

March 31, 2024

2024 年洛谷愚人节比赛 题解

2024 愚人节团队@Luogu

TOC

A / Nine Gates

B / Scoring a Problem

C / duoL tuO ecnetneS eht daeR

D / Always Accepted

E / Game

F / MBTI for Numbers

G / Again

H / You Need Help?

I / Problems about Linear Algebra

J / Just Some Superficial Examination

K / Checkmate

L / Just Some Superficial Examination Cont'd

M / A Unique Dream

注意到 $2^9 = 512$ ，结合标签里面的状压 dp，推出 $0 - 511$ 实际上需要在二进制下分离出 9 位。
对样例进行分离：

```
001001000
010010001
111111111
100010100
```

打开提示中剪贴板，给出了三个图，红色的部分是听的牌，第一个图的听三六，而第一个样例输入恰好只有第三位和第六位是 1，第二个图听二五九，样例第二个恰好只有这三位是 1，第三个图一到九都听，样例三的每一位都是 1。这三个有图，所以答案是 1。

不存在一种清一色牌型恰好听一五七，所以样例四答案是 0。

可以找到规律：对于输入，二进制分解后得到一个 9 位的 01 串，串中第 i 位是 1 表示听这张牌，第 i 位是 0 表示不听这张牌。如果存在一种清一色牌型的听牌集合与串表示的听牌集合相同，则 $a_k = 1$ ，否则 $a_k = 0$ 。（ k 是输入）

因为对称性，正序/倒序分解无所谓。

这个问题可以记忆化搜索或者状压 dp 解决。（所以算法标签真的是本题的算法标签）

这里提供一个简单的思路：设 $f_{\{i_1 \dots i_9\}}$ 表示一有 i_1 张，二有 i_2 张，以此类推，有了几个 aaa/abc/aa 形，其中 i 的每一位都 ≤ 4 ，所以状态只有 5^9 种。然后枚举 $i_1 + i_2 + \dots + i_9 = 13$ 的集合，再枚举 $1 - 9$ ，加上之后判断对应的 f 值是否为 5，如果为 5 则听这张。根据 $1 - 9$ 的结果，二进制压缩后将 a_k 设为 1。（ k 是压缩后结果）

std：

```
#include<bits/stdc++.h>
using namespace std;
const int N=1953125;
int f[N],g[512];
int main()
{
    //freopen("1.in","r",stdin);
    //freopen("1.out","w",stdout);
    memset(f,-1,sizeof f);
    f[0]=0;
    for(int i=0;i<N;i++)
    {
        if(f[i]==-1||f[i]==5) continue;
        int a[9],cnt=0,ii=i;
        memset(a,0,sizeof a);
        while(ii)
        {
            a[cnt++]=ii%5;
            ii/=5;
        }
        if(f[i]==0)
        {
```

```

        for(int j=0,tmp=1;j<9;j++,tmp*=5)
            if(a[j]+2<=4)
                f[i+tmp*2]=1;
    }
    else
    {
        for(int j=0,tmp=1;j<9;j++,tmp*=5)
            if(a[j]+3<=4) f[i+tmp*3]=f[i]+1;
        for(int j=0,tmp=1;j+2<9;j++,tmp*=5)
            if(a[j]+1<=4&&a[j+1]+1<=4&&a[j+2]+1<=4) f[i+tmp*31]=f[i]+1;
    }
}
for(int i=0;i<N;i++)
{
    int a[9],cnt=0,ii=i,tot=0;
    memset(a,0,sizeof a);
    while(ii)
    {
        a[cnt++]=ii%5;tot+=ii%5;
        ii/=5;
    }
    if(tot==13)
    {
        int val=0;
        for(int j=0,tmp=1;j<9;j++,tmp*=5)
            if(a[j]+1<=4&&f[i+tmp]==5) val|=(1<<j);
        g[val]=1;
    }
}
int T;
cin>>T;
while(T-->0)
{
    int n;
    cin>>n;
    cout<<g[n]<<endl;
}
return 0;
}

```

B / Scoring a Problem

By [E.Space](#)

样例输入是一些知识点的组合，容易想到对应的是 NOI 大纲的难度之和。样例 1 可以用来验证。

C / duoL tuO ecnetneS eht daeR

By [Indjy](#)

标题倒过来是“大声读出来句子”，所以把题目描述读出来然后录音倒放就可以听见“输出 1231212 ”，1231212 就是答案。

D / Always Accepted

By [E.Space](#)

你一定能通过这道题。

这句话的意思是，任何的提交，只要没有 CE，就会显示通过。但是显示通过不意味着拿到满分。如果你按照题面上的方式写一个 A+B 的话，那么可以得到 10 分。

正确做法是，叛逆一点。题面上写了“注意别超时哦”，然后你试一下交一个会 TLE 的程序，然后你就过了。本题任何 TLE 的提交都会得到满分。

E / Game

By [E.Space](#)

输出格式要求输出一个仅包含 `wasd` 的字符串。结合题面中写有一个 14×14 的屏幕，被分成了左右两半，试着交一发之后可以发现，这道题有 196 个测试点，分成两个子任务。评测结果中 WA 和 MLE 交错，看起来像一个迷宫。于是可以意识到这道题是要你玩一个迷宫游戏。剩下的部分就是靠在提交中摸索了。

这个游戏的显示方式是这样的：TLE 是玩家当前位置，MLE 是墙，WA 是空地，AC 是终点。地图大小是 25×25 的，当玩家试图向屏幕外围离开第 6 行、第 9 行、第 6 列、第 9 列时，地图会往对应的方向卷动，除非已经到了地图的边界。

但是，如果只是走到了终点，只能得到 50 分。地图上有 10 个 WA 的格子，上面的运行时间显示的是 5ms（其他都是 0ms）。每走到过一个这样的格子可以得到 5 分。其中有 3 个这样的格子被墙围住了，但是通过观察可以发现，有些墙上的运行空间上写着 x.00KB（x 的值在 6 到 9 之间，不同于一般的 0）。这些墙可以通过往墙上撞 x 次来打开。

另外，这个游戏每 4 个小时地图就会刷新一次，做的时候需要注意不要离刷新时间太过接近。地图由于当前时间相关的随机种子生成，赛后仍然会刷新。

小 E 的吐槽：这题自定义计分脚本长度总和达到了 49410B，差点就超出洛谷的限制了，我还压缩了好久才压到这个长度。

F / MBTI for Numbers

By [tiger2005](#)

本题的正确解法是：将 0-9 转化为对应的英文单词，然后数 e 的个数和 i 的个数。如果 e 的个数更多则答案是 e，i 的个数更多则答案是 i，否则答案为 pi。

本题追加的提示为：将题目中的称谓换为英文，提示选手从英文的视角想问题。

G / Again

By [VinstaG173](#)

写个从零开始解决此题的全过程。

首先要学会读简谱！这个我不会教，但应该挺容易学的。

然后我们去听听歌。歌名《Again》，作曲横山克即可搜到，是一首非常好听的纯音乐。

这里如果上网查简谱，可能会查到很多个不同版本，而且几乎都和音乐原版不一样，不知道是改编还是扒谱人扒错了。

事实上原版旋律扒谱后应该是这个样子：

アゲイン

横山 克

1=#F $\frac{4}{4}$

$\left| \begin{array}{ccccccc} \underline{\underset{\cdot}{3}} & \underline{\underset{\cdot}{4}} & \underline{\underset{\cdot}{5}} & 1 & \underline{\underset{\cdot}{5}} & \underline{1 \ 2} & \left| \underline{3 \ 1} \ \underline{\underset{\cdot}{5} \ 4} \ \underline{1 \ \underset{\cdot}{5}} \ \underline{\underset{\cdot}{2} \ \underset{\cdot}{1}} \right| \end{array} \right.$
 $\left| \begin{array}{ccccccc} \underline{3 \ 2} & \underline{3 \ 6} & \underline{6 \ 6} & \underline{3 \ 2} & \left| \underline{3 \ 2} \ \underline{3 \ 7} & \underline{7 \ 7} & \underline{3 \ 2} \right| \end{array} \right.$
 $\left| \begin{array}{ccccccc} \underline{2 \ 1} & \underline{2 \ 5} & \underline{5 \ 5} & \underline{2 \ 1} & \left| \underline{\underset{\cdot}{7} \ 1} \ \underline{2 \ 5} & \underline{5 \ 2} & \left| \end{array} \right.\right.$
 $\left| \begin{array}{ccccccc} \underline{3 \ 2} & \underline{3 \ 6} & \underline{6 \ 6} & \underline{3 \ 2} & \left| \underline{3 \ 2} \ \underline{3 \ 7} & \underline{7 \ 7} & \underline{\underset{\cdot}{1} \ 5} \right| \end{array} \right.$
 $\left| \begin{array}{ccccccc} \underline{\underset{\cdot}{2} \ \underset{\cdot}{1}} & \underline{7 \ \underset{\cdot}{1}} & \underline{\underset{\cdot}{1} \ 7} & \underline{5 \ 2} & \left| \underline{\underset{\cdot}{2} \ \underset{\cdot}{3}} & \underline{\underset{\cdot}{3} \ \underset{\cdot}{3}} & \underline{\underline{\underset{\cdot}{2} \ \underset{\cdot}{2} \ \underset{\cdot}{2} \ \underset{\cdot}{2}}} \parallel \end{array} \right.$

我们和题目中给出的简谱对比一下：

アゲイン

横山 克

1=G $\frac{4}{4}$

$\left| \begin{array}{ccccccc} \underline{\underset{\cdot}{3}} & \underline{\underset{\cdot}{2}} & \underline{\underset{\cdot}{5}} & 1 & \underline{\underset{\cdot}{5}} & \underline{1 \ 2} & \left| \underline{3 \ 1} \ \underline{\underset{\cdot}{5} \ 4} \ \underline{1 \ 3} \ \underline{\underset{\cdot}{2} \ \underset{\cdot}{1}} \right| \end{array} \right.$
 $\left| \begin{array}{ccccccc} \underline{3 \ 2} & \underline{3 \ 6} & \underline{6 \ 6} & \underline{3 \ 5} & \left| \underline{\underset{\cdot}{2} \ \underset{\cdot}{2}} & \underline{3 \ 7} & \underline{7 \ 7} \ \underline{3 \ 2} \right| \end{array} \right.$
 $\left| \begin{array}{ccccccc} \underline{2 \ 1} & \underline{\underset{\cdot}{5} \ 5} & \underline{5 \ 5} & \underline{2 \ 1} & \left| \underline{\underset{\cdot}{7} \ 1} \ \underline{2 \ 5} & \underline{5 \ \underset{\cdot}{1}} & \left| \end{array} \right.\right.$
 $\left| \begin{array}{ccccccc} \underline{3 \ 2} & \underline{3 \ 6} & \underline{6 \ \underset{\cdot}{1}} & \underline{3 \ 2} & \left| \underline{3 \ 2} \ \underline{\underset{\cdot}{2} \ 7} & \underline{7 \ 7} & \underline{\underset{\cdot}{1} \ 5} \right| \end{array} \right.$
 $\left| \begin{array}{ccccccc} \underline{\underset{\cdot}{2} \ \underset{\cdot}{1}} & \underline{7 \ \underset{\cdot}{1}} & \underline{\underset{\cdot}{1} \ 7} & \underline{5 \ \underset{\cdot}{5}} & \left| \underline{\underset{\cdot}{2} \ \underset{\cdot}{2}} & \underline{\underset{\cdot}{2} \ \underset{\cdot}{3}} & \underline{\underline{\underset{\cdot}{2} \ \underset{\cdot}{2} \ \underset{\cdot}{2} \ \underset{\cdot}{2}}} \parallel \end{array} \right.$

事实上这个过程不需要扒谱，只需要知道简谱大概的含义，错的位置错的都很离谱。要么就音高相差甚远，要么就能明显发现曲子中音高和相邻音相同而谱子上却不同。

考虑到在原曲中找到适合提取答案的音其实是不容易的，因此答案很有可能藏在错误的音符中。另外，看见错误的音符横跨三个八度，和整个旋律的进行明显非常不搭调，出题人一定不会毫无意义地把错误音符设置成这个样子。

提取出来的错误音符为（设 + 为上加点，- 为下加点）：

1. 第 1 小节第 2 个音 2；
2. 第 2 小节第 6 个音 3；
3. 第 3 小节最后一个音 5；
4. 第 4 小节第 1 个音 2-；
5. 第 5 小节第 3 个音 5+；
6. 第 6 小节最后一个音 1++；
7. 第 7 小节倒数第 3 个音 1+；
8. 第 8 小节第 3 个音 2-；
9. 第 9 小节最后一个音 5+；
10. 第 10 小节第二个音 2-。

发现每个小节恰好有一个，很有规律！很有可能是对的！

然后问题来了，这一串音符是什么东西呢？输出的应该是**大写英文字母**。怎么把它变成大写英文字母呢？

事实上这里会发现题面中的谱里还有一个不对的地方：1=G。这点如果上网查了简谱可能会发现，也可能不会发现。如果真的很懂音乐可能会发现，不过不作要求。

另外在谱子里甚至特意把标题写成了日文片假名，整个谱子上只有 G 这一个大写英文字母了。

总之，可能可以想到用 1=G 转换成英文字母。如果会一点音乐可能会知道除了简谱常用的首调唱名法还有一种固定唱名法，以及音名，1=G 中的 G 是音名的意思。但是这样提取出来的可能是形如 A5 的音名，而不是纯粹的大写英文字母。

这一步可能有一点和出题人对电波。事实上我们提取的不只是音名，而是把 1 看作 G 后将音符按音高顺序与字母进行对应。会发现 2- 就是 A，而 1++ 也只有 U，所以很有可能是对的。

这里还有一件巧事，G 调的 7 事实上是 #F，而 F 音应该是 b7，不过正好没有碰到需要 F,M,T 这些对应音名为 F 的音符的字母，否则我还不知道该用 7 还是用 b7。

总之用这种方法进行提取，可以得到字符串 HIKARUNARA。就算不知道这是什么意思，随便在什么地方都能查到这是《四月是你的谎言》的第一首片头曲（OP），也是一首很好听的歌。Again 也是出自这部作品的 OST，因此猜测这就是正确答案，提交即可通过。

H / You Need Help?

By [E.Space](#)

题面中一大堆和“帮助”有关的字眼提示要去往帮助中心。点到“主站操作指南”标签可以看到有一个叫做“I need help!”的页面（比赛结束后可能会删掉）。页面提示你答案是 findfromthepast。输出该答案可以获得 50 分。根据答案含义的提示，你需要从过去寻找答案。于是需要找到 GitHub 仓库里这个页面的上一版本，即可得到答案 walkthroughofamagicwildcard。

I / Problems about Linear Algebra

By [E.Space](#)

页面翻到最底下，可以看见让你把题目背景中的内容翻译成英文。三个加粗部分的翻译分别为 elimination, space 和 problem。这意味着需要从 [E.Space](#) 的题目中寻找答案。点开 [E.Space](#) 的个人主页可以找到一些题目。

子任务 1 对应着《数正方体》一题。提示是题面中的“画在纸上”以及数据范围里的 $1 \leq n, m \leq 100$ 和 $A_{\{i,j\}} \geq \max\{A_{i+1,j}, A_{i,j+1}\}$ 。答案是字符画大小的两个维度。

子任务 2 对应着《拯救还是毁灭》一题。提示是题面中的“拯救世界”的字眼和 $n \times n$ 的矩阵。这道题中当 n 给定时，最坏情况下的输入需要至少 $2n(n-1)$ 次操作才能还原矩阵。所以需要输出 $2n(n-1)$ 。

子任务 3 对应着《神秘序列》一题。提示是题面的序列、“操作”，还有数据范围中的 $0 \leq a_i \leq i$ 。这道题的每次操作是找到第一个 $a_i = i$ 的位置，然后把 a_i 清零且把 a_1 到 a_{i-1} 的值全部加 1。所以倒回去就是每次把第一个 0 变成 i ，然后把前面的项全部减 1。注意要输出到最后一个非 0 项。

子任务 4 对应着《最大权独立集问题》(P9111) 一题。提示是数据范围中的 $1 \leq n \leq 400$ 以及题面中的“线性代数题目”字眼和整个题目模型。做法就是按照这个题题目模拟一次操作。

子任务 5 对应着《拧螺丝》。提示是“老板”的字眼和变量名。这道题题面上写着，“每隔 10 分钟老板就会来一次”，所以每小时老板会来 6 次。然后老板的收入就是收走的螺丝个数。通过样例可以发现，如果小 E 做完了一个模块，那么之后老板就没有收入了；以及小 E 需要在保证时间最短的前提下选择被拿走螺丝数量最少的方案。

J / Just Some Superficial Examination

By [E.Space](#)

题单在团队题单列表里。题面中写到要寻找题单中不和谐的地方。

题单 #1 中除了一个绿题其他题都是蓝题。所以不和谐的地方是这个绿题。

题单 #2 中除了一个题之外其他题的算法标签中都有贪心。

题单 #3 中除了一个题的题号是质数之外其他题的题号都是合数。

题单 #4 每个题的题目名称结尾都是一个数字。十个题分别是 1 2 2 4 5 6 7 8 9 0。所以不和谐的地方是第三题。

题单 #5 中的题目按照题目名称字典序排序后发现，首字母是 A 到 Z 且题号中的字母是 ABCD 循环。但其中有一个题不满足这个条件。

题单 #6 中除了一个题之外每个题的题目名称长度都是 7 个字母。

题单 #7 中除了一个题之外每个题的算法标签都有 3 个，且都是两个字。

题单 #8 的题目名称首字母连起来是 LUOGU APRIL FOOL'S DAY'S CONTEST。但第五题的首字母其实是 E。

题单 #9 的题号数字连起来是 14159265358979... 是 π 的小数部分。但有两位是错的。

题单 #10 的每道题题目名称中都有一个单词 Set，但其中一个是 Sets。

题单 #11 的题目按照题号排列后的难度是红橙黄绿蓝紫的循环，但有一个不对。

题单 #12 中除了一个题是 D 题，其他都是 B 题。

找到这些题目后要做什么呢？注意到几乎所有题单都包含了 26 个题（#4 是 10 个），考虑把不和谐的题在题单中的位置提取出来，用 a1z26 替换一下（#4 是数字 3），可以得到 the3rdletter。输出 the3rdletter 或 thethirdletter 可以得到 25 分。

然后考虑把这些挑出来的题目的题目名称列出来：


```
Bad Luck Island
Prime Gift
Mysterious Code
Rectangle Painting 2
Rooks Defenders
Cinema
Gift
Evil
Array
Similar Sets
Greedy Change
Odd Mineral Resource
```

按顺序提取第三个字母可以得到 `disconfirmed`。

K / Checkmate

By [VinstaG173](#)

签到题之一。

出题人挺喜欢写故事，但是写得都不好。

有个彩蛋：题目中的所有字母按照出现顺序连起来是 `TRIANGLE`，提示棋盘的排布方法。奇怪的变量名也是这个的一部分，而不是某种奇怪语言的“总数”和“黑”之类单词的首字母。

事实上是个很传统的诈骗题。

注意到规则中加粗部分，以及不能操作者败，我们发现三枚棋子两两相切时谁都动不了，只要把两枚黑子和一枚白子这样相切在一起后剩下的黑子全部用这两枚黑子挡住，那么先手一开始便动不了了。

有特殊情况： $L = 1$ 时显然 E 必须等于 0， $L = 2$ 时只要有黑子就一步必杀，所以也需要 $E = 0$ 。只需要特判 $L = 2$ ，其余情况输出 $L - 1$ 即可。注意开 `long long`。

L / Just Some Superficial Examination Cont'd

By [E.Space](#)

这题是 J 题的后续。

你需要在给定的题单里分别找到一个题填入原来那个不和谐的位置。

注意需要互不相同。

答案是唯一的。除 J 题题解中的描述外，#6 还要注意题号是按顺序的。

但是我忘了 #5 排序之前也是按顺序的了，谢罪。

M / A Unique Dream

By [E.Space](#) & [VinstaG173](#)

几个出题人 puzzlehunt 入脑了想出个 meta，然后就有了这个题。由于没有小题答案，正好题目标题都用的英文，于是就想从标题提取了。整个题几乎都是在比赛前赶 ddl 搞出来的。

小 E 向 VG 吐槽 Again 标题太短（这时 Game 的标题还是 Arcade Hall），而由于去年的《四月是你的谎言》在 G，因此今年也想把 Again 放在 G，因此在想可能的提取方式，正好 Again 中有恰一个字母 G，因此想到了提取标题中**唯一的**与题号相同的字母的位置。

最初小 E 排出了一个顺序后提取出了一堆 14，暴力查找单表对应后找到了 `monotonic beat`，但是 VG 不太喜欢。

经过了漫长的凑答案过程后 VG 凑出了一个可以单表为 `wanswerofmeta` 的排序，稍微改了两道题目的标题（Bonus:你能猜出原来的标题以及为什么要改吗?），但是感觉非常绝妙：

```
Nine Gates <a6>
Scoring a Problem <b12>
duoL tu0 ecnetneS eht daeR <c9> h1 d12
Always Accepted 12 <d14>
Arcade Hall c3 d5 <e6> h7
MBTI for Numbers <f5> e12
Again <g2>
You Need Help? d7 <h8> 11
Problems about Linear Algebra <i15>
Just Some Superficial Examination <j1> f14 c16 119
Checkmate <k5>
Just Some Superficial Examination Cont'd j1 f14 <119>
A Unique Dream <m12>

6 12 9 14 6 5 2 8 15 1 5 19 12
w a n s w e r o f m e t a
```

但是这个 `w` 实在太鸡肋了，一个字母也找不到合适的单词替代，甚至 `answer` 前面需要用 `an` 而不能用 `a`。

这时候的出题组还在想用 1~26 对字母 `a-z` 进行单表，fx 甚至提出对于 13 题每题只有在 100 分 AC 后能获得单表的两位字母，后来实在不知道怎么写自然的提示，VG 陷入无法使用这个耗费大量精力凑出来的答案的境地，但是还没有死心。

这时候救世主 tiger2005 出现并提出对单表进行一些美化，小 E 想到了可以微调一下题目的背景（把之前的“位置”改成了现在的“时刻”），于是讨论出了时钟的创意，VG 这时想到反正已经不是 26 个字母的单表了是否可以加入非字母字符，可能可以解决第一个字符的问题。

然而 Nine Gates 作为九莲宝灯的英文不适合改，在询问小 E Arcade Hall 是否适合更改并得到肯定答复后将其标题改为了 Game，然后小 E 决定了第一位的非字母字符为代码中代表 not 的 `!`，就有了最终版的答案：`!answerofmeta`，意为 not answer of meta。事实上这题并没有 puzzlehunt 中 meta problem 的性质，使用的是标题而不是答案，不需要完成其他题目，一开始就可以做，所以也可以说答案“不是 meta 的答案”。

最后 tiger2005 制作了漂亮的表盘，鸣谢 tiger2005。

另外鸣谢在此题出题过程中参与讨论的 WYXkk 和 minstdfx，他们提供了很多有意思的 idea，但最终因为各种原因没能实现，实在遗憾。

题外话：因为此题的原因，有点麻烦的麻将题被放在了 A，而签到题之一被放在了 K，不过巧的是其他比较简单的题一起放在了 BCD，或许一定程度影响了选手开题和榜的变化。