



# 2019牛客暑期多校训练营 第五场

[www.wodddd.zscoder](http://www.wodddd.zscoder)



牛客竞赛

AC.NOWCODER.COM



## A – The power of Fibonacci

- 这个题目是某次翻Wikipedia的时候想到的
- [https://en.wikipedia.org/wiki/Fibonomial\\_coefficient](https://en.wikipedia.org/wiki/Fibonomial_coefficient)
- $F[i] - F[i-1] - F[i-2] = 0$
- $F[i]^2 - 2 F[i-1]^2 - 2 F[i-2]^2 + F[i-3] = 0$
- $F[i]^3 - 3 F[i-1]^3 - 6 F[i-2]^3 + 3 F[i-3] + F[i-4] = 0$
- 可以推广吗？可以的，就是Fibonomial Coefficient，
- 通过简单的分析可以得到递推
- 因为m的范围是1000，需要结合多项式取模





## B - Quadratic equation

- 这个题目是某次翻Wikipedia的时候想到的
- [https://en.wikipedia.org/wiki/Quadratic\\_residue](https://en.wikipedia.org/wiki/Quadratic_residue)
- 解二次方程，核心在于求二次剩余（求平方根）
- 如果  $p \% 4 = 3$  ,  $x^2 \% p = a$
- 那么  $x = \pm \text{pow}(a, (p+1)/4, p)$
- 然后就和一般的方程一样解就可以了
- 判断是否有解用
- [https://en.wikipedia.org/wiki/Euler%27s\\_criterion](https://en.wikipedia.org/wiki/Euler%27s_criterion)



## C – Inversions of all permutations

- 这个题目是某次翻Wikipedia的时候想到的
- [https://en.wikipedia.org/wiki/Gaussian\\_binomial\\_coefficient](https://en.wikipedia.org/wiki/Gaussian_binomial_coefficient)
- Wikipedia中举了一个只有0和1的例子，可以推广到任意多重集的排列。



## D – Knapsack Cryptosystem

- 这个题目是某次翻Wikipedia的时候想到的
- [https://en.wikipedia.org/wiki/Merkle%E2%80%93Hellman\\_knapsack\\_cryptosystem](https://en.wikipedia.org/wiki/Merkle%E2%80%93Hellman_knapsack_cryptosystem)
- 本来打算出一些更数学的加密算法（比如ElGamal）但是考虑到做题体验就出了这个
- 这个题和Wikipedia的内容没什么关系，就是折半搜索





## E – All men are brothers

- 为了照顾到题目难度和类型，出了一个并查集。
- 核心就是每次合并考虑减少了多少。
- 减少的数量等于合并的两个集合大小乘以，再乘以从其他集合中选出2个不在一个集合内的方案数
- 从其他集合中选出2个不在一个集合内的方案数  
可以先计算任选2个的方案数，再减去来自同一个集合的方案数





## F – Birthday Reminders

- zscoder的动态规划
- 设 $f[i][j][k]$ 为有 $j$ 个人 $\leq i$ 时间许愿，并且恰好 $k$ 个人 $i$ 时间许愿，的方案数
- 转移的话，枚举 $i+1$ 天多少人许愿，进行转移即可。
- 需要满足2个条件
- 许愿的人不能太少（许愿时间 $\leq i+1$ 的，无论如何都会在 $\leq i+1$ 的时间许愿）
- 许愿的人不能太多（不能多过本身 $i+1$ 天的  $+k$ ）





## G – Checkers

- zscoder的状态压缩动态规划
- 注意到横纵坐标和为奇数的点，横纵坐标和为奇数的点，可以分开考虑。
- 每个黑色棋子相当于一条边。
- 需要的白色棋子个数由每个连通块内度数为奇数的点的个数决定
- 如果个数为0，需要一个，否则需要个数除以2个
- 注意每个连通块分别考虑，需要连通性状态压缩动态规划







## H – Cutting Bamboos

- zscoder的可持久化线段树
- 数组区间 $[l, r]$ 内所有数字和 $t$ 取最小值然后求和，这个问题可以用可持久化线段树解决
- 然后每次询问再做一个二分就可以了
- 实际上不需要二分，可以直接计算出答案。





## I – KM and M

- zscoder的算数题
- 咖啡鸡认为可以按位来做
- 这样变成了，等差数列除以一个数字（2的次幂）求和
- 这是一个经典题，可以用欧几里得算法来做。





## J – Symmetrical Painting

- zscoder的排序模拟题
- 最优解的斜率只会在 $L*2$ ,  $L+R$ ,  $R*2$ 变化
- 预处理出来这些位置
- 排序，求分段函数最小值就可以了。



# Thanks

