The Preliminary Contest for ICPC Asia Xuzhou 2019

Who is better?

一道水的很认真的数学模板题。任性而生硬的扩展中国剩余定理+斐波那契博弈。

首先是扩中的推导：

我们知道，中国剩余定理可以解决多个同余方程求解的问题，但是会有ai和aj互质的条件限制。

然而本题并没有这样的限制。

因此就引出了扩展中国剩余定理：

假设只有两式时： x ≡ r1 (mod a1), x ≡ r2 (mod a2)

```
    因此： x = k1 * a1 + r1 , x = k2 * a2 + r2

    然后:  k1 * a1 + r1 = k2 * a2 + r2

           k1 * a1 ≡ r2 - r1 ( mod a2 )

           exgcd ( a1 ,a2 ,k1 ,k2 )可求得a1,a2,k1

然后求得： x = a1 * k1 + r1  此时x只是满足上述两式的一个特解

得通解： x0 = k * lcm (a1,a2) + x

即合为一式： x0 ≡ x ( mod lcm (a1,a2) )

由数学归纳法： ans即为满足所有式子的最小正整数解
```

然后求出了ans也就是题干中要求的n，再来看这个博弈游戏。

大佬的博客讲解的很清晰了：https://blog.csdn.net/dgq8211/article/details/7602807

so easy

这里提供标程的解法

q的值比较小，所以解题应该从q入手

用并查集模拟实现一个链表

用map模拟并查集，初始时每个点的父亲指向后面第一个可用的点。

当删除一个点i时，令x的父亲等于x+1的父亲

查询时直接输出 x 的父亲

https://paste.ubuntu.com/p/ZfhHwtnDTx/

Buy Watermelon

- 题意：给定一个整数w表示西瓜的重量，切成两半之后每一部分都是2的倍数公斤。如果满足要求输出YES，否则输出NO
- 题解：判断w是否能被2整除，另外还需要考虑w的值不能为2（分成两半之后各自为1，不是2的倍数）。

Carneginon

由于$q*(|S|+|T|)\leq 1e7$, 对于每一次询问，用KMP判断即可

- 如果 $|T|>|S|$, T为模式串，利用KMP判断T是否为S的子串。若是，输出`my child!`；否则输出`oh, child!`
- 如果 $|T|<|S|$, S为模式串，利用KMP判断S是否为T的子串。若是，输出`my teacher!`；否则输出`senior!`
- 如果 $|T|=|S|$, 直接判断两者是否相等。若相等，输出`jntm!`；否则，输出`friend!`

XKC's basketball team

- 题意：给定一个序列，从每一个数后面比它大至少 m 的数中求出与它之间最大的距离。如果没有则为 -1。
- 题解：从后向前维护一个递增的队列，从后往前遍历，若当前的数大于队尾就进队，否则从该队列中二分找最小的比自己大至少 m 的数，二者之间的距离即为答案。
- 若当前数小于队尾，那这个数一定没有队尾的数优，因为它既比队尾的数靠前，又比它小。
- 时间复杂度 $O(nlogn)$ 。
- 此题也可以用ST表+二分 等方法写出。

Little M's attack plan

题意：

给定一颗n个结点的树，每个结点都有一个权值，有q次询问，每次询问有两个数v,k，求与v的距离小于等于k的结点的权值和是多少？

思路：

如果是个有根树，且每次询问改为：求以v为根的子树中，与v的距离小于等于k的结点的权值和是多少。先考虑简化的题目。先把询问离线下来，维护一个层数的树状数组，按照dfs的顺序遍历每个点，第一次遍历到这个点的时候，如果发现这个点上有询问，那么就记录一下树状数组上层数大于该点，且层数差小于k的权值和。最后将该点的子树遍历完，要回溯的时候，再记录一下权值和，两次权值和的差值就是该次询问的答案。记这个答案为F(v,k)。

现在考虑该题目，这个题目与简化问题的差别就是该问题不要求是子树，只要求距离。那么考虑到询问的k小于100，于是我们可以把每次询问拆成k个。

即询问答案=F(v,k)+F(fa[v],k-1)-F(v,k-2)+F(fa[fa[v]],k-2)-F(fa[v],k-3)+......

这样就把询问拆分成q*k个上述简化后的问题，按照简化问题的算法就可求解。

Colorful String

建立一棵回文树，在回文树上 dfs 维护每个点的种类数。

function

首先容易发现 $f(xy) = f(x) + f(y)$

于是原式展开为 $\sum_{i=1}^{n}(n-i+1)f(i)$

每个质数角度考虑贡献，记 $g(p) = (n+1)\left\lfloor\frac{n}{p}\right\rfloor - psum(\left\lfloor\frac{n}{p}\right\rfloor)$,其中 $sum(x) = \sum_{i=1}^{x} i$, 则质数 p 的贡献 为 $g(p) + g(p^2) + g(p^3) + \cdots$ (与考虑求 $n!$ 中质数 p 的出现次数类似的思路)

对于 $\sqrt{n}$ 内的质数，暴力计算其贡献即可。对于大于 $\sqrt{n}$ 的质数，显然贡献为 $g(p)$ ，min_25 筛分块即可.

query

对于一个排列 p , 所有满足题面描述的二元组 (i,j) 的数量是 $nlogn$ 级别，可全找出来，然后对询问离线， 即是一个二维偏序查询问题，用树状数组统计即可

Random Access Iterator

树形 dp，记 $dp[u]$ 表示以 u 为根的子树，从 u 开始运行题面算法，得到正确答案的概率。

深度最深的叶子 u 的 $dp[u] = 1$ ，其他叶子 $dp[u] = 0$。转移时，考虑取不到的概率即可。

Center

两个点可以确定一个中心点，O（n^2）枚举中心点，设cnt[Xc][Yc]为中心点（Xc，Yc）被枚举到次数，
那么对于中心点（Xc，Yc），需要补上的点的个数为n-2*cnt[Xc][Yc]-（点（Xc，Yc）在原来的点集里出现的次数），
可以用map或者哈希记录cnt的值。
另外，如果把输入的坐标都乘以2，那么枚举到的中心点的坐标就都是整数了。

Dice

问题可以简化为两部分：

①骰子可以向相邻位置翻滚，问从一处到另一处最少需要几步

②下一个应当去哪个位置的多阶段决策问题

①通过bfs小数据范围打表，可以得到规律

即出现在同行或相邻行(同列或相邻列)时，需要额外消耗步数，

其余均为曼哈顿距离，特别地，满足规律如下，

```
int dis(int x,int y) //x和y分别为两点横纵坐标差的绝对值
{
    if(x==0&&y==0)return 0;
    if(x==1&&y==1)return 6;
    if(x>y)swap(x,y);
    if(x==0||x==1)return x+y+2*(y%4!=0);
    return x+y;
}
```

②求出①的规律后，状压dp，求最小距离即可

注意，由于(0,0)已认为被访问过，

当(0,0)出现在输入中时，应当被忽略

Longest subsequence

对于答案字符串来说，一定是和t串的前面部分一致，从一个字母开始比t的字符大，以后的字符就都取上就行了。

从头到尾扫描s串，维护一个26长度的数组a［26］，a[i]代表到当前位置以前取子序列，结尾为'a'+i的最长长度。

对于每个位置，比当前的字母小的都可以拼接上和t相同的部分，后面的字符全取，去构造出去答案，逐步更新即可。

注意两种特殊情况，

①字典序严格更大

②可以取的串的前缀和t相同，但是比t长(不能和t串相同)。