，每个小方块的六面颜色都是相同的。

小明通过观察得到了这个方块的六视图，现在他想知道的是，这个方块中最多可能有多少小方块。

快点告诉他，不然他要拿方块砸你了。

【输入文件】

输入文件为cube.in。

一个测试点中包含有多组数据。

第1行包含一个正整数 N ，表示方块的边长。

接下来 N 行，从左至右依次给出这个方块前，左，后，右，上，下的视图，相邻视图之间用空格隔开。

一个视图对应 N 个长度为 N 的串，其中字符‘.’表示该位置可以被看穿，而大写字母代表颜色（不同字母代表不同颜色）。

输入以 $N = 0$ 结束。

【输出文件】

输出文件为cube.out。

对于每组数据，输出一行Maximum weight: ans gram(s)，其中ans表示这组数据的答案。

【输入样例】

3

.R. YYR .Y. RYY .Y. .R.

GRB YGR BYG RBY GYB GRB

.R. YRR .Y. RRY .R. .Y.

2

ZZ ZZ ZZ ZZ ZZ ZZ

ZZ ZZ ZZ ZZ ZZ ZZ

0

【输出样例】

Maximum weight: 11 gram(s)

Maximum weight: 8 gram(s)

【数据规模与约定】

对于50%的数据， $N \leq 5$ 。

对于100%的数据，保证 $N \leq 10$, 数据组数 ≤ 5000 。

注意，数据中输入的小方块之间可能是不联通的，也就是说方块可能被分成了许多部分 ~~//(属于力与树同在)//~~。

下棋

(chess / chess.in / chess.out)

【题目大意】

Alice和Bob在玩这样一个游戏：

有一个 n 个点， m 条边的有向图，每条边都有自己的颜色。保证图中无环。

一开始，有 Q 个棋子分布在不同的点中，然后两人轮流进行如下的操作：

1. 选择一个包含棋子的点 x 。从该点中选择一个棋子。
2. 选择一个颜色的集合 G 。
3. 对于一条由点 x 连出的路径 r ，如果满足 $color(r) \in G$ ，则在边 r 的终点中增加一个棋子。
4. 将点 x 中所选的棋子删除。

不能操作的人输。现在你需要知道，在双方都采用最优策略的情况下，先手Alice是否可以取得胜利。

【输入文件】

输入文件为chess.in。

第1行包含两个整数 N, M ，表示所给图的点数和边数。

第 $2 \Rightarrow M + 1$ 行表示图中的边。一行包含三个整数 st, ed, col ，分别表示所给边的起点，终点以及颜色。

接下来一行包含一个整数 Q ，表示图中的棋子个数。

最后一行包含 Q 个整数，表示这些棋子所在的点。

【输出文件】

输出文件为chess.out。

当你可以胜利时输出1，否则输出0。

【输入样例】

```
2 1
2 1 1
1
2
```

【输出样例】

```
1
```

【输入样例】

```
2 1
2 1 1
1
1
```

【输出样例】

```
0
```

【样例解释】

在第一个样例中，你可以把唯一的棋子删除，然后在点1中增加一个棋子。对方无法继续操作。另一种方法是把唯一的棋子删除，然后不在任何点中增加棋子。

【数据规模与约定】

为了防止一些情况，该题的一个数据中包含五个小数据。你的程序需要对每个小数据分别运行一次。只有一组数据中的五个小数据都正确，你才能得到这个数据的满分，否则该数据得0分。

一共有十个数据，每个数据占十分。

对于前两个数据，有 $Q \leq 20, N \leq 5, M \leq 10$ 。

对于第三个数据，有 $Q \leq 100, N \leq 20, M \leq 100$ 。

对于所有数据，有 $Q \leq 100000, N \leq 200, M \leq 5000$ ，路径的颜色 $col \leq 5000$ 。

树

(tree / tree.in / tree.out)

【题目大意】

有一棵 n 个点的树，点的标号为 $1 \rightarrow n$ 。树中的边有边权。给你 M 个询问，每个询问包含三个参数 l, r, pos ，你要求出标号在 $l \rightarrow r$ 之间的所有点中，到节点 pos 距离最近的点离 pos 有多远。

【输入文件】

输入文件为tree.in。

第1行包含一个正整数 N ，表示树的节点个数。节点的编号为 $1 \Rightarrow N$ 。

第 $2 \Rightarrow N$ 行每行表示一条树边。一行包含3个正整数 x, y, d ，分别表示这条边的两个端点以及长度。

第 $N + 1$ 行包含一个正整数 M ，表示有多少询问。

第 $N + 2 \Rightarrow N + M + 1$ 行每行表示一个询问的信息。一行包含3个正整数 l, r, pos ，表示该询问的三个参数。

为了卡掉离线算法，输入中的 pos 实际上等于 $pos \oplus lastans$ 。其中 $lastans$ 表示上次询问的答案，初始为0。

【输出文件】

输出文件为tree.out。

对于每个询问，输出一个数 ans ，表示答案。

一个输出占一行。

【输入样例】

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

【输出样例】

```
1
0
2
```

【数据规模与约定】

对于50%的数据，保证 $N \leq 1000, M \leq 1000$ 。

对于70%的数据，保证所给的树是随机的。

对于100%的数据，保证 $N \leq 100000, M \leq 100000$ ，保证询问的 $l \leq r$ 。

对于100%的数据，保证树中不会有超过25000的链。