2024团队赛5 题解

首先题目难度跟植物/僵尸强度呈正相关，~~如果虎扑评分是对的~~

A - 小盆菇

判断输入串的前19个字符是不是"You are right, but "，是则输出AI，否则输出Human

G - 旗帜僵尸 [洛谷U461007]

分类讨论，手模样例可知

棋子数量为 1 时，先手必败

棋子数量为 2 时，先手必胜

棋子数量为 3 时，先手必败

棋子数量为 4 时，分类结果如下：

```
.....
oooo.
.....
.....
.....
```

```
.....
oo...
oo...
.....
.....
```

以上两种情况为先手必败

```
.....
oo...
.oo.
.....
.....
```

```
.....
ooo..
.o...
.....
.....
```

```
```

```
.....
ooo..
..o..
.....
.....
```

以上三种情况先手必胜

F - 西瓜香蒲 [Gym105257C]

用图论建模，将i 和a_i 连边。手玩一下样例，发现若编号$\leq n$,出度都为 1; 编号$> n$,出度都为 0 (既然有$2n$个座位，那么肯定坐不满，也就是说有些点既没有出边又要要求联通) 。 所以建模后的图无非形成了两种结构：**环和有向树。** 分类讨论，对于环：答案为环上节点个数，因为环上所有的点可以选，但是环外的点无论如何都不能到自己心仪的位置。可以用拓扑排序求解，没有标记过的点也就是环上的点。

对于有向树：找一条最长链，链上的所有点都可以到自己心仪的位置。求法可以用先建反图， DFS 跑一遍最大深度即可。

H - 西瓜坚果 [Gym105257G]

(https://www.luogu.com.cn/article/o8hlzo76)

这题可以先暴力打标，看一下规律，以 $x=9$ 为例：

0	10	20	30	40	50	60	70	80	90
1	11	21	31	41	51	61	71	81	91
2	12	22	32	42	52	62	72	82	92
3	13	23	33	43	53	63	73	83	93
4	14	24	34	44	54	64	74	84	94
5	15	25	35	45	55	65	75	85	95
6	16	26	36	46	56	66	76	86	96
7	17	27	37	47	57	67	77	87	97
8	18	28	38	48	58	68	78	88	98
9	19	29	39	49	59	69	79	89	99

蓝色的就是没有消失的数字，观察一下，可以发现，这些数字组成的新的数列就是九进制数。

再来看一下 $x=8$：

0	10	20	30	40	50	60	70	80	90
1	11	21	31	41	51	61	71	81	91
2	12	22	32	42	52	62	72	82	92
3	13	23	33	43	53	63	73	83	93
4	14	24	34	44	54	64	74	84	94
5	15	25	35	45	55	65	75	85	95
6	16	26	36	46	56	66	76	86	96
7	17	27	37	47	57	67	77	87	97
8	18	28	38	48	58	68	78	88	98
9	19	29	39	49	59	69	79	89	99

不难发现，把红色部分和绿色部分互换就是九进制数了，观察一下，其实把绿色部分中大于 x 的数位减 1 就是红色部分了，所以最后得出结论：

- 当 $x=9$ 时，把 n 看作九进制数，转化成十进制数即可。
- 当 $x\neq 9$ 时，先把 n 中所有大于 x 的数位减 1，然后重复上一步的操作就可以了。

最后不要忘了自然数含 0，所以最终答案还要加 1。

L - 冰瓜大喷菇 [CodeForces - 1543D1]

$x\oplus z=y\Longrightarrow x\oplus y=z$ 所以我们可以想一种方法让前面的询问不会对现在的询问产生影响。 我们可以考虑顺序枚举 1 到 n ,每次输出 $i\oplus(i-1)$, 当询问到第 i 次时原先的 x 变为：
$x\oplus(0\oplus 1)\oplus(1\oplus 2)\cdots\oplus((i-1)\oplus i)$ 根据异或运算的结合律化为：
$x\oplus 0\oplus(1\oplus 1)\oplus(2\oplus 2)\cdots\oplus((i-1)\oplus(i-1))\oplus i$

$\because x\oplus 0=x$ and $x\oplus x=0$

$\therefore$ 原式$=x\oplus i$

当$x=i+1$ 时会在枚举到 $i+1$ 时猜中。主要就是一个转换和抵消的想法，还是不难想的。

B - 钢刺坚果王 [Codeforces - 1512G]

此题关键在于用欧拉筛筛因数和。

设数i的因数和为$f(i)$,当前第二重循环 (可参考欧拉筛代码)枚举到的素数为p_j ,则分为三类 1.i 为素数，$f(i)=i+1$ 2.i 不可被 p_j 整除，$f(i\times p_j)=f(i)+f(i)\times p_j$ 。因为乘上p_j 就使原数的因子数增加了一倍，增

加的因子是原数每个因子分别乘上p_j 。 3.i 可被 p_j 整除，$f(i \times p_j) = f(i) + (f(i) - f(i/p_j)) \times p_j$ 。因为 i/p_j 中的每个因子乘上 p_j 都 会造成因子的重复计算，所以要去掉重复出现的因子。

C - 冰瓜香蒲 [洛谷 - P1966]

给定两个序列$a_{1\cdots n}, b_{1\cdots n}$每次操作可将任意序列中相邻两位置交换。

求：要使$\sum(a_i - b_i)^2$ 取最小值，所需最少操作次数。

$1 \leqslant n \leqslant 10^5$,$0 \leqslant a_i, b_i < 2^{31}$,$\forall 1 \leqslant i < j \leqslant n, a_i \neq a_j \land b_i \neq b_j$ 。

发动你的人类智慧，你的潜意识告诉你两列火柴排名相同的火柴排在同一行时就是答案

$\sum(a_i - b_i)^2 = \sum(a_i^2 + b_i^2 - 2a_ib_i)$
上式取最小值等价于 $\sum a_ib_i$ 取最大值。

若$a_i \leqslant a_j$, $b_p \leqslant b_q$,

则 $(a_ib_p + a_jb_q) - (a_ib_q + a_jb_p) = (a_i - a_j)(b_p - b_q) \geqslant 0$, 故 $a_ib_p + a_jb_q \geqslant a_ib_q + a_jb_p$.

综上，应使 a, b内部相对顺序一致。

线性代数中有讲：交换一个序列相邻两个位置，该序列的逆序对改变 1

比较直观地，要使操作次数最小，有一种可行方案是：只对 b 进行操作，使之内部相对顺序与a一致。 因为是相对顺序，所以我们可以先进行离散化，问题变为使 a, b 完全一致。 设$c_{a_i} = i$,则b_i应该最终移到 b 中位置c_{b_i}。

于是我们再进行一波处理：$b_i \leftarrow c_{b_i}$ 。 处理后，问题变为使 b_i 升序。 此时的操作次数即为 b 的逆序对数，可用归并或树状数组求出。

D - 寒冰仙人掌 [洛谷 - U457368]

dp 解法

考虑 $dp_{i,j}$ 表示 i 个人有总共对应了 j 个不同的礼物的方案数，新加进来的礼物可以被这 i 个人之一所对应，移除这一个礼物之后可能还有 i 个人有至少一个礼物，也可能只剩 $i-1$ 个人有至少一个礼物

$dp_{i,j} = i \times (dp_{i,j-1} + dp_{i-1,j-1})$

当然 $dp_{i,i}$ 答案为 $fac[i]$，即 i 的排列数量

组合数学解法：

这个问题是一个经典的组合数学问题，可以通过容斥原理 (Inclusion-Exclusion Principle) 来解决。

首先，不考虑每个人至少分到一个苹果的限制，每个人可以得到任意数量的苹果，那么总共有m^n种分配方法

(每个苹果都有 m 种分配的可能性)。

然而，这样计算会包括一些不符合题目要求的情况，即有些人可能一个苹果也没有分到。我们需要从总数中减去

这些情况。

1.当有一个人没有分到苹果时，相当于只有$m-1$个人在分n个苹果，这样的分配方法有$(m-1)^n$种。由于

有 m 个人，所以这样的情况总共有$m \times (m-1)^n$ 种。

2.但是，我们在第一步中减去的情况中，有些情况在第二步又被减去了两次(比如两个人都没有分到苹果的情况),所以我们需要再加回来。当有两个人没有分到苹果时，相当于$m-2$个人分n个苹果，这样的分配方法有$(m-2)^n$ 种。这样的情况总共有$\binom{m}{2} \times (m-2)^n$ 种。

以此类推，我们可以得到以下的递推公式：

$$Total = m^n - \binom{m}{1} \times (m-1)^n + \binom{m}{2} \times (m-2)^n - \binom{m}{3} \times (m-3)^n + \ldots + (-1)^{m-1}\binom{m}{m-1} \times 1^n$$

如果$m=n$,则最后还需要加上$(-1)^m$的情况，即每个人都恰好分到一个苹果的情况，这种情况只有 1 种。

$$Total = \sum_{k=0}^{m-1}(-1)^k\binom{m}{k}(m-k)^n$$

如果 $n=m$,则总数还要加 1。

J - 影流窝瓜王 [Gym - 105257L]

首先一个关键的观察，如果一个数是k进制下的 LNC 数，则这个数在k进制下的各数位，除了前导零 外，不能包含其它的0。一个特殊的情况就是末尾不能为0。 假如先手时石子个数末尾为 0,则一定拿完后石子个数末尾非0。而后手可以拿走末位个石子来使末位恢复到0。如此操作，先手最后会遇到没有石子可以拿的局面。因此只要石子个数末尾为0，则先手必败， 否则先手必胜。 考虑解决原问题，我们发现只需要找到一个k,使得$n \not\equiv 0 \pmod k$,暴力从小到大枚举这个k时间复 杂度是低于$O(\log n)$ 的。

时间复杂度上界为$O(T\log n)$。

E - Cupid魅惑菇射手 [洛谷 - U462454]

由于$\max \times \min = a_u \times a_v$,这部分是可以简单计算的，答案是：$(\sum_{u \in subtree(i)} a_u)^2$.

考虑$\max(a_u, a_v) \times |a_u - a_v|$ 其实是 $\max^2 - \max \times \min$ 。 剩下的问题是 $\sum_u, v \in subtree(i)(\max(a_u, a_v))^2$. 对于子树问题，可以想到按照每一条边$(u, v)$ (v是u的某个儿子)依次加入。具体过程为： • $subtree(u)$初始为u。 ·计算$subtree(v)$和当前$subtree(u)$之间点对的答案。(跨越u节点的部分)。 ·把$subtree(v)$子树内的答案直接累加。(不跨越u节点的部分)。 • $subtree(u) \leftarrow subtree(u) + subtree(v)$ (将v的子树加入到u中)。 对于第四步容易想到启发式合并/线段树合并。 如果采用启发式合并，复杂度是$O(n\log^2 n)$ 的。如果采用线段树合并，复杂度是$O(n\log n)$ 的。当 然也可以选择 dsu on tree,复杂度$O(n\log^2 n)$.

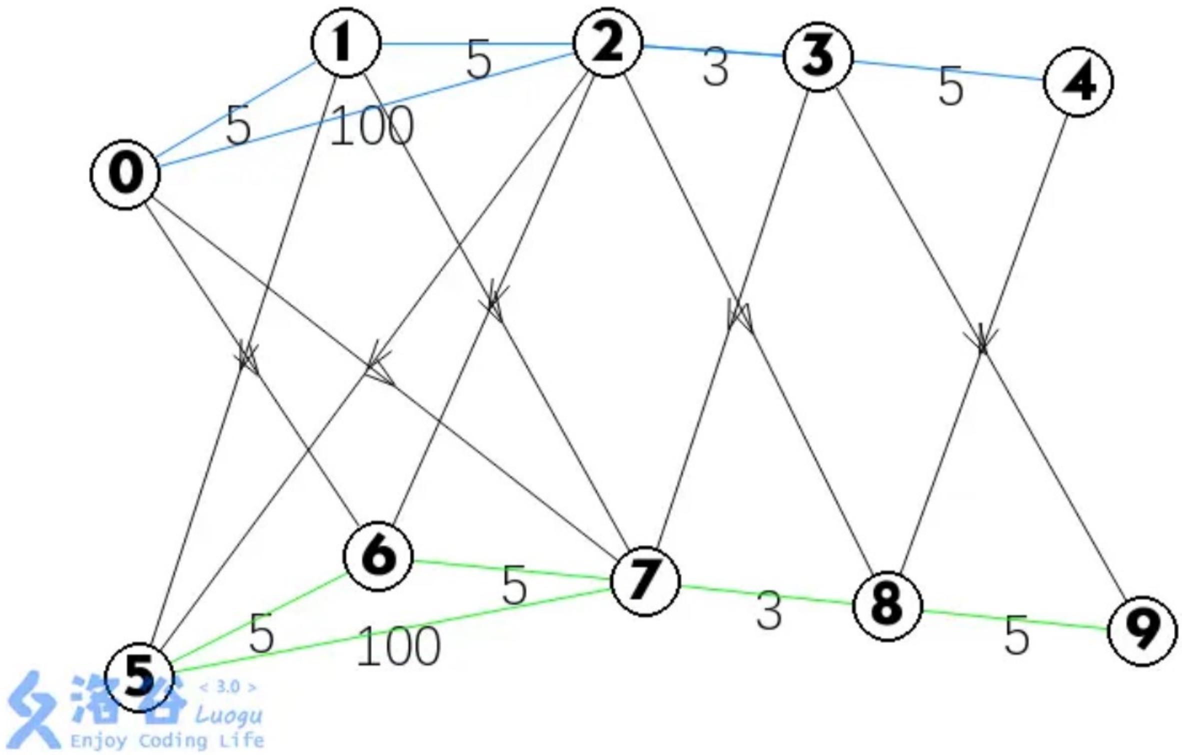
I - 红眼伽刚特尔 [洛谷 - P7109]

思路貌似太难，建议直接看 https://www.luogu.com.cn/article/glj03eal

K - 向日葵女王 [洛谷 - P4568]

套路题，分层图。

以样例为例（使用 @EternalAlexander 这位dalao的OI Painter绘制）：



各层内部正常连边，各层之间从上到下连权值为0的边。每向下跑一层，就相当于免费搭一次飞机。跑一遍从s到$t+n*k$的最短路即可。

M - 宾利巨人 [Codeforces - 1981D]

首先考虑两个数相乘不同，我们发现两对不同质数相乘一定不同。所以我们不妨只使用质数来构造数列，这一定比使用其它数更优。考虑m个质数最多能构造出多长的数列。我们将每个质数看成一个点，将两个质数的乘积看作这两个质数的边，题目中构造数列就变成了找到图上一条通路，要求相邻乘积不同的约束变成了一条边只能经过一次。注意每个点都有自环。 我们发现这是欧拉通路，考虑欧拉通路的判定条件。 ·当m 是奇数时：所有点度数为偶数，因此存在欧拉回路，路径长度就是：$\frac{m(m-1)}{2}+i+1$。加的 i 是自环边。 ·当m是偶数时：所有点度数为奇数，但是无所谓，我们可以隔一条边删一条边，删掉1-2,3- 4,5-6...这$\frac{m-2}{2}$条边，使得只剩下两个点度数为奇数。路径长度：$\frac{m(m-1)}{2}+i+1-\frac{m-2}{2}$. 在求出最少需要的点数后，我们直接在图上跑欧拉通路即可。