非传统题选讲

N_z_

January 11, 2023

Will liuhengxi achieve LGM in grade 9?

By dXqwq, history, 10 days ago,

> As far as I know, he is in grade 9 now, which is one year younger than orzdevinwang.
>
> He is gaining rating in recent contests and previous LGMs in grade 9 are djq-cpp and orzdevinwang(comment below if you know more).
>
> Meanwhile I'm stuck between 2600~2700, hope I will become stronger next year o(>_<)o

+49 | dXqwq | 10 days ago | 5

Comments (5)

Write comment?

N_z__ — 10 days ago, # Edit | +34

As a classmate of liuhengxi,I think he will.

→ Reply

moonrise__ — 10 days ago, # ^ | +68

As a classmate of liuhengxi, I think he wants a girlfriend.

→ Reply

User_Carrot — 10 days ago, # ^ | +20

As a classmate and also a fan of liuhengxi, I think he will. And he wants a girlfriend btw.

→ Reply

相信大家都知道非传统题有哪些吧。

序

相信大家都知道非传统题有哪些吧。
如果内容太水可以主动坐到前面来回答问题。

Example

~~uoj8 Quine~~

Problem

有值为 $1,2,3\cdots,9$ 的 9 个面包，你和交互库轮流取面包，先取到有三个面包和为 15 的获胜，有 600 次你先手，1200 次你后手，保证交互库随机。

当你没有败，且胜率的平方 ≥ 0.8 时你获得满分。

Solution

注意到这个规则等同于一个井字棋，进行贪心即可。

Solution

注意到这个规则等同于一个井字棋，进行贪心即可。
当然，由于状态数很少，你可以 dp 计算你下一步方案的胜率。

1 前言

2 交互
- P7135 小 B 的面包
- soj1707 【STR #53】鸽子四子棋
- CF1773H Hot and Cold
- CF1773I Interactive Factorial Guessing

3 通信

4 提答

5 题目

Problem

$5 \times 5 \times 5$ 的棋盘，你先手，和交互库下重力四子棋（棋子会按重力下落，连成 4 个子（13 个方向任意一个都可以））。

30 个点，交互库会在测试点编号 $\times 2$，之后随机下。

对于前 10 个测试点，你每赢一次能得到 3 分，平一次能得到 1 分。

对于中间 10 个测试点，你每赢一次能得到 5 分，平一次能得到 2 分。

对于后 10 个测试点，你每赢一次能得到 10 分，平一次能得到 5 分。

得分与 100 取 min。

Solution

题解做法：写一个 dfs 函数，对四步后的所有状态给他一个权值，权值越大理应的可能性越大，这个权值是所有自己还有机会组成连续四个棋子的地方，用一个数分别除以每一个地方要下到的最小步数，最后在加起来，同时减去对方的这个数，最后就是权值，中间那四步下哪一步博弈一下好了。

Solution

题解做法：写一个 dfs 函数，对四步后的所有状态给他一个权值，权值越大理应的可能性越大，这个权值是所有自己还有机会组成连续四个棋子的地方，用一个数分别除以每一个地方要下到的最小步数，最后在加起来，同时减去对方的这个数，最后就是权值，中间那四步下哪一步博弈一下好了。

我个人做法：先点 $(3,3)$，对于一个局面，先取自己能赢的点，然后取对手能赢的点，对每个点按照先权值 1，再权值 2 的顺序排序，权值 1 是点上这个点之后自己利用这个一步能赢加上之后自己利用这个有机会赢的情况数与对手这个情况数的 max，权值 2 是两步的情况数。发现这个开局很菜。

Solution

题解做法：写一个 dfs 函数，对四步后的所有状态给他一个权值，权值越大理应的可能性越大，这个权值是所有自己还有机会组成连续四个棋子的地方，用一个数分别除以每一个地方要下到的最小步数，最后在加起来，同时减去对方的这个数，最后就是权值，中间那四步下哪一步博弈一下好了。

我个人做法：先点 $(3,3)$，对于一个局面，先取自己能赢的点，然后取对手能赢的点，对每个点按照先权值 1，再权值 2 的顺序排序，权值 1 是点上这个点之后自己利用这个一步能赢加上之后自己利用这个有机会赢的情况数与对手这个情况数的 $\max$，权值 2 是两步的情况数。发现这个开局很菜。

考虑开局下模仿棋，交互库很菜，就过了。

CF1773H Hot and Cold

Problem

交互库有一个点 $(x, y), 0 \le x, y \le 10^6$，你可以询问一个点，会返回与上一个点相比里目标点近还是远了，或是距离相同，或是找到了。第一次的回答没有作用，你要在 64 次询问内找到目标点。

交互库是自适应的，并且你不知道输出含义是近了，远了还是相同，但是保证相同返回对应相同单词，不同返回对应不同单词。

Solution

考虑询问 $(0,0),(0,1),(1,0)$ 如果返回相同，那么这就是近了。考虑对 x,y 同时二分，3 次可以使区间变成 $\frac{1}{4}$，即对于返回与开始两个相同就是近了，否则当做远了处理。

Solution

考虑询问 $(0,0),(0,1),(1,0)$ 如果返回相同，那么这就是近了。考虑对 x,y 同时二分，3 次可以使区间变成 $\frac{1}{4}$，即对于返回与开始两个相同就是近了，否则当做远了处理。

否则询问 $(0,maxn),(maxn,0),(maxn,maxn),(maxn,maxn-1)$ 最后一次询问作为近了，如果最开始两次询问中第一次是近了，那么在 $(0,0),(0,maxn)$ 中二分，否则在 $(0,0),(maxn,0)$ 中二分。

Solution

考虑询问 $(0,0),(0,1),(1,0)$ 如果返回相同，那么这就是近了。考虑对 x,y 同时二分，3 次可以使区间变成 $\frac{1}{4}$，即对于返回与开始两个相同就是近了，否则当做远了处理。

否则询问 $(0,maxn),(maxn,0),(maxn,maxn),(maxn,maxn-1)$ 最后一次询问作为近了，如果最开始两次询问中第一次是近了，那么在 $(0,0),(0,maxn)$ 中二分，否则在 $(0,0),(maxn,0)$ 中二分。

询问复杂度 $\Theta(3\times\log_2(maxn))$，有一定的细节（指 +22）。

2022-2023 ICPC, NERC, Northern Eurasia Onsite (Unrated, Online Mirror, ICPC Rules, Teams Preferred)

Final standings

Double click (or ctrl+click) each entry to view its submission history

Standings

#	Who	=	Penalty	A	B	C	D	E	F	G	H	I	J	K	L
1 (44)	JackF 被 skc 杀死了: realskc, User_Carrot, N_z__	8	1572	+2 00:59	+ 02:12		-3	+1 00:17	+1 00:08	+ 04:36	+22 03:46	+7 01:39		+1 01:15	
	* N_z__	1					+17								
	Accepted Tried			667 959	301 372	14 25	145 182	1877 2233	2165 2822	87 119	150 215	176 237	36 63	716 943	2 36

1 前言

2 交互
- P7135 小 B 的面包
- soj1707 【STR #53】鸽子四子棋
- CF1773H Hot and Cold
- CF1773I Interactive Factorial Guessing

3 通信

4 提答

5 题目

Problem

交互库有一个整数 $n \in [1,5982]$，你可以询问 $n!$ 的第 x 位不超过 10 次，猜出这个 n。

$t \le 100$ 组数据。

Solution

考虑计算询问每一位带来的信息熵。可是这是不对的。

Solution

考虑计算询问每一位带来的信息熵。可是这是不对的。

一共只有 5982 组，交互库很容易全部询问一遍，所以我们可以当作交互库是自适应的，那么上面一种做法是有问题的。

Solution

考虑计算询问每一位带来的信息熵。可是这是不对的。

一共只有 5982 组，交互库很容易全部询问一遍，所以我们可以当作交互库是自适应的，那么上面一种做法是有问题的。

我们可以计算询问哪一位使得这一位数码在 $0, 1, 2, \cdots, 9$ 中最多的一个最少，询问之，可是还是不对。

Solution

考虑计算询问每一位带来的信息熵。可是这是不对的。

一共只有 5982 组，交互库很容易全部询问一遍，所以我们可以当作交互库是自适应的，那么上面一种做法是有问题的。

我们可以计算询问哪一位使得这一位数码在 $0, 1, 2, \cdots, 9$ 中最多的一个最少，询问之，可是还是不对。

考虑这种做法会 T，考虑预处理第一次询问，以及分类讨论第一次询问结果确定第二次询问内容（我还预处理了第三次），即可。

1 前言

2 交互

3 通信

- uoj454 【UER #8】打雪仗
- soj1269 THUSC2021 D1T4

4 提答

5 题目

Problem

Alice 和 Bob 两人通信，Alice 得到一个长为 $2n$ 的 01 串，Bob 得到 n 个位置。

你需要进行通信，两人只能发送 01 值，多的一方不超过 m 个，让 Bob 输出这 n 个位置的 01 值。

Solution

我们首先给出一个 50 分做法，Bob 每次告诉 Alice 自己是要接下来第一个还是第二个。

Solution

我们首先给出一个 50 分做法，Bob 每次告诉 Alice 自己是要接下来第一个还是第二个。

考虑有什么问题，如果有数据要 $n+1,n+2...2n$，那么询问次数就超了。

Solution

我们首先给出一个 50 分做法，Bob 每次告诉 Alice 自己是要接下来第一个还是第二个。

考虑有什么问题，如果有数据要 n+1,n+2...2n，那么询问次数就超了。

我们可以采用随机化的方法规避刻意构造的数据，实测可以通过。

Solution

我们首先给出一个 50 分做法，Bob 每次告诉 Alice 自己是要接下来第一个还是第二个。

考虑有什么问题，如果有数据要 n+1,n+2...2n，那么询问次数就超了。

我们可以采用随机化的方法规避刻意构造的数据，实测可以通过。

考虑一个更好的做法，把数列三等分，那么其中至少有一段 Bob 需要超过一半，那么对于剩下两段，Bob 传自己需要哪些，对于过半一段，Alice 给出整段，可以证明满足限制。

Solution

我们首先给出一个 50 分做法，Bob 每次告诉 Alice 自己是要接下来第一个还是第二个。

考虑有什么问题，如果有数据要 n+1,n+2...2n，那么询问次数就超了。

我们可以采用随机化的方法规避刻意构造的数据，实测可以通过。

考虑一个更好的做法，把数列三等分，那么其中至少有一段 Bob 需要超过一半，那么对于剩下两段，Bob 传自己需要哪些，对于过半一段，Alice 给出整段，可以证明满足限制。

uoj 上还有一个更优但是复杂度过高的做法，这里就不讲了。

1 前言

2 交互

3 通信
- uoj454 【UER #8】打雪仗
- soj1269 THUSC2021 D1T4

4 提答

5 题目

Problem

编码与解码一棵 $n \le 80$ 个点的无标号有根儿子有顺序的无根树，用 int128 存储。

多测，$t \le 10^5$。

Solution

我们考虑这棵树的括号序列，考虑头和尾一定匹配，删去，得到一个长 $2n-2$ 的括号序列。

考虑这样的序列总数是 $Cat(69) < 2^{128}$，对其编码即可。

实现上可以采用 dp。

Problem

给出 12 条公理和 MP 规则（即若 A 和 $A \to B$ 成立，那么 B 成立），证明 14 个定理。

Problem

给出 12 条公理和 MP 规则（即若 A 和 $A \to B$ 成立，那么 B 成立），证明 14 个定理。

公理：

1.$(A \to A)$
2.$((A \to (B \to C)) \to (B \to (A \to C)))$
3.$((A \to B) \to ((B \to C) \to (A \to C)))$
4.$((A \to (A \to B)) \to (A \to B))$
5.$((A \to (\neg B)) \to (B \to (\neg A)))$
6.$((\neg(\neg A)) \to A)$
7.$((A \wedge B) \to A)$
8.$((A \wedge B) \to B)$
9.$(A \to (B \to (A \wedge B)))$
10.$(A \to (A \vee B))$
11.$(B \to (A \vee B))$
12.$((A \to C) \to ((B \to C) \to ((A \vee B) \to C)))$

Problem

定理:

1.$((A \to A) \lor B)$

2.$(((A \to A) \to (((A \to B) \to ((B \to C) \to (A \to C))) \to ((A \to B) \to ((B \to C) \to ((A \lor B) \to C))))) \to (((A \to B) \to ((B \to C) \to (A \to C))) \to ((A \to A) \to ((A \to B) \to ((B \to C) \to ((A \lor B) \to C))))))$

3.$(((A \land B) \land C) \to A)$

4.$((A \land B) \to (A \lor B))$

5.$((A \to B) \to ((C \to A) \to (C \to B)))$

6.$((A \to B) \to ((D \to (C \to A)) \to (D \to (C \to B))))$

7.$(A \to (B \to A))$

8.$((C \to (B \to A)) \to ((C \to B) \to (C \to A)))$

9.$(((\neg A) \to (\neg B)) \to (B \to A))$

10.$((A \to B) \to ((\neg B) \to (\neg A)))$

11.$(A \to ((\neg A) \to B))$

12.$((A \to (B \to C)) \to ((A \land B) \to C))$

13.$(A \lor (\neg A))$

14.$((A \to B) \to (B \lor (\neg A)))$

Proof

证明中会省略 MP 的使用。

Proof

1.$((A \to A) \vee B)$

Proof

1.$((A \to A) \lor B)$

$(A \to A)$（公理 1）

$((A \to A) \lor B)$（公理 10）

Proof

2.$(((A \to A) \to (((A \to B) \to ((B \to C) \to (A \to C))) \to ((A \to B) \to ((B \to C) \to ((A \lor B) \to C))))) \to (((A \to B) \to ((B \to C) \to (A \to C))) \to ((A \to A) \to ((A \to B) \to ((B \to C) \to ((A \lor B) \to C))))))$

Proof

2.$(((A \to A) \to (((A \to B) \to ((B \to C) \to (A \to C))) \to ((A \to B) \to ((B \to C) \to ((A \lor B) \to C))))) \to (((A \to B) \to ((B \to C) \to (A \to C))) \to ((A \to A) \to ((A \to B) \to ((B \to C) \to ((A \lor B) \to C))))))$

$(((A \to A) \to (((A \to B) \to ((B \to C) \to (A \to C))) \to ((A \to B) \to ((B \to C) \to ((A \lor B) \to C))))) \to (((A \to B) \to ((B \to C) \to (A \to C))) \to ((A \to A) \to ((A \to B) \to ((B \to C) \to ((A \lor B) \to C))))))$

（公理 2）

Proof

3.$(((A \wedge B) \wedge C) \to A)$

Proof

3.$(((A \wedge B) \wedge C) \to A)$

$(((A \wedge B) \wedge C) \to (A \wedge B))$（公理 7）

$((A \wedge B) \to A)$（公理 7）

$(((A \wedge B) \wedge C) \to A)$（公理 3）

Proof

4.$((A \wedge B) \to (A \vee B))$

Proof

4.$((A \wedge B) \rightarrow (A \vee B))$

$((A \wedge B) \rightarrow A)$（公理 7）

$(A \rightarrow (A \vee B))$（公理 10）

$((A \wedge B) \rightarrow (A \vee B))$（公理 3）

Proof

5.$((A \to B) \to ((C \to A) \to (C \to B)))$

Proof

$5.((A \to B) \to ((C \to A) \to (C \to B)))$

$((C \to A) \to ((A \to B) \to (C \to B)))$（公理 3）

$((A \to B) \to ((C \to A) \to (C \to B)))$（公理 2）

Proof

6.$((A \to B) \to ((D \to (C \to A)) \to (D \to (C \to B))))$

Proof

$6.((A \to B) \to ((D \to (C \to A)) \to (D \to (C \to B))))$

$(((C \to A) \to (C \to B)) \to ((D \to (C \to A)) \to (D \to (C \to B))))$（定理 5）

$((A \to B) \to ((C \to A) \to (C \to B)))$（定理 5）

$((A \to B) \to ((D \to (C \to A)) \to (D \to (C \to B))))$（公理 3）

Proof

7.$(A \to (B \to A))$

Proof

7.$(A \to (B \to A))$

$((A \wedge B) \to A)$（公理 7）

$(A \to (B \to (A \wedge B)))$（公理 9）

$(A \to (B \to A))$（定理 6）

Proof

8.$((C \to (B \to A)) \to ((C \to B) \to (C \to A)))$

Proof

8.$((C \to (B \to A)) \to ((C \to B) \to (C \to A)))$

$((B \to (C \to A)) \to ((C \to B) \to (C \to (C \to A))))$（定理 5）

$((C \to (B \to A)) \to (B \to (C \to A)))$（公理 2）

$((C \to (B \to A)) \to ((C \to B) \to (C \to (C \to A))))$（公理 3）

$((C \to A) \to (C \to (C \to A)))$（定理 7）

$((C \to (B \to A)) \to ((C \to B) \to (C \to A)))$（定理 6）

Proof

9.$(((\neg A) \to (\neg B)) \to (B \to A))$

Proof

$9.(((\neg A) \to (\neg B)) \to (B \to A))$

$(((\neg A) \to (\neg B)) \to (B \to (\neg(\neg A))))$（公理 5）

$((\neg(\neg A)) \to A)$（公理 6）

$(((\neg A) \to (\neg B)) \to (B \to A))$（定理 6）

Proof

$10.((A \to B) \to ((\neg B) \to (\neg A)))$

Proof

$10.((A \to B) \to ((\neg B) \to (\neg A)))$

$((A \to (\neg(\neg B))) \to ((\neg B) \to (\neg A)))$（公理 5）

$(A \to (\neg(\neg A)))$（样例 1）

$((A \to B) \to (A \to (\neg(\neg B))))$（定理 5）

$((A \to B) \to ((\neg B) \to (\neg A)))$（公理 3）

Proof

$11.(A \to ((\neg A) \to B))$

Proof

$11.(A \to ((\neg A) \to B))$

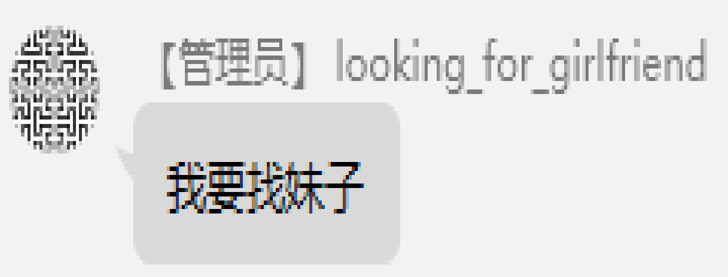
$(A \to ((\neg B) \to A))$（定理 7）

$(((\neg B) \to A) \to ((\neg A) \to (\neg(\neg B))))$（定理 10）

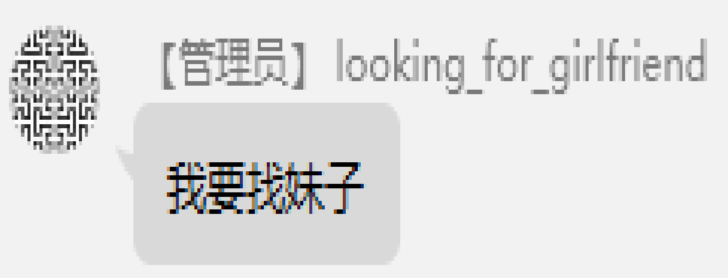
$(A \to ((\neg A) \to (\neg(\neg B))))$（公理 3）

$((\neg(\neg B)) \to B)$（公理 6）

$(A \to ((\neg A) \to B))$（定理 6）

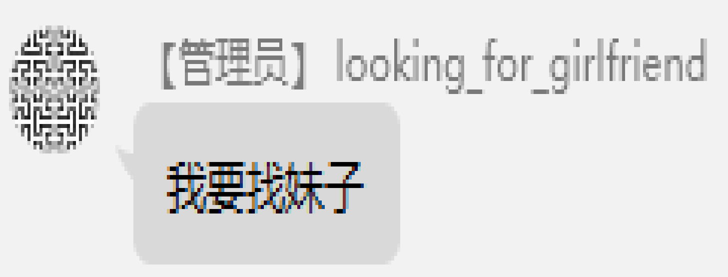

[管理员] looking_for_girlfriend
我要拔妹子

soj1348 【STR #41】小 H 的推理系统



~~接下来是一些比较抽象的东西，炸一下看看还有谁在。~~

soj1348 【STR #41】小 H 的推理系统



~~接下来是一些比较抽象的东西，炸一下看看还有谁在。~~
~~接下来我也不会表述了，就贴题解了，有兴趣可以课后来找我。~~

Proof

$12.((A \to (B \to C)) \to ((A \wedge B) \to C))$

Proof

12.$((A \to (B \to C)) \to ((A \wedge B) \to C))$

这一命题与公理 12 异曲同工，只可惜它不在公理之中。它将是我们证明定理 13 的关键工具之一。

命题的意义是将前件中的 $\to$ 折叠起来（两次），而 $((A \wedge B) \to *)$ 只会展开一层，这意味着我们需要分为两个阶段进行证明，它们之间的桥梁是 $((A \wedge B) \to (B \to C))$。借助公理 3、定理 8 与样例 2 可以完成这一证明。

Proof

13.$(A \vee (\neg A))$

Proof

13.$(A \lor (\neg A))$

这一命题似乎很难处理：它甚至不包含 $\to$。我们将从定理 11 入手进行证明。

定理 11 可以通过定理 12 转化为 $((A \land (\neg A)) \to B)$。它包含 $\land$，但本命题包含 $\lor$，这启发我们研究本命题的逆否命题，即 $(\neg(A \land (\neg A)))$。接下来的任务是：

- 证明 $(\neg(A \land (\neg A)))$ 可证；
- 证明 $((\neg(A \land (\neg A))) \to (A \lor (\neg A)))$ 可证，即逆否命题能推导出原命题。

Proof

先证明 $(\neg(A \wedge (\neg A)))$ 可证。如果某一命题 C 永真，那么证明 $(C \to (\neg(A \wedge (\neg A))))$ 即可。考虑它的逆否（利用公理 5）$((A \wedge (\neg A)) \to \neg C)$，它可以通过定理 11 和定理 12 证明。具体地，可以取 C 为 $(A \to A)$。

再证明 $((\neg(A \wedge (\neg A))) \to (A \vee (\neg A)))$ 可证。同样考虑其逆否命题 $((\neg(A \vee (\neg A))) \to (A \wedge (\neg A)))$：直观地，如果我们能证明 $((\neg(A \vee (\neg A))) \to A)$ 与 $((\neg(A \vee (\neg A))) \to (\neg A))$，就可以按下面的顺序凑出目标：

Proof

- $(A \to ((\neg A) \to (A \wedge (\neg A))))$ (公理 9)
- $((\neg(A \vee (\neg A))) \to A)$
- $((\neg(A \vee (\neg A))) \to ((\neg A) \to (A \wedge (\neg A))))$ (公理 3)
- $((\neg(A \vee (\neg A))) \to (\neg A))$
- $((\neg(A \vee (\neg A))) \to (A \wedge (\neg A)))$ (定理 8)

此时我们的任务仅仅是证明 $((\neg(A \vee (\neg A))) \to A)$（另一式同理）；而这可以通过逆否和公理 10 得到。

~~你完成了分类讨论定理，来练习一下吧。~~

Proof

14.$((A \to B) \to (B \vee (\neg A)))$

Proof

14.$((A \to B) \to (B \lor (\neg A)))$

定理 13 提供了一种得到所有可证命题的证明序列的方法；对本命题的证明就是这一方法的习题。事实上，我们只需证明以下两个命题：

- $(A \to ((A \to B) \to (B \lor (\neg A))))$
- $((\neg A) \to ((A \to B) \to (B \lor (\neg A))))$，

然后利用公理 12 将两个前件组成 $(A \lor (\neg A))$，最后套用定理 13 证明之。更一般地，若要证明一个包含 n 个命题符的命题，使用类似的分类讨论的技巧后，便只用证明 2^n 个更简单的命题。

1 前言

2 交互

3 通信

4 提答
- soj1348 【STR #41】小 H 的推理系统
- soj1521 一群数字的合作
- soj1640 一个集合的划分

5 题目

Problem

阅读 checker.cpp，对于.in，给出.out。

Problem

```
#include"testlib.h"
using namespace std;

int n,m=0,use=0;
map<string,double>mp;

double dRead(InStream&ouf,int&use)
{
    string opt=ouf.readToken();
    if(opt=="+"||opt=="-"||opt=="*"||opt=="/"||opt=="^")
    {
        double x=dRead(ouf,use),y=dRead(ouf,use),z;
        if(opt=="+") z=x+y;
        else if(opt=="-") z=x-y;
        else if(opt=="*") z=x*y;
        else if(opt=="/") z=x/y;
        else z=pow(x,y);
        if(isfinite(z)) return z;
        else quitf(_wa,"expression error");
    }
    else if(mp.count(opt))
    {
        use++;
        if(opt=="44") use++;
        return mp[opt];
    }
    else quitf(_wa,"unknown string: '%s'",opt.c_str());
}
```

Problem

```
int main(int argc,char* argv[])
{
    registerTestlibCmd(argc,argv);
    double res;
    mp["4"]=4;
    mp[".4"]=0.4;
    mp["4!"]=24;
    mp["sqrt4"]=2;
    mp["44"]=44;
    n=inf.readInt();
    res=dRead(ans,m);
    if(fabs(res-n)>1e-4) quitf(_wa,"std is wrong");
    res=dRead(ouf,use);
    if(fabs(res-n)>1e-4) quitf(_wa,"result = %lf, n = %d",res,n);
    if(use<m) quitp(3.3,"use = %d orz",use);
    else if(use==m) quitf(_ok,"use = %d",use);
    else quitp(0.3,"use = %d",use);
    return 0;
}
```

Solution

题面翻译：你需要用尽可能少的 4 表示出一个正整数 n。表达式中可以使用加、减、乘、除、幂运算。表达式中的数可以是 0.4、2、4、24、44。为了让表达式中只有数字 4，请在输出中用字符串.4、sqrt4、4、4!、44 代替上面提到的五个数。

Solution

题面翻译：你需要用尽可能少的 4 表示出一个正整数 n。表达式中可以使用加、减、乘、除、幂运算。表达式中的数可以是 0.4、2、4、24、44。为了让表达式中只有数字 4，请在输出中用字符串.4、sqrt4、4、4!、44 代替上面提到的五个数。

搜索即可通过，你可能需要一些剪枝技巧。

Problem

你有一个集合 $\{1, 2, \cdots, n\}$，你需要把它划分成两个集合，使得这两个集合中不含有勾股数组。如果正整数 a, b, c 满足 $a^2 + b^2 = c^2$，则 (a, b, c) 是勾股数组。

Solution

满分是论文，论文证明了 $n \leq 7824$。

Solution

满分是论文，论文证明了 $n \leq 7824$。
我们首先考虑如果知道了 n，我们如何给出一组合法的解。

Solution

满分是论文，论文证明了 $n \le 7824$。

我们首先考虑如果知道了 n，我们如何给出一组合法的解。

注意到勾股数组是可以快速生成的（利用高斯整数的推论给出本原勾股数），那么可以写成一个 3-SAT 问题~~（开始 NP 了）~~。

Solution

满分是论文，论文证明了 $n \leq 7824$。

我们首先考虑如果知道了 n，我们如何给出一组合法的解。

注意到勾股数组是可以快速生成的（利用高斯整数的推论给出本原勾股数），那么可以写成一个 3-SAT 问题~~（开始 NP 了）~~。

然后考虑如何拿高分，实际跑的时候发现有解得很快，无解的在 n 大的时候跑不出（你可能需要好一些的 3-SAT 板子来观察到这一点。

Solution

满分是论文，论文证明了 $n \leq 7824$。

我们首先考虑如果知道了 n，我们如何给出一组合法的解。

注意到勾股数组是可以快速生成的（利用高斯整数的推论给出本原勾股数），那么可以写成一个 3-SAT 问题~~（开始 NP 了）~~。

然后考虑如何拿高分，实际跑的时候发现有解得很快，无解的在 n 大的时候跑不出（你可能需要好一些的 3-SAT 板子来观察到这一点。

你可以尝试二分直到找到答案。

~~相信没有人会闲着没事做这些东西。~~

soj 1515

soj 1520

soj 1215

CF 1752A