# The 2019 Aisa Nanchang First Round Online Programming Contest

## Magic Master

模拟题：单独用一个数组模拟指针，实现未翻开的牌跳过后面所有翻开的牌，直接指向下一张未翻开的牌。然后按规则模拟即可。

## Fire-Fighting Hero

图论题-单源最短路径：添加一个顶点，连接各个救火团队所在的救火点，路径长度均设为0，设该顶点为源，即变成了单源最短路径问题。使用两次Dijkstra算法可求出两个最短路径的最大值。比较时将救火团队的乘以T进行比较可避免分数操作。

## Pangu Separates Heaven and Earth

签到题

## Hello 2019

题解：建立一颗长度为$n$的线段树，每颗线段树维护矩阵乘法，复杂度$25nlogn$

## Interesting Series

展开$F$得

$$F_n = (1 - a^n)/(1 - a)$$

，对于这一类经典问题有生成函数

$$(x + a^{s_1}) * (x + a^{s_2}) * (x + a^{s_3}) * .... * (x + a^{s_n})$$

, $x^{n-k}$ 的系数就是所求，分治FFT即可，复杂度

$$n * log^2(n)$$

# The Nth Item

**题目大意**：给定一个序列 $F(n) = 3 * F(n-1) + 2 * F(n-2)(n >= 2), F(0) = 0, F(1) = 1. Q$次询问每次询问第$N$项对$998244353$取模的值。

**题解**：$Q$很大，只允许我们在线O(1)回答，于是可以推出该序列的通项公式为 $F(n) = \frac{\sqrt{17}}{17} * ((\frac{(3+\sqrt{17})}{2})^n - (\frac{(3-\sqrt{17})}{2})^n)$，再求出$17$在模$998244353$下的二次剩余，每次就可以两个快速幂求出答案，得到了一个$log(n)$的回答方法。

但是这样还是不够，我们可以预处理出式子中两个数字的0次幂，$1$次幂，到$\sqrt{10^9}$次幂，$0 * \sqrt{10^9}$次幂，$1 * \sqrt{10^9}$次幂...到$\sqrt{10^9} * \sqrt{10^9}$次幂。这样就可以回答时免去快速幂的一个$log$，做到$O(1)$在线回答。

## Megumi With String

### 题意：

给定长为$l$的串$S$, 定义一个字符串$str$的$\mathbf{value}$为$G(p)$当且仅当$str$是$S$中的某一个子串，其中$p$为子串长度，$G(x)$是一个$k$次多项式。否则我们认为它的$\mathbf{value}$为$0$。对于一个长度为$n$的串$T$,它可能由任意一个$n$个小写字符构成，而它的$\mathbf{power}$定义为它的所有子串的$\mathbf{value}$和。现在有$m$个操作，每次操作我们要在$S$末尾加入一个小写字符$c$，问每次操作后随机串$T$的期望$\mathbf{power}$为多少。

### 解题思路：

先来考虑没有修改的情况。本题中$n$很大，难以直接dp。但是经过观察可以发现，对于确定的长度$p$, $S$的每一个长度为$p$的本质不同的子串的权值是相等的。同时，这些子串在完全随机的字符串中出现的期望次数也是相等的。因此，对于每个$p \in [1, l]$，我们可以直接算出长度为$p$的一个子串的期望贡献。这个贡献被产生的次数等于$S$中长度为$p$的本质不同的子串个数，可以用后缀自动机进行求解。对$S$建出SAM后，每个节点会对长度在其$len$与其父节点的

$len$之间的一段产生$+1$的贡献。由于贡献一定是连续的一段，可以对每个长度的期望贡献求前缀和后维护。

由于后缀自动机是在线算法，所以支持修改很容易，只需每次计算被改变的节点的贡献即可，单次修改是常数复杂度。在字符集大小为常数的情况下，总时间复杂度为$O((l+m)k)$

---

# Yukino with Subinterval

题意：给一个数组$a$，支持单点修改，询问区间中值域在一定范围内的最长连续段的个数。

我们直接暴力去数连续段显然不太方便，因此我们考虑将整个连续段压缩成一个点。

我们考虑将每一个连续端都只统计最左边的数，那么我们可以定义一个新的数组$b$，其中，$b_i = a_i$当且仅当$a_i \neq a_{i-1}$，否则$b_i = 0$.

之后这个问题就被转化成求数组$b$在区间$[l+1, r]$，值域在$[x, y]$的个数。

而这个问题是一个经典的主席树的问题，对于每个查询，我们只需要遍历一下我们建出来的主席树即可。鉴于本题需要支持单点修改，因此可以考虑用树状数组维护一下主席树。但是这种树套树的方法常数比较大，而笔者大概将大常数生成器们的树套树都卡掉了，而如果想要用该法通过该题则必须要有比较厉害的卡常技巧。

而有经验的选手肯定可以发现，其实这个问题实质上就是一个三维偏序的问题，因此我们可以离线通过cdq分治解决。虽然时间复杂度跟树套树相同，但是在常数上cdq分治优势巨大。

最后注意，因为我们判断的是数组$b$在区间$[l+1, r]$的状态，因此最后我们还必须要特判一下最左端的数$b_l$的大小。树套数或者cdq分治的整体的时间复杂度均为：$\mathcal{O}(nlogn^2)$

---

# Enju With Euler function

题意：给你由100个数组成的数列，问你这些数是否是欧拉函数中的其中连续的一段。

暴力的做法是用欧拉筛，筛出范围内的欧拉函数，而在每一位进行一次check。判断数列是否为当前段。这样时间复杂度是$\mathcal{O}(T * 1.6 * 10^8 * 100)$.明显不行。

首先，欧拉函数对于第$i$位的三元组$\{\varphi(i), \varphi(i+1), \varphi(i+2)\}$是唯一的。

这样判断的时候通过已给数列前三个在欧拉函数中确定位置，如有，对于每个数字check一下即可，否则亦无解。

不过，这样仍然要打出完整的欧拉筛，是还未足够的。

接下来需要运用一个性质，

$$f(n)=\begin{cases}\varphi(n*p)=\varphi(n)*p, & \text{if } n \text{ mod p=0} \\ \\ \varphi(n*p)=\varphi(n)*(p-1), & \text{else}\end{cases}$$

将$k$个2分别代入原式并推导可发现如下式子：

$$f(n)=\begin{cases}\varphi(n*2^k)=\varphi(n)*2^k, & \text{if } n \text{ is even} \\ \\ \varphi(n*2^k)=\varphi(n)*2^{k-1}, & \text{if } n \text{ is odd}\end{cases}$$

假设有一个32的倍数$j$,对于三元组$\{j,j+32,j+64\}$除以32一定为偶奇偶或奇偶奇，而除以16一定都为偶数，别对两种情况通过上述定理进行预处理，则可以得到可能的$\{a[\frac{i}{32}],a[\frac{i}{32}+1],a[\frac{i}{32}+2]\}$。

正确题解：那么我们可以进行dfs，对于已给数列$a$，枚举$i\in[1,32]$，取出$\{a[i],a[i+32],a[i+64]\}$，其中必定存在有一组三个位数均为32的倍数。预处理出所有可能的$\{a[\frac{i}{32}],a[\frac{i}{32}+1],a[\frac{i}{32}+2]\}$。那么欧拉筛就只需筛出$\frac{1.6*10^8}{32}=5*10^6$的数量。然后查找出现的三元组，对于每种可能的三元组的位置$\times 32$回归原位做一遍check，则可以得到答案，而且理论可能的三元组需要check的很少，即使check非正确答案也很快可退出。

假定加入的所有三元组都进行了\text{check}，并且每组样例可加入的三元组数量32个。

那么，时间复杂度$\mathcal{O}(T*(5*10^6+100\sqrt{1.5*10^8}+32log(32)\ )\ )$。

由于数据离线，可以使用压位预处理一下，压位保存三元组所在样例以及位置，复杂度可以可降到$\mathcal{O}(5*10^6\lceil\frac{T}{64}\rceil+T*(100\sqrt{1.5*10^8}+32klog(32*64)\ )\ )$ ($k$是常数)(该式子欧拉筛外复杂度仍然不会跑满）。