预期难度

签到：EJ

$easy$: FAB

$middle-esay$: GK

$middle-hard$: LHD

$hard$: IC

E

找到第一个和a_1互质的数，后面依次填"232323"或者"323232"

J

假设$a_1 \leq a_n$，那么如果第一步选择了a_n,a_1就再也不能选了。

以此类推，总的状态数量是$O(n)$的。

F

在第t秒时整棵树上的节点数为$(k+1)^t$,在给定的数据范围下树高不会超过25。

因此我们可以直接按照求LCA的方法，每次求编号较大的点的父亲节点，往上跳即可。

查询父亲可以用预处理每层的节点数来解决。

父亲节点坐标$fa=(x-num-1)/k+1$,其中num为上一秒树上的节点数。

具体查询num时$O(log)$的$lower_bound$或者$O(1)$的双指针都是可以的。

A

Solutions

根据费马小定理，$10^{p-1} \equiv 1 \pmod{p}$在$p$为任意合法输入时成立。

Solution-1（数学）

令$k = p - 1$，则$10^{tk} \equiv 1 \pmod{p}$在$t$取任意非负整数时成立。

对题给的询问进行分析可知，
$Q(l, r, k) = \sum a_{r-tk} = \sum \frac{a_{r-tk} \cdot 10^{n-r+tk}}{10^{n-r+tk}} \equiv 10^{r-n} \sum a_{r-tk} \cdot 10^{n-r+tk} \pmod{p}$。

即$10^{n-r} Q(l, r, k) \equiv \sum a_{r-tk} \cdot 10^{n-r+tk} \pmod{p}$。

通过多个上述同余方程等式两侧分别相加可得
$\sum_{i=1}^{k} 10^{n-r_i} Q(l_i, r_i, k) \equiv \sum_{i=1}^{n} a_i \cdot 10^{n-i} = x \pmod{p}$。

Solution-2（构造）

由定理变换可得$10^{p-1} - 1 \equiv 0 \pmod{p}$，也即$10^{p-1} - 1$被$p$整除。我们想要求的答案$x \bmod p$，可由$x \bmod 10^{p-1} - 1 \bmod p$连续取余得到。

显然，$10^{p-1} - 1 = 99999 \cdots 9$。

观察$x \bmod 9$的计算方法，我们会发现可以由所有数位相加后模9得到。（比如$1234 \bmod 9 = (1 + 2 + 3 + 4) \bmod 9 = 1$）

观察$x \bmod 99$的计算方法，我们会发现可以由所有数位两两分组后，按组相加再模99得到。（两两分组的含义是：十位和个位为一组，千位和百位为一组，组成若干个两位数，比如$12345 \bmod 99 = (1 + 23 + 45) \bmod 99 = 69$）

以此类推可以得到更多类似数的余数计算方法，实际上，其原理其实就是同余关系。

于是我们只需将数间隔若干位为一组，相加后取模即可，可以通过题给的询问快速得到。

记$s = \min(100, n)$，两种做法的时间复杂度均为$O(s^2 + sq)$。

值得一提的是，当n较小时，可能会出现$n < p - 1$的情况，但是我们不难发现，当询问轮数超过n时，前n次询问恰好按十进制位表示了数x，而剩余的询问结果一定是0，在代码上稍作处理即可。

B

观察可知删除和修改操作本质是相同的

于是相当于删除$2k$个数

$dp[i][j]$表示前i个保留了j个且保留了最后一个的最大价值

枚举上一次保留的位置可以$O(n)$转移

总时间复杂度:$O(n^3)$

G Good Permutation

因为区间只有相离和不包含的关系。所以所有关系可以用一个树形结构表示出来。

当前节点在u,儿子节点为$v_1, v_2 \dots v_n$,那么转移相当于是将所有儿子和其他的散点放在一起重排。

$dp[u] = dp[v_1] \times dp[v_2] \times \dots \times dp[v_n] \times (u_{len} - v_{1\,len} - \dots - v_{n\,len} + n)!$

时间复杂度$O(nlogn)$主要在建树的复杂度上。

K

对每一个$1*1$的格子分别讨论，然后圆弧肯定是若干个$\frac{\pi}{2}$或者$\frac{\pi}{3}$的和。

题目内测的时候没人发现有歧义，但是赛时确实给部分队伍带来了影响，非常抱歉！

L:

定义$f[i]$代表只考虑以i往后的字符串中$icpc$的个数

对于询问$[l,r]$,考虑$f[l]-f[r+1]$的含义,这个计算的是$icpc$都在$[l,r]$内以及i在$[l,r]$内,cpc在$[r+1,n]$,ic在$[l,r]$内,pc在$[r+1,n]$内,icp在$[l,r]$,c在$[r+1,n]$内

于是这个通过容斥变成了子问题,可以在$O(n+q)$时间内完成

H

对于树上的一个点对，若两点之间没有其他关键点，这两个点之间的距离会被统计进答案。

对于一对距离为d的点，那么等于$d+1$个位置被占用了，还有$n-d-1$个位置可以选。

那么总的贡献为
$d\times(2\times C(n-d-1,0)+3\times C(n-d-1,1)+..+(n-d+1)\times C(n-d-1,n-d-1))$

由于$\sum i\times C(n,i)=\sum n\times C(n-1,i)=n\times 2^{n-1}$

所以答案为$d\times(n-d+3)\times 2^{n-d-2}$

对这个式子，可以用换根dp解决，这个式子类似于$\sum x\times y\times 2^y$,我们需要维护x整体减一，y整体加一和x整体加一，y整体减一的操作，维护转移可以新开一些辅助变量$\sum x\times 2^y$ $\sum y\times 2^y$ $\sum 2^y$来转移。时间复杂度$O(n)$。

D

假如翻转的区间是$[h,t]$

将$A=a[1,h-1],B=a[h,t],C=a[t+1,n],D=a[t,t-1,\ldots,h+1,h]$($D$是翻转后的$B$)

于是翻转之后的序列变为$A+D+C$

$(A+B+C)^2=A^2+B^2+C^2+2*A*B+2*A*C+2*B*C$

会发现有改变的项只有$B^2,2*A*B,2*B*C$

考虑对于原序列的这三项怎么求

对于B^2,枚举两个元素i,j,可能包含它的有$i*(n-j+1)$种情况

对于$A*B$,枚举两个元素i,j,可能包含它的有$(j-i)*(n-j+1)$种情况

对于$B*C$同理

对于D^2,枚举两个元素i,j,那么它最终贡献到的位置是$2h+2t-i-j$,其中$(1\le h\le i,j\le t\le n)$

会发现$V=2h+2t-i-j$对应的情况数是等差数列加一段平的再加一段等差数列

于是可以利用二阶差分维护

对于其他情况同理

总时间复杂度$O(n^2)$

I

实际上只有在快要升级的时候，单次使用的道具数量才会超过1。

那么实际上只有不超过M个需要做dp的状态。

然后这个dp可以用李超线段树优化。

需要注意的是$M = 1$的时候答案是0，有些写法可能需要特判。

C

考虑公差的范围

我们假设首项是b，公差是d

最少的操作次数是$(b*n+n*(n-1)/2*d-\sum a_i)/2$

同时我们可以计算出答案的上界是$(a\text{中最大元素}*n-\sum a_i)/2+2*n$

根据前两项可以得出大概范围$n*d\leq 2*10^6$

考虑确定d要求所有位置都等差怎么做

首先将表达式写成绝对值的和

分奇数偶数讨论只后 表达式是形如$|x-a_i|$或者$|x-a_i|/2$

这个是比较经典的带权最小值

最小值一定取在$(n+2)/3$上

由于要考虑到奇偶，所以答案需要计算$x=a[(n+2)/3]$ 和 $x=a[(n+2)/3]+1$

回到这个题实际上要求拆成两个等差数列

所以我们需要维护一个前缀和后缀

这个可以利用线段树来进行维护