

2018 HIT@WH ACM Newbie Cup

分析 & 题解

分析

分析

基本信息

- Writer : 李旭
- Checker : 邵若忱、杨浩、高翔宇

命题思路：

1. 不像 2017 一样一题金到铁
2. 学了什么就考什么，保证每个题都有 C 标程
3. 每道题都能开一开，都能有一个基本思路
4. 主要考察思路和实现能力，经典算法偏少

分析

预期难度分级：

- Easy : C、 G
- Mid-Easy : A、 E、 H
- Mid-Hard : B、 D
- Hard : F

预计：

- Easy : 75% ~ 100% 通过所有问题
- Mid-Easy : 30% ~ 50% 能过至少一题， 5% ~ 25% 能过所有题
- Mid-Hard : 10% 能过至少一题， 极少有人通过两题
- Hard : ~~真的有人能过吗，你是不是看过叉姐 Slider~~

分析

最快解题：

- A：陈靖昌，13:42
- B：邹皓杰，13:39
- C：杨卓宸，13:11
- D：唐泓煊，13:22
- E：关舒文，16:13
- F：无人解出
- G：吴迪，14:55
- H：蒋元，13:42

分析

顽强拼搏奖：

- 刘方未，G 题，17:04

队伍情况分析：

- 109 人有有效提交
- 共 1405 份提交
- 共 255 份 AC 提交

分析

题目通过比：

- A : 8 / 221
- B : 13 / 152
- C : 109 / 314
- D : 4 / 177
- E : 1 / 1
- F : 0 / 41
- G : 32 / 110
- H : 88 / 397

分析

标程信息：

- A : 550ms / 2.3 KB
- B : 40ms / 819 Bytes
- C : 1ms / 285 Bytes
- D : 180ms / 588 Bytes
- E : 10ms / 317 Bytes
- F : 3500ms / 3.3 KB
- G : 20ms / 240 Bytes
- H : 1ms / 270 Bytes

题解

题解

TL;DR 版

- A : 两次排序，注意细节问题
- B : 观察规律，DFS 或者模拟位运算讨论中心点
- C : 签到题
- D : 走 1 步，间隔 1 次以上跑 K 步，到 D 点方案， $\text{dp}[D][2]$
- E : 展开式子，发现系数没有作用，讨论 n 奇偶性
- F : 倍增一次然后对格子编码做 FFT，之后统计答案
- G : $000n$
- H : `x + 20190001 if x < 20190001 else 20192018`

题解

Easy

题解

Easy

C. Cities: Skylines

感觉不用说吧？这个题不会做打回去重学 C 语言吧？

实在不行读读样例问题代码改改就能做吧？

```
long long a, b;  
scanf("%lld%lld", &a, &b);  
  
if (a < 0 && b < 0)  
    printf("%lld\n", -a * b);  
else  
    printf("%lld\n", a * b);
```

题解

Easy

G. Grand Theft Auto

前期榜歪了，中后期过的人开始变多了。

重点是理解两个运算律： $x \oplus x = 0, x \oplus 0 = x$ 。

理解了之后发现，我们只需要构造全是 0 的情况然后塞一个 x 进去就行。

极其容易得到这个构造： $0\ 0\ 0\ n$ 。

当然还有其他构造，例如： $1\ n + 1\ 1\ n + 1$ 或者分奇偶讨论（标程做法）。

题解

Mid-Easy

题解

Mid-Easy

A. Arcaea

大模拟 + 码农题，没想到把大家安排的不明不白的。

数据处理题，细节比较多：

1. 存储 ptt 使用 `double`，不要做取两位操作
2. 对 ptt 计算规则和数据理解，比如 ptt 下限为 0，给出 ptt 的顺序
3. 两次排序中，如何标记第一次排序产生的影响
4. 输出答案 `printf("%.2f\n", ans);`，补足 2 位

(P.S. 验题组三位银牌大佬平均 +2 :))

题解

Mid-Easy

E. Ever17 -the out of infinity-

给唯一通过的大佬点赞！

展开原式，能观察到具有如下形式：

$$ax^{bx^{cx^{dx^{\cdots}}}}$$

且这个幂塔共有 n 层，最外层为 0。

讨论这个幂塔，发现若有零点，零点只可能为 0。

题解

Mid-Easy

E. Ever17 -the out of infinity-

接下来讨论零点存在性。最终这个幂塔会具有的形式是：

$$0^{0^{0^{\cdots}}}$$

由于存在 $0^0 = 1, 1^0 = 0^1 = 1$ ，我们可以立即得知只有奇数个 0 堆叠成的幂塔才能使得其求值后为 0。

所以答案很简单了，如果有奇数层，则有且仅有一个零点：0，否则无零点。

系数没有任何作用。

输出解释只是吓吓大家，别打我

题解

Mid-Easy

H. Hacknet

~~你觉得出题人真的会把 AC 代码就放在题目里面给你交？~~

把函数复制下来自己跑一下样例应该就能发现规律：大于 20190001 就是 20192018，否则是 $x + 20190001$ 。

另一种提倡的观察方法是在函数内部用 `printf` 观察每一次递归调用的参数和返回值，然后用小输入去测试，也容易观察到递归过程的特性。

题解

Mid-Easy

H. Hacknet

不严格证明一下。

首先 $x < 20190001$ 的情况我就不讲了，我们来看 $x = 20192019$ ，观察一下调用过程：

$$f(20192019) = f(f(20190000)) = f(20192018)$$

然后

$$f(20192018) = f(f(20189999)) = f(20182017) = \dots$$

这么递归下去最终会有

$$f(20190001) = f(f(20187982)) = f(20190000) = 20192018$$

题解

Mid-Easy

H. Hacknet

经过上面的观察， $x = 20190000 + b(1 \leq b \leq 2018)$ 的情况下，最终都会回归到 20192018。同时也有， $f(20192018) = 20192018$ ，20192018 是这个函数的不动点。

对于任意 $x \geq 20190001$ ， x 总具有这样的形式： $x = 20190000 + 2019a + b$ ，根据递归过程， $2019a$ 会被消耗，最终剩下的就是 $f(\cdots f(20190000 + b) \cdots)$ ，这样的形式内部求值会得到 20192018，最终求值结果也就是 20192018。

题解

Mid-Hard

题解

Mid-Hard

B. Battlestations Pacific

观察 S_3 :

DDCDDCC

中心点必为 **D** , 前后几乎是一致的 , 除了前后部分中心点刚好翻转 , 前面为 **D** 后面为 **C** 。

又由于变换规则 , 每一个新串的前半部分都是原串 , 也就是说 , S_1 的第一个字符和 S_J 的第一个字符是完全一致的。

所以 , 我们只需要讨论到 S_{62} 即可 ($2.33 \times 10^{18} < 2^{62}$) 。

题解

Mid-Hard

B. Battlestations Pacific

接下来怎么做，生成这个完整串吗？

完成保存这个串需要 $2^{62} \approx 4 \times 10^{18}$ 个字节，大概是 3.6×10^6 TB 的内存，根本不可能。

实际上，根据我们的观察，发现我们只需要讨论 K 的位置即可，如果 K 在中心点，直接输出当前中心点即可，如果 K 在前面，我们就从前面找，否则从后面找。

这样的实现很简单，可以 DFS 可以位运算枚举，复杂度 $\mathcal{O}(\log K)$ ，完全足够。

题解

Mid-Hard

D. Dragon Quest

榜被带歪了。实际上这个题并不是直接就能做的。

裸动态规划题目。

本题的模型可以转化成如下样子：我们从原点出发，每次可以向右走 1 格，间隔 1 次以上可以跑 K 步，有多少种方案到达第 D 格？

我们考虑造成的总伤害为 d ，那么我们是怎么到达这个 d 的？

- 从 $d - 1$ 使用一次普通攻击
- 从 $d - k$ 使用一次技能

上面这个几乎就是模型的全部。但是如果我們这么做，会有一个問題：怎么保留间隔一次以上这个限制？

题解

Mid-Hard

D. Dragon Quest

间隔一次以上这个限制我们可以这么考虑，我们把每一次攻击拆成使用普通攻击到 d 的方案和使用技能攻击到 d 的方案，那么：

- 使用普通攻击到 d 的方案数是 $d - 1$ 中使用

- i. 普通攻击

- ii. 特殊技能

攻击的方案数之和。

- 使用技能攻击到 d 的方案数是 $d - k$ 中使用普通攻击的方案数。

那么最终答案就是 $\text{dp}[D][0] + \text{dp}[D][1]$ 。复杂度 $\mathcal{O}(D)$ 。

题解

Mid-Hard

D. Dragon Quest

本题有两个 Bouns Problem：

- 模数修改为 $10^9 + 7$ ，除了动态规划还有什么做法？
(张翼乘风同学的做法，但是本题不能用)
- 我想询问一个区间 $[L, R]$ 之内的方案数和，怎么做？

以上两个题之后都将会追加到校 OJ 上，有兴趣的同学可以做做看。

题解

Hard

题解

Hard

F. Fire Emblem

给你们 8s 结果真的有人以为暴力能过啊.....

观察题目，意识到统计所有点对的复杂度是 $\mathcal{O}(N^4)$ ，时间限制内绝对是过不去的。

考虑如下的式子：

$$D_{x,y} = \sum_{i=1}^N \sum_{j=1}^N C_{i,j} C_{i+x,j+y}$$

其中 $D_{x,y}$ 代表水平距离是 x 以及垂直距离是 y 的点对个数。 $C_{i,j}$ 代表点 (i, j) 的点的数目。

题解

Hard

F. Fire Emblem

我们就把题目转化成了一个乘积的形式。接下来，我们对点做一下编码：

$$D_{xN+y} = \sum_{i=1}^N \sum_{j=1}^N C_{iN+j} C_{(i+x)N+j+y}$$

再做一下变形：

$$D_{xN+y} = \sum_{i=1}^N \sum_{j=1}^N C_{iN+j} C_{iN+j+(xN+y)}$$

题解

Hard

F. Fire Emblem

变量替换一下：

$$D_A = \sum_{i=1}^N \sum_{j=1}^N C_B C_{B+A}$$

这样就变成了一个一维多项式的形式了，我们可以方便的写出生成函数形式：

$$f(x) = \sum_{i=1}^N \sum_{j=1}^N C_B x^{-B} \cdot C_{B+A} x^{B+A}$$

这样， x^A 的系数就是 D_A 。

题解

Hard

F. Fire Emblem

当然，上面的多项式需要计算仍然是枚举两个多项式所有项，复杂度依然是 $\mathcal{O}(N^4)$ ，这里我们使用 FFT 加速，就可以做到 $\mathcal{O}(N^2 \log N)$ 的复杂度，最后统计一下答案即可。

有一个细节没有提：这么做的情况下，如果两个点跨行了，那么 $D_{x,y}$ 编码后的计算会出现问题，我们可以倍增一下，也就是在原图右边拼一个新的大小为 N^2 的全为 0 的图，这样如果出现跨行，必定是右侧的 0 点和左侧的点组合，对答案贡献值为 0，就不会产生影响了。

结果和成绩我们会尽快统计

Thank you for participation!