

# 牛客暑期ACM多校训练营

第 9 场-[www.nowcoder.com](http://www.nowcoder.com)



牛客网  
NOWCODER

# 写在前面.....

---

- 看起来题目不慎出的很难.....
- 但是我觉得要是杜瑜皓来做，估计就AK了.....
- 感觉都是很套路/正常的题目。
- （除了翻车的F。）

# IE. Music Game

---

## 题目大意

$n$ 个位置，第 $i$ 个位置是1的概率是 $p[i]$ 。  
问所有极长连续是1的区间长度的 $m$ 次方和的期望。

## 题目解法

这个题 $m=2$ 和 $m=3$ 都是原题，进一步推广的方法也很显然。  
如果 $m$ 太大了会有精度问题，一个非常直接的想法就是用整数来出这个题。  
直接暴力实现的话，时间复杂度大概是 $O(nm^2)$ 左右。  
然后发现原本的推广，可以做一些常数优化，这个题目就可做了。  
BZOJ 3450: Tyvj1952 Easy  
BZOJ 4318: OSU!

# IF. Typing practice

## 题目大意

建Trie图，然后求最短路。

## 题目解法

在翻题解的过程中，我发现很多人都错误的实现了AC自动机/TRIE图。

下定决心要卡掉所有错误的实现，就诞生了这么一个题。

错误原因主要是这样的，在每次匹配的过程中暴力跳Fail指针，对于一般的匹配是没有问题的。

因为退一步的前提是要进一步。（类似出栈次数一定 $\leq$ 进栈次数）

但是如果是建TRIE图的话，就不能每个点暴力跳了。（因为往回跳，不一定是因为匹配失败）

平时TRIE图随便写就对，是因为需要建TRIE图的题目，都允许 $O(L^2)$ 建图。

而如果只允许 $O(L)$ 建图的话，就必须是最优化的问题，而不能是计数问题。

并且按照这个题意出题，数据并不是特别好造，所以最后就翻车了。

赛后反思的话，认为这个题目不需要 $n$ 个串，只需要1个串写起来就已经足够麻烦了。

或者是 $n$ 个串，要求出现所有 $n$ 个串。但是这样后面的DP复杂度就变成了 $O(2^n * L)$

# IH. Prefix Sum

---

## 题目大意

修改单点，问 $k$ 次前缀和之后的一个位置的值。

## 题目解法

类似的题目早就出现过

<http://poj.org/problem?id=3468>

就是一个2次的例子，这个题目有一个2个树状数组的做法，可以推广到更多次前缀和。

一些同学使用了分块，也可以通过这个题目。

# IA. Circulant Matrix

---

## 题目大意

已知 $a$ 和 $x$ 的FWT结果是 $b$ ，输入 $a$ 和 $b$ ，求 $x$ 。

## 题目解法

多年之前做到了这个题

<http://poj.openjudge.cn/campus2014/D/>

大概就是循环矩阵，利用循环卷积解方程。

这个方法显然是可以推广到FWT的。

所以就出了这个题

做法也很简单，把FWT中的除法，换成乘法即可。

# IC. Gambling

---

## 题目大意

一个面试题

## 题目解法

假设当前胜率是 $p$ ，赢了一次之后胜率会变成 $p+q$ ，输了一次会变成 $p-q$ ，这次应该下注 $2q \cdot 2^{2n-1}$ 。  
根据这个可以得到一个 $O(n^2)$ 的DP做法。  
通过分析可以发现每次回答的实际上是一个组合数（乘以2的次幂）  
然后就可以通过预处理，每次回答组合数的方式通过这个题目了。

# IG. Longest Common Subsequence

---

## 题目大意

问四个串的LCS。

## 题目解法

LCS肯定是不可做，这个题之所以可做是因为有性质，出现次数少。  
对于每个数字，

显然最多有 $8n$ 个点，用CDQ分治，或者KD树暴力，就可以得到LCS的长度了。



# ID. The number of circuits

---

## 题目大意

问一个图的回路个数

## 题目解法

首先感谢yanQval对于这类题目的指点。

这种题目看到题，不需要思考，可以直接猜出来这是一个线性递推。

然后用高斯消元，或者Berlekamp Massey算法，找到线性递推就可以了。

做这个题目之前需要了解行列式求值和[https://en.wikipedia.org/wiki/BEST\\_theorem](https://en.wikipedia.org/wiki/BEST_theorem) 这个定理

其他有趣的题目

BZOJ 1494. [NOI2007]生成树计数

Project Euler 458

# IB. Enumeration not optimization

---

## 题目大意

NOIP2017 D2T2，计数版。

## 题目解法

之前看到Luogu上很多人用随机和错误的状态压缩过了，觉得很不开心，于是出了这个题目。因为算期望，性质非常好，做法是比较多的。

我的做法的时间复杂度是 $O(n^2 3^n)$ 。  
大概就是注意到每条边对答案的影响都是线性的，考虑这个系数，如何计算

# II. Juggernaut

---

## 题目大意

用0和1填 $n*m$ 的网格，每行每列方案

## 题目解法

这个题主要分为2（3部分）

第一部分是同构怎么做（不填最后一行，最后一列，直接算出他们）

第二部分是异或为0怎么做（经典题，见冬令营课件<http://www.wodddd.com/www.wodddd5.pdf>）

第三部分是模非质数怎么做（全程对 $nmp$ 取模，最后直接除以 $nmp$ ）

数据并不是很强，用Python也是可以通过的。

（感谢Quailty对于第三部分的讨论）

# IJ. Maze

---

## 题目大意

一个迷宫，每个墙会随机出现。  
问所有联通块大小的平方和的期望。

## 题目解法

平方和，可以转化为放置红色和蓝色两个小球，两个小球之间是连通的。  
然后插头DP，每个格子转移，全局记录是否放过红色球，是否放过蓝色球，红色和蓝色是否连通。  
然后对于插头是每个联通块，再记录是否包含红色球，是否包含蓝色球，然后逐格转移就可以了。

# Thanks