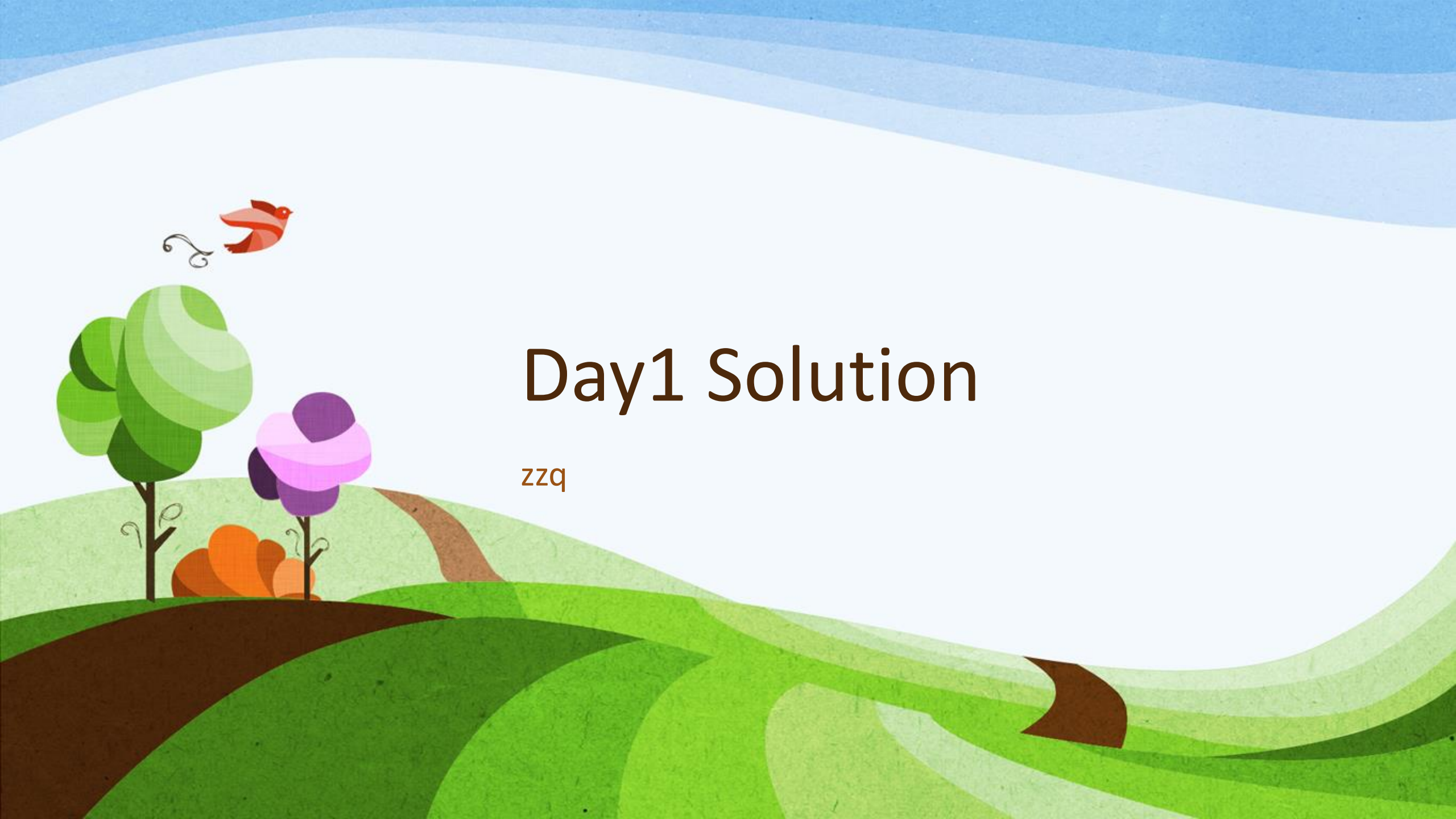

Day1 Solution

zzq

前言

- 每题讲题前会有吐槽环节，要喷出题人的可以排队上来喷
- 大家看一下自己的得分，特别是第三题的每个点得分，等会儿可能会点通过某些点的人上来分享

str

吐槽环节

str

- 先考虑字符集为2的情况，那么相当于原串和每位取反的串等价。
- 那么我们只要算出 原串 和 每位取反的原串 共有多少个不同子串，除以2就是答案。
- 查询若干个串中有多少个不同子串可以用广义后缀自动机，这里不赘述。

str

- 一种很自然的推广是枚举4个字符的4!种排列并计算会得到多少个不同子串，然后除以4!。但是这个做法有点问题，例如A只会得到A,C,G,T这四种串，得不到4!个串。
- 考虑改进这个做法，我们从1到4枚举一个值s，然后从4个字符中选出s个对应到1..s，计算这样不同的子串个数。例如当s=2时枚举出{A,C},{A,G},{A,T},{C,G},{C,T},{G,T}，对应到{1,2}或{2,1}。不在选定的s个字符中的字符不能选择，那么就直接把串割成两半就行了。例如原串是ACG，那么如果枚举出{A,G}对应{1,2}，就插入串1和2。注意这里是枚举完所有这C(4,s)*s!个方案之后算总共的不同子串个数。

str

- 考虑现在一个串的贡献。假设一种子串（同构意义下）有t种不同的字符，对于t<=s，会有贡献C(s,t)*t!（即选出1..s中这些字符占的，再把这些字符排列）。我们从小到大枚举t算出这样串的个数，然后对于s>t扣除这些串的贡献即可。

wlk

吐槽环节

wlk

- 考虑令f[i][j][k]表示大象在i号点，当前串的后缀最长匹配了a串长度为j的前缀，匹配了b串前k个字符，期望要多少步才会停止行走。
- 直接消元是O((n|a||b|)^3)的，考虑优化。
- 注意到f[?][?][k]只会从f[?][?][k]和f[?][?][k+1]转移，考虑从大到小枚举k，这样f[?][?][k+1]就是已知的了，就只要对f[?][?][k]进行消元了。
- 这样复杂度就变成了O((n|a|)^3|b|)的。

cal

吐槽环节

cal1

- 相信大家都会做。

```
input 1
pow 2 A[1] 100000
output A[2]
```

cal2

- 写一个循环结构。
- 以一个变量作为循环变量，每次+1，并且读入A[当前值]这个位置，如果它等于输入的n就跳出循环。接下来每次输出A[当前值]，然后-1，如果它等于0就跳出循环。
- -1就+(MOD-1)就可以了。

cal3

- 求出前缀积之后除一下就可以了，众所周知x的逆元就是x^(MOD-2)。
- 循环结构大同小异。

cal4

- 求出(-1)^a，如果是1就输出1，否则输出0。

cal5

- 不存在三次单位根。
- 注意到评分参数很松，考虑复杂度更差一些的做法。
- 考虑实现将a二进制分解，即是要每次判断是否是偶数，如果是奇数就-1，然后每次/2，直到=0。再开一块空间和一个循环变量记下分解结果，之后倒过来还原a就是要每次*2，如果记录下来这位是1就+1。只要在这个过程中实现mod 3就行了。这样就只会涉及到每次把一个[0,5]的数mod 3。
- 直接写if(x==3||x==4||x==5) x-=3;可能会被卡常，但是我们可以在A[0..5]中存入mod 3的值，直接查表就行了。
- 注意如果二进制分解写的不好可能要特判a=0。

cal6

- 和上一个点基本一样，只不过改成了两个数。

cal7

- 这次是把一大堆数加起来，但是好像长度限制是O(n)的?
- 考虑利用原根g=3，那么算a1+a2+...+an，我们可以转化为g^a1*g^a2*...*g^an。乘和快速幂都只要一条指令，那么我们只要想办法做一个离散对数就可以了。
- 考虑BSGS。由于MOD-1=10000120=520*19231，考虑预处理(g^520)的0..19230次方并存在A的对应位置，这个可以简单地用一个循环预处理。接下来我们就只要每次*g，直到找到A中一个不是0的位置。接下来我们每次/g，同时把指数-1，直到还原成原来的数。注意指数=0的时候指数-1要为MOD-2而不是MOD-1。

cal8

- 求最大公约数。
- 欧几里得算法并不是那么容易实现，但是我们可以使用stein算法。实现stein算法只需要实现判断奇偶性、比大小、加减法。比大小也只要分解出来比较最高位就可以了。

```
def gcd(a,b):
    sf=1
    while a%2==0 and b%2==0:
        sf*=2
        a>>=1
        b>>=1
    while a%2==0:
        a>>=1
    while b!=0:
        while b%2==0:
            b>>=1
        if a>b:
            a,b=b,a
        b-=a
    return a*sf
```

* 可参见 https://en.wikipedia.org/wiki/Binary_GCD_algorithm

cal9

- 判断一个数是否是质数。
- 认真观察评分参数发现居然是O(log)级别，这哪够分解质数。
- 考虑打表。先把输入二进制分解，然后建一个trie状的结构。每个节点会判断某一位的值并跳转到对应的孩子节点的标签，到了叶子就输出是否是质数。
- 类似这样：

```
label a
if A[0] 1 aa
if 1 1 ab
label aa
if A[1] 1 isprime
if A[1] 0 notprime
label ab
if A[1] 1 isprime
if A[1] 0 notprime
```

- 如果代码长度超长了可以加入在整个子树状态相同时不再递归直接输出，大概能短50%左右。

cal10

- 注意到只有output的输出个数=常数个数，而只用output又无法完成任务，那么一定需要一个循环结构。
- 考虑只用一种常数。一种很自然的想法是：

```
add 1 1 1
output A[1]
```

- 但是如果把A[1]用作循环变量，A[1]的值就会改变。但是我们可以把A[2]也就是A[A[1]]用作循环变量！

cal10

```
add 1 1 1
label bg
output A[1]
output A[1]
...
output A[1]
if A[A[1]] 1 out
add A[1] A[A[1]] 1
if 1 1 bg
label out
```

讲完了

谢谢大家