水题选讲

2019.7.

CEOI2016 Kangaroo

求有多少个排列 $\{p_1, p_2, \ldots, p_n\}$ 满足 $p_1 = s$, $p_n = t$ 且对于 $1 < i < n$ 满足 $p_{i-1} < p_i$ 且 $p_{i+1} < p_i$ 或 $p_{i-1} > p_i$ 且 $p_{i+1} > p_i$。输出答案对 $10^9 + 7$ 取模。
数据范围：$1 \leq s, t \leq n \leq 2000$, $s \neq t$。

CEOI2016 Kangaroo

令 $f(i,\ j,\ x)$ 表示前 i 个点已经填好了，构成了 j 条不包含 s，t 的链，s 和 t 所在的链状态为 x。

CEOI2016 Kangaroo

令 $f(i,\ j,\ x)$ 表示前 i 个点已经填好了，构成了 j 条不包含 $s,\ t$ 的链，s 和 t 所在的链状态为 x。
转移时枚举第 $i\ +\ 1$ 个点连上哪些链即可。

画仙人掌

给 n 个点仙人掌，给每个点分配一个坐标 $(x,\ y)$，使得边不相交。$x,\ y$ 为 1 到 n 的整数。
$n \leq 10^5$

画仙人掌

考虑树怎么做，每个点 x 坐标为深度，y 坐标为 dfs 序。

画仙人掌

考虑树怎么做，每个点 x 坐标为深度，y 坐标为 dfs 序。
首先建出 *dfs* 树，对每个环中间新建一个点，向环上每一个点连边。新建的点成为虚点，虚点坐标可以是实数，然后不管环上边，对得到的树进行分配坐标。

AGC030F

有一个长度为 $2N$ 的数列，A_1，A_2，...，A_{2N}，
$A_i \in \{-1, 1, 2, ..., 2N\}$。非 -1 的数在 A 中只会出现最多一次。
现在要将所有 -1 替换成 1 到 $2N$ 中的一个数，使得 A 为排列。
定义 B_1，B_2，...，B_N，其中 $B_i = min(A_{2i-1}, A_{2i})$。求不同的 B 的个数，对 $10^9 + 7$ 取模。
数据范围：$n \leq 300$。

AGC030F

考虑一对 (A_{2i-1}, A_{2i})，如果它们都有值了就直接扔掉，剩下的只有两种：$(-1,-1)$ 和 $(-1,x)$ 。对于 $(-1,-1)$ ，我们忽略它们的位置关系，最后答案乘上其个数的阶乘。

考虑从大到小填数，记 $f(i,j,k)$ 表示当前考虑到 i ，有 j 个 -1 没有匹配，有 k 个 x 没有匹配，考虑转移：

如果 i 是某个 x ：

1. 不匹配，转移到 $f(i-1,j,k+1)$
2. 匹配一个 -1 ，转移到 $f(i-1,j-1,k)$

如果 i 不是某个 x ：

1. 不匹配，转移到 $f(i-1,j+1,k)$
2. 匹配一个 -1 ，因为是无序的所以直接转移到 $f(i-1,j-1,k)$
3. 匹配一个 x ，因为 x 的位置有关所以有 k 种方案转移到 $f(i-1,j,k-1)$

传送门

有 n 个结点，m 个传送门，一个传送门双向连接两个结点，每当你从这两个结点之一跳进这个传送门时，你会花费一定的时间传送到另一个点处。现在你想用尽量短的时间从 S 点走到 T 点。这是一个简单的最短路问题。

但实际情况是这些传送门里有一个是损坏的。但你不知道损坏的是哪个，只有当你站在一个传送门的连接的两个结点之一时才能知道这个传送门是否是坏的。问你最少需要多少时间使得你能保证在这些时间内能从 S 点走到 T 点。

$n, m \leq 2 * 10^5$

传送门

首先考虑怎么求删掉一条边后相邻两个点到 T 的距离。建出最短路树，如果删掉的不是连向父亲的边，则最短路不变，否则肯定是走到子树（包括自己）中的一个点，然后跳到子树外面。对每个点维护一下连接子树内和子树外的所有边以及价值，可以弄个堆维护一下。求 u 时，将它所有孩子的边集的权值更新一下，再合并起来，从小往大找到第一条可以跳出子树的边，把在这条边之前的边都删掉就行了。可以使用可并堆或启发式合并。

传送门

对每个点 u，记 $d(u)$ 表示 u 到 T 的最短路，$e(u)$ 表示删掉它和最短路上父亲的边后的最短路。令 $dp(u)$ 表示 $S = u$ 时的答案。每次找到 dp 值最小的点来更新其它的点的 dp 值即可。用 u 更新 v 时的转移为 $dp(v) = \min(\max(dp(u) + w(u, v), u == parent\ v?e(v) : d(v)))$。

loj6130

给 n，k，A_1，A_2，...，A_n，执行如下程序：

for i **from** 1 **to** k
  for j **from** 1 **to** $n-1$
    if $A_j > A_{j+1}$
      swap(A_j, A_{j+1})

输出之后的 A 序列。
数据范围：$1 \leq k < n \leq 200000$。

loj6130

维护排序比较难，尝试确定每个数的最终位置。

loj6130

维护排序比较难，尝试确定每个数的最终位置。
对于一个数 A_i 如果前面有 $\geq k$ 个数比它大，那么它最终序列的位置一定是在 $i - k$。

loj6130

维护排序比较难，尝试确定每个数的最终位置。
对于一个数 A_i 如果前面有 $\geq k$ 个数比它大，那么它最终序列的位置一定是在 $i - k$。
否则考虑对于 $j < i$, $A_j > A_i$，最终位置 A_i 都会在 A_j 前面。
直接排序即可。

ARC073F

你有两个整数a和b。
现在n个操作，依次执行，每次给你 x_i，你选择两个整数中的一个y变成 x_i，代价为 $|x_i - y|$。
求做完所有操作的最小代价。
数据范围：$n,\ x_i,\ a,\ b \leq 200000$。

ARC073F

设 $f[i]$ 表示做完前 i 个操作，其中一个整数变成 x_i，另一个变成 x_{i-1} 的最小代价。
第一次操作枚举是哪个变成 x_1 做两次dp，以 a 为例，那么 $f[1] = |a - x_1|$，然后 $x_0 = b$，即为初值。
转移是 $f[i] = min(f[j] + sum(j+1, i-1) + |x_i - x_j - 1|)$。
按绝对值讨论拆开，然后维护两颗线段树即可。

Arithmetic of Bomb II

现有一个长度为 N 包含 Bomb 运算表达式。Bomb 也就是 $\#$ 号，会展开一些普通表达式。比如 $(1-2+3)\#(3)$ 表示 $1-2+3$ 出现了 3 次，将会被展开为 $1-2+31-2+31-2+3$ 。先将Bomb 表达式中所有的 $\#$ 号展开，使其成为只包含加、减、乘运算符的表达式，再求解这个表达式的结果，对 10^9+7 取模。

数据范围 $N \leq 2 * 10^6$

Arithmetic of Bomb II

考虑对展开式从左向右扫，将当前答案表示成 $a+b*c$ 的形式。
当下一个字符为数字 d 时，(a,b,c) 变成 $(a,b,10*c+d)$。
当下一个字符为 $*$ 时，(a,b,c) 变成 $(a,b*c,0)$。注意此时将 c 改为 0 为了方便之后接入数字。
当下一个字符为 $+$ 时，(a,b,c) 变成 $(a+b*c,1,0)$。
当下一个字符为 $-$ 时，(a,b,c) 变成 $(a+b*c,-1,0)$。
使用矩阵快速幂即可。

Snuke the Phantom Thief

二维平面上有 n 个宝石，位置为 $(x,\ y)$ 价值为 w。有 m 条限制，每个限制形如在给定的与一条坐标轴平行的直线下方最多只能取 x 个宝石。问最多能取的宝石的价值是多少？

$n \leq 80,\ m \leq 320$。

Snuke the Phantom Thief

首先枚举选了多少个物品 k，则限制可以转化为 $y = a, x = b$ 的直线下方左方选取数量为区间 $[l, r]$。
考虑只有一维怎么建，按坐标大小从小到大连边，容量为限制 $[l, r]$。当一个物品在限制直线的上方右方时，将那个限制连一条到这个物品容量为 1，费用为价值的边即可。
若存在可行流，则有解，更新答案。

简单数据结构题

给一棵n个点的树，点权开始为0，有q次操作，每次操作是选择一个点，把树上到它距离为1的点点权+1，并输出操作后这些点点权的异或和。
数据范围：$n, q \leq 5 * 10^5$

简单数据结构题

注意到 $x\ xor\ (x+1) = 2zerobit(x)\ -\ 1$，zerobit(x)表示x的最低0位（即lowbit(x)）。
距离为1的点可以分成孩子和父亲，那么相当于就是支持单点+1，一个点的孩子+1，查询某个点孩子的异或和或者单点值。

假设我们对于每个点的儿子维护一个集合，那么相当于我们要支持维护一个集合，支持插入、删除、全局+1和查询全局异或和。

考虑 $x\ xor\ (x+1) = 2zerobit(x)\ -\ 1$，那么我们对每个集合存下当前的异或和，我们全局+1的时候只要查询集合里zerobit为每个位的分别有几个就行了。

简单数据结构题

考虑把每个集合中的数用一棵trie来维护，trie第一层为最低位，越下面位数越高。
考虑全局+1的过程，如果最低位为0，那么变成了1，如果最低位为1，那么变成了0，并且进位，这实际上就是交换左右子树，然后递归左子树。查询zerobit为每个位的个数的话只要在trie上往0边走就好了，插入和删除也都可以直接在trie上进行。
复杂度O(n log(n))（假设n、q同阶）。

染色游戏

给一个长度为 n 的序列，选出一个上升子序列使得 $\sum a - \sum \frac{b(b+1)}{2}$ 最大，其中 a 是被选择的所有数，b 是所有空白连续段的长度。
数据范围：$n \leq 10^6$。

染色游戏

令 f_i 表示最后一个选择的数为 i 时的最大价值，容易发现 $f_i = max_{a_j \le a_i, j<i}\ f_j + a_i - \frac{(i-j)(i-j-1)}{2}$。
没有 $a_j \le a_i$ 时就是一个简单的斜率优化问题，于是我们按照权值分治后就能套用斜率优化了。归并排序预处理每一层排序的结果，时间复杂度 $O(nlogn)$。

Func

已知 $F(1)=1$ $F(2i)=F(i)$ $F(2i+1)=F(i)+F(i+1)$ 。输入 n ，输出所有满足 $F(m)=n$ 的正奇数 m 输出时对 998244353 取模。
数据范围：$n \leq 10^6$

Func

首先根据归纳法可以得出 F(n) 和 F(n+1) 是互质的且对于 n 为偶数有 $F(n+1) > F(n)$ 。
题目要求某个奇数 m 使 $F(m) = n$ ，所以 $F(m-1)$ 是一个 1 到 $n-1$ 中与 n 互质的数，枚举之。
现在我们知道了这个函数在相邻两个整数的取值
$F(i-1) = x, F(i) = y$ 可以通过倒推的方式得出 i 的值。首先判断 x, y 的大小关系得出 i 的奇偶性，然后可以根据递推式递归到 $\frac{i}{2}$ 或 $\frac{i-1}{2}$ 的情况。这样一层层递归的话每次计算最坏是 O(n) 的。算上之前的枚举，总时间复杂度 $O(n^2)$
如果每次同时递归多层，相当于把“辗转相减”改成“辗转相除”这样每次计算复杂度 $O(logn)$，总时间复杂度 $O(nlogn)$。

AGC001E

给 $A_1, \ldots, A_n, B_1, \ldots, B_n$，求 $\sum_{i,j} \binom{A_i+B_i+A_j+B_j}{A_i+A_j}$。

数据范围：$n \leq 200000$, $1 \leq A_i, B_i \leq 2000$

AGC001E

$\binom{x+y}{x}$ 可以想象成从 $(0,0)$ 出发只能向右或向上到 (x,y) 的方案数。
那么 $\binom{A_i+B_i+A_j+B_j}{A_i+A_j}$ 可以理解成从 $(-A_i,-B_i)$ 出发只能向右或向上到 (A_j,B_j) 的方案数。
这些方案数求和相当于从一个点集任意一点出发，只能向上向右走到另一个点集任意一点的方案数总和，这可以用 DP 解决。

cheese

有 n 对pair $(v_i,\ m_i)$，你能最多分割两对pair，即将 $(v,\ m)$ 分成 $(pv,\ pm)$ 和 $((1-p)v,\ (1-p)m)$。
分割后你希望将所有pair分成两组使得 v 之和相同且 m 之和相同。
数据范围：$n \leq 10^5$。

cheese

把每个pair看成矩形 $\frac{m_i}{v_i} \times v_i$，总宽度为 $V = \sum v_i$，总面积为 $M = \sum m_i$。
按照 v_i 排序。
问题变成选出一段宽为 $\frac{V}{2}$ 的使得面积为 $\frac{M}{2}$。二分答案即可。

网友串

网友喜欢混乱的、毫无规律可言的字符串：网友喜欢看到长度为偶数的且前半段不等于后半段的字符串。举例来说，串 ‘0011‘ 就符合网友的审美，但是 ‘001‘, ‘0000‘ 都不符合网友的审美。
最近，在老板的要求下，网友要组织举办一个小活动，他需要给这个活动写一段标语。根据老板的要求，标语应当由 n 个单词组成，第 i 个单词 s_i 是一个长度为 a_i 的 ‘01‘ 字符串。
因为网友闲着没事，他定义这个标语中第 i 个单词是混沌的当且仅当存在 $1 \leq j < i$ 使得字符串 $s_j s_i$ 是符合网友审美的。例如在标语 ‘01‘, ‘10‘, ‘111‘ 中，只有第二个串是混沌的。
显然可能的标语一共有 $2^{\sum_{i=1}^{n} a_i}$ 种。网友现在想要对每一个 $k \in [0, n]$，计算恰好有 k 个单词是混沌的标语个数。
数据范围：$1 \leq n \leq 50, 1 \leq a_i \leq 7$。

网友串

首先，奇数和偶数可以分开处理，最后再将答案卷积得到最终答案。
预处理每一对网友数是否能够构成混沌串。
我们用三元组 (i,j,s) 来描述一个状态，表示处理了前 i 个数，出现了 j 个混沌串，并且能与集合 s 中的串组成混沌串的字符串出现过至少 1 次。
一个直观的想法是动态规划，记 $dp_{i,j,s}$ 表示到达 (i,j,s) 的方案数，利用预处理的结果，我们可以轻松地完成转移，但可能的 s 似乎很多。但实际上，可能出现的 s 少之又少，因此直接按照此方式动态规划即可。
时间复杂度 $O(N^2 * Cnt * 2^{a_i})$，其中 Cnt 为合法的 s 的个数，考虑 $a_i = 1, 3, 5, 7$ 时，$Cnt = 238$，考虑 $a_i = 2, 4, 6$ 时，Cnt=106。

tree

给出一颗 n 个点带边权有根树，有如下操作，共 q 次：
1 0 u v：询问u到v路径上边的权值的方差
1 1 u v a b：将u到v路径上边的权值从x变成ax+b
2 0 u: 询问u子树中边的权值的方差
2 1 u a b：将u子树中边的权值从x变成ax+b
3 0 u v: 询问集合S(u,v)中边的权值的方差
3 1 u v a b: 将集合S(u,v)中边的权值从x变成ax+b
其中S(u,v)定义为：如果一条边的某个顶点在u到v的路径上，那么这条边属于S(u,v)。
所有运算在模10^9+7下进行。
数据范围：$n, q \leq 10^5$。

tree

$\frac{\sum(x_i-\bar{x})^2}{n} = \frac{\sum x_i^2}{n} - (\frac{\sum x_i}{n})^2$ 所以只用记录元素的和以及元素的平方和。将修改表述成 $ax+b$ 的形式，可以合并：
$(ax+b)*c+d = (ac)x+(bc+d)$。如果问题在序列上，可以用线段树解决。
没有操作3直接按照树链剖分即可。
考虑把重边和轻边分开维护，先分配所有重边的编号，对于轻边我们把一条重链上作为上点的轻边一起分配编号，这样就行了。

GP of Siberia

字符串是好串，当且仅当它本质不同的子序列（可不连续）数量是奇数。
给定 n 个串 $s1, s2, \ldots, sn$, 求 $n!$ 种拼接方法中好串的数量。
数据范围：$n \leq 20$。

GP of Siberia

考虑如何计算本质不同子序列个数，用 $dp('a',' b', \ldots,' z')$ 表示以每个字符结尾的子序列个数，令 tot 表示总数。可以发现在对 2 取模意义下就是交换 tot 和 dp 中的一个值，故总共只会有一个奇数，所以 dp 的值只有 27 种，对其用 dp of dp即可。

Balanced Sequence

有 n 个括号序列，你可以把它们排列之后首尾相连拼在一起，然后删去一些字符使得剩下的是合法括号序列。
$\sum |S| \leq 5 * 10^6$

Balanced Sequence

首先可以对每个串把匹配的括号删去，剩下来为 a 个) 和 b 个 (。
令 sum 表示和，m 为前缀和最小值，则能保留的最长长度为 $n - sum + 2m$。由于 n 固定，则要最大化 m。
$a < b$ 肯定排在 $a > b$ 前面。
同为 $a < b$ 则 a 小的排前面。
同为 $a > b$ 则 b 大的排前面。

打怪兽

打一个怪兽要先掉 a_i 滴血再补 b_i 滴血。树上每个点有个怪兽，打完一个点父亲才能打这个点，问最少要多少血使得血始终非负。

$n \leq 10^6$

打怪兽

如果可以任意排列则方案与上一题一样。
每次先选择顺序最小的，如果其父亲是根，则打这个怪兽，否则将其和父亲合并，意思是打完父亲就打它。(a_i, b_i)，(a_j, b_j) 合并完为
$(max(a_i, a_i - b_i + a_j), -a_i + b_i - a_j + b_j + max(a_i, a_i - b_i + a_j))$。

异或

给定一个长度为 n 的序列 x (0 下标)，每一次操作为将序列 x 变为序列 y ，满足 $y_i = \oplus_{j=0}^{k-1} x_{i+j} mod n$，其中 $\oplus$ 表示异或。现在给一个序列 x，参数 k 和数字 T，求 T 次操作后的序列是什么。

数据范围：$n \leq 5 * 10^5$，$T \leq 10^9$。

异或

每次操作相当于在模 2 循环卷积意义下乘上 $f = \sum_{i=0}^{k-1} x^i$。
模 2 意义下的多项式其实有很多性质，比如平方是很好求的，只要将所有 x^i 变成 x^{2i} 即可。所以 f^{2^t} 是非常好求的。
考虑乘上 f^{2^t}，实际上就是把 x 变成 y，其中 $y_i = \oplus_{j=0}^{k-1} x_{i+j*2^t} mod n$。可以直接暴力模拟。

SRM733Hard

求一个n个点竞赛图的哈密尔顿回路。
数据范围：$n \leq 50$。

SRM733Hard

有结论：n 个点竞赛图有哈密尔顿回路当且仅当 SCC 个数是 1。
考虑每次增加一个点 i，维护每个 SCC 哈密尔顿回路。
令 lst 为当前图 SCC 缩点后的链上最后一个有向连 i 边的点的 SCC。
令 fst 为当前图 SCC 缩点后的链上第一个有从 i 连边的点的 SCC。
若 $lst < fst$ 则暴力插入。
若 $lst = fst$ 则更新 lst。
若 $lst > fst$ 则合并 fst 到 lst 中所有 SCC。
总复杂度 $O(n^2)$。

CF868 F

给序列 $a[1 \ldots n]$ 和 k，分成 k 个连续子段，使每一段中重复数对个数的和最小。
数据范围：$2 \leq n \leq 10^5$, $2 \leq k \leq \min(n, 20)$。

CF868 F

令 $f_x(i)$ 表示前 x 段覆盖了 $a[1 \dots i]$ 最小代价。转移具有单调性。

CF868 F

令 $f_x(i)$ 表示前 x 段覆盖了 $a[1 \dots i]$ 最小代价。转移具有单调性。
用分治进行优化。
计算 $f_x(i),\ (l_1 \le i \le r_1)$ 对 $f_{x+1}(mid_2)$ 的贡献可以类似莫队算法，记录指针 $s,\ t$ 和当前 $a[s \dots t]$ 中每个数出现次数以及重复数对个数 cur，可以发现指针移动次数与递归函数中下标移动是同阶的，故复杂度为 $O(knlogn)$。

找子矩阵

给定 $n * m$ 个数字，可以修改不超过 A 个数字为 0。问选 B 个不相交的列数为 m 的矩形所能得到的和的最大值是多少。
$1 \leq n \leq 100,\ 1 \leq m \leq 3000,\ 0 \leq A \leq 10000,\ 1 \leq B \leq 3,\ |w_{i,j}| \leq 10^9$

找子矩阵

令 $F(i,\ j,\ k,\ 0/1)$ 表示考虑过前 i 行，改了 j 个数字，k 个矩形开始选了，当前行选不选的最大值。
有转移 $f(i,\ j,\ k,\ 0)\ =\ max(f(i\ -\ 1,\ j,\ k,\ 0),\ f(i\ -\ 1,\ j,\ k,\ 1))$, $f(i,\ j,\ k,\ 1)\ =\ max\{f(i\ -\ 1,\ j\ -\ a,\ k\ -\ 1,\ 0)\ +\ g(i,\ a),\ f(i\ -\ 1,\ j\ -\ a,\ k,\ 1)\ +\ g(i,\ a)\}$。$g(i,\ a)$ 为第 i 行改 a 个数字的和。考虑 $f(i,\ j,\ k,\ 1)$ 的转移，由于 $g(i,\ a\ +\ 1)\ -\ g(i,\ a)\ \geq\ g(i,\ a\ +\ 2)\ -\ g(i,\ a\ +\ 1)$，所以转移具有单调性，直接分治转移即可。

CodeFestival16FinalH

A 和 B 玩游戏。有 n 个数，从左到右 1 到 n，每个数有权值 a_i。A 在第 1 个数上，B 在第 2 个数上，每次左边的那个人选择右边一个没有被人占过的位置过去，若无这样的位置游戏结束。两人的得分为所占的所有格子的权值和。

q 次询问，每次询问给 x，问 $a_n = x$ 时 A 得分 $-$ B 得分最大值是多少。（两人均最优策略）

数据范围：$1 \leq n \leq 200000$, $0 \leq a_i \leq 10^6$, $\sum_{i=1}^{n} a_i \leq 10^6$, $q \leq 200000$, $0 \leq x \leq 10^9$。

CodeFestival16FinalH

令 dp_i 表示 $[i...n]$ 游戏先手 $-$ 后手得分的最大值，可以发现 $dp_i = |a_i - dp_{i+1}|$。
令 $f_{i,x}$ 表示当 $dp_{i+1} = x$ 时，dp_1 的值。则 $f_{i,x} = f_{i-1,|a_i-x|}$，即将 f_{i-1} 向右平移 a_i 位，再以 a_i 为轴翻转。用 *deque* 维护即可。
每次询问找到后缀和 $> x$ 的最大的 i，答案就是 $f_{i,x-\sum_{j>i} a_j}$。
时间复杂度 $O(n + q + \sum_{i=1}^{n-1} a_i)$。

无限之环

游戏在一个 $n \times m$ 的网格状棋盘上进行，其中有些小方格中会有水管，水管可能在方格某些方向的边界的中点有接口，所有水管的粗细都相同，所以如果两个相邻方格的公共边界的中点都有接头，那么可以看作这两个接头互相连接。水管有 15 种形状。
游戏开始时，棋盘中水管可能存在漏水的地方。
形式化地:如果存在某个接头，没有和其它接头相连接，那么它就是一个漏水的地方。
玩家可以进行一种操作：选定一个含有非直线型水管的方格，将其中的水管绕方格中心顺时针或逆时针旋转 90 度。
现给出一个初始局面，请问最少进行多少次操作可以使棋盘上不存在漏水的地方。
$nm \leq 2000$

无限之环

对格子黑白染色，对每个格子的四条边建四个点。一种颜色与 S 相连，另一种与 T 相连。
一个插口对初始位置连，再将初始位置向另外三个位置连。
两个插口的先对上左连，上对下连，左对右连。
三个插口对三个初始位置连，每个点再对剩下来一条边连。
四个插口直接连。
跑最小费用最大流即可。

uoj445

交互题。有长度为 n 的排列，a_0，a_1，…，a_{n-1}。求一个排列 p_0，p_1，…，p_{n-1} 满足 $a_{p_i} < a_{p_{i+1}}$。有函数 $super\_sort(int * p, int\ n, int * q)$，传入的 n 要满足 $n \leq k$，k 给定。q 为对 p 的一个重排列，满足 $a_{q_i} < a_{q_{i+1}}$。调用 $super\_sort()$ 的次数不能超过限制 L。

数据范围：$n = 2^{19}$，$k = 16$，$L = 562000$。

uoj445

用线段树来辅助排序，$node[l...r]$ 表示区间 $[l,\ r]$ 中 a 最小的下标。每次取出 $node[0...n-1]$ 然后将这个点删去，将这个点到根的路径update一下。建树的时候连续的 k 个可以一起建，这样一次可以建 4 层。 update的时候，最深的 4 层可以用一开始建树的结果update，再往上的点直接将距离路径距离为 1 且不在路径上的点拉出来拍个序就行了，这样每次update只用 1 次 $super\_sort()$。

CF908G

定义 $S(n)$ 为将 n 在 10 进制下的所有数位从小到大排序后得到的数。例如：$S(1) = 1, S(50394) = 3459, S(323) = 233$。给定 X 求 $\sum_{i=1}^{X} S(i)$ 对 $10^9 + 7$ 取模的结果。

CF908G

定义 $S(n)$ 为将 n 在 10 进制下的所有数位从小到大排序后得到的数。例如：$S(1) = 1, S(50394) = 3459, S(323) = 233$。
给定 X 求 $\sum_{i=1}^{X} S(i)$ 对 $10^9 + 7$ 取模的结果。
数据范围：$1 \leq X \leq 10^{700}$。

CF908G

考虑如何计算答案，可以通过分别计算每一位对总和的贡献来求。定义 $cnt(i,\ x)$ 为 $S(1),\ S(2),\ ...,\ S(X)$ 中第 i 位为 x 的数量。

CF908G

考虑如何计算答案，可以通过分别计算每一位对总和的贡献来求。定义 $cnt(i,\ x)$ 为 $S(1),\ S(2),\ ...,\ S(X)$ 中第 i 位为 x 的数量。

直接求 $cnt(i,\ x)$ 仍然较为困难，考虑差分。令 $dlt(i,\ x)$ 为第 i 位上有多少个数 $\geq\ x$。对于一个 $S(y)$ 中第 i 位 $\geq\ x$ 的数 y，可以发现其必须满足有至少 i 个数位 $\geq\ x$，这样就可以进行dp了。

CF908G

考虑如何计算答案，可以通过分别计算每一位对总和的贡献来求。定义 $cnt(i,\ x)$ 为 $S(1),\ S(2),\ ...,\ S(X)$ 中第 i 位为 x 的数量。

直接求 $cnt(i,\ x)$ 仍然较为困难，考虑差分。令 $dlt(i,\ x)$ 为第 i 位上有多少个数 $\geq\ x$。对于一个 $S(y)$ 中第 i 位 $\geq\ x$ 的数 y，可以发现其必须满足有至少 i 个数位 $\geq\ x$，这样就可以进行dp了。

另 $dp(i,\ j,\ x,\ cmp)$ 表示填了前 i 位，有至少 j 个数字 $\geq\ x$，与 N 的大小关系位 cmp。

CF908G

考虑如何计算答案，可以通过分别计算每一位对总和的贡献来求。定义 $cnt(i,\ x)$ 为 $S(1),\ S(2),\ ...,\ S(X)$ 中第 i 位为 x 的数量。

直接求 $cnt(i,\ x)$ 仍然较为困难，考虑差分。令 $dlt(i,\ x)$ 为第 i 位上有多少个数 $\geq\ x$。对于一个 $S(y)$ 中第 i 位 $\geq\ x$ 的数 y，可以发现其必须满足有至少 i 个数位 $\geq\ x$，这样就可以进行dp了。

另 $dp(i,\ j,\ x,\ cmp)$ 表示填了前 i 位，有至少 j 个数字 $\geq\ x$，与 N 的大小关系位 cmp。

转移时直接枚举第 $i\ +\ 1$ 位数字是多少即可。

CF986C

给 n 个点。每个点有个权值 a_i $(0 \leq a_i < 2^m)$。两个点 u, v 之间有边当且仅当 $a_u \otimes a_v = 0$。问有多少个联通块。

CF986C

给 n 个点。每个点有个权值 a_i $(0 \leq a_i < 2^m)$。两个点 u, v 之间有边当且仅当 $a_u \otimes a_v = 0$。问有多少个联通块。
数据范围：$0 \leq m \leq 22$, $1 \leq n \leq 2^m$。

CF986C

求联通块个数不难想到用 bfs 算法。然而原图边数过多，所以想办法优化边数。

CF986C

求联通块个数不难想到用 bfs 算法。然而原图边数过多，所以想办法优化边数。
除了给定的 n 个点，新建 2^m 个点。对于原图中的每个点 i 连向新建的第 x 个点，其中 x 满足 $x \otimes a_i = 0$ 并且 x 的二进制表示中 1 的个数最多。并对新建的第 x 个点向第 y 个点连边，其中 y 满足 $x \otimes y = x$, $x \oplus y$ 为 2 的次幂。

CF1012F

你准备去 N 个国家进行旅行，去第 i 个国家的旅行会在第 s_i 天的早上出发，第 $s_i + len_i - 1$ 天的晚上回家。

CF1012F

你准备去 N 个国家进行旅行，去第 i 个国家的旅行会在第 s_i 天的早上出发，第 $s_i + len_i - 1$ 天的晚上回家。
你有 P 本护照，在每次旅行前必须让其中一本护照办理该国的签证，如果在第 x 天开始对某本护照办理第 i 个国家的签证，那么第 x 天不能在旅行，且第 $x + t_i$ 天中午可完成签证拿回该护照（允许在旅行时拿到）。判断旅行计划能否完成，如果能，给出一种签证方案（时间及哪本护照）。

CF1012F

你准备去 N 个国家进行旅行，去第 i 个国家的旅行会在第 s_i 天的早上出发，第 $s_i + len_i - 1$ 天的晚上回家。
你有 P 本护照，在每次旅行前必须让其中一本护照办理该国的签证，如果在第 x 天开始对某本护照办理第 i 个国家的签证，那么第 x 天不能在旅行，且第 $x + t_i$ 天中午可完成签证拿回该护照（允许在旅行时拿到）。判断旅行计划能否完成，如果能，给出一种签证方案（时间及哪本护照）。
数据范围：$1 \leq N \leq 22, 1 \leq P \leq 2$。

CF1012F

$P = 1$ 时怎么做？

CF1012F

$P = 1$ 时怎么做？

由于 N 的范围只有 22，不妨考虑状压dp。令 $f(S)$ 表示办完 S 集合的所有签证，拿回护照的时间最早是多少。

CF1012F

$P = 1$ 时怎么做？
由于 N 的范围只有 22，不妨考虑状压dp。令 $f(S)$ 表示办完 S 集合的所有签证，拿回护照的时间最早是多少。
转移时直接枚举下一个办签证的国家即可。

CF1012F

可以发现，如果有 2 个护照，对每个护照办签证的时间基本上是相似的，差别仅仅在于每个护照所签的国家进行旅行时该护照必须在身上。

CF1012F

可以发现，如果有 2 个护照，对每个护照办签证的时间基本上是相似的，差别仅仅在于每个护照所签的国家进行旅行时该护照必须在身上。

不妨修改 $f(S)$ 的定义，表示用一个护照办完 S 集合的所有签证，并且用这个护照旅行 S 的所有国家的拿回护照的最早时间。

CF1012F

可以发现，如果有 2 个护照，对每个护照办签证的时间基本上是相似的，差别仅仅在于每个护照所签的国家进行旅行时该护照必须在身上。
不妨修改 $f(S)$ 的定义，表示用一个护照办完 S 集合的所有签证，并且用这个护照旅行 S 的所有国家的拿回护照的最早时间。

转移时枚举下一个办签证的国家，判断是否可行并更新其它的 f 值。在判断可行时，可以按照下一个办签证国家时间长短的顺序依次枚举。可以证明这样求出的更新其它 f 的值是单调不减的，所以用指针维护即可。时间复杂度 $O(2^N\ N)$。

CF1012F

可以发现，如果有 2 个护照，对每个护照办签证的时间基本上是相似的，差别仅仅在于每个护照所签的国家进行旅行时该护照必须在身上。
不妨修改 $f(S)$ 的定义，表示用一个护照办完 S 集合的所有签证，并且用这个护照旅行 S 的所有国家的拿回护照的最早时间。

转移时枚举下一个办签证的国家，判断是否可行并更新其它的 f 值。在判断可行时，可以按照下一个办签证国家时间长短的顺序依次枚举。可以证明这样求出的更新其它 f 的值是单调不减的，所以用指针维护即可。时间复杂度 $O(2^N\ N)$。
最后求答案是只需要枚举第一个护照办理的签证即可。

三元环计数

给定一个 n 个点 m 条边无向图，问有多少个三元组 (u, v, w) 满足两两之间有边相连。

三元环计数

给定一个 n 个点 m 条边无向图，问有多少个三元组 (u, v, w) 满足两两之间有边相连。
将所有点按度数从小到大排序，如果度数相同按点编号排序，u 的排名记作 rnk_u。

三元环计数

给定一个 n 个点 m 条边无向图，问有多少个三元组 (u, v, w) 满足两两之间有边相连。
将所有点按度数从小到大排序，如果度数相同按点编号排序，u 的排名记作 rnk_u。
枚举三元组中 rnk 较小的两个点 u, v，再枚举与 u 相连的 w $(rnk_w > rnk_u, rnk_w > rnk_v)$。

三元环计数

复杂度分析

如果 u 的度数 $>$ $\sqrt{m}$ 则称 u 为大点，否则为小点。

三元环计数

复杂度分析

如果 u 的度数 $> \sqrt{m}$ 则称 u 为大点，否则为小点。

1. 如果 u, v 都是小点，则 u 的邻居不超过 $\sqrt{m}$ 个。

三元环计数

复杂度分析

如果 u 的度数 $> \sqrt{m}$ 则称 u 为大点，否则为小点。

1. 如果 u, v 都是小点，则 u 的邻居不超过 $\sqrt{m}$ 个。
2. 如果 u, v 至少一个是大点，则满足 $rnk_w > rnk_u$, $rnk_w > rnk_v$ 的 w 不超过 $\sqrt{m}$ 个。

三元环计数

复杂度分析

如果 u 的度数 $>\sqrt{m}$ 则称 u 为大点，否则为小点。

1. 如果 u, v 都是小点，则 u 的邻居不超过 $\sqrt{m}$ 个。
2. 如果 u, v 至少一个是大点，则满足 $rnk_w > rnk_u$, $rnk_w > rnk_v$ 的 w 不超过 $\sqrt{m}$ 个。

所以总复杂度 $O(m\sqrt{m})$。

四元环计数

给定一个 n 个点 m 条边无向图，问有多少个四元组 (u, v, w, x) 满足 $(u, v), (v, w), (w, x), (x, u)$ 有边相连。

四元环计数

给定一个 n 个点 m 条边无向图，问有多少个四元组 (u, v, w, x) 满足 $(u, v), (v, w), (w, x), (x, u)$ 有边相连。
将所有点按度数从小到大排序，如果度数相同按点编号排序，u 的排名记作 rnk_u。

四元环计数

给定一个 n 个点 m 条边无向图，问有多少个四元组 $(u,\ v,\ w,\ x)$ 满足 $(u,\ v),\ (v,\ w),\ (w,\ x),\ (x,\ u)$ 有边相连。
将所有点按度数从小到大排序，如果度数相同按点编号排序，u 的排名记作 rnk_u。
对于每个点 u，枚举两条边 $(u,\ v),\ (v,\ w)$，满足 $rnk_w > rnk_u,\ rnk_w > rnk_v$，每访问到一个 w 给答案加上 w 的标记，并给 w 的标记加 1。

四元环计数

复杂度分析

与三元环计数类似。

loj3070

给一个 n 个点 n 条边的无向连通图 $G = (V, E)$，定义 $d(u, v)$ 为 u 和 v 的最短距离，这里 G 中的每条边权值均为 1。
在所有满足 $u, v \in V$，且 $u < v$ 的点对 (u, v) 中随机选择一个，然后求出 $d(u, v)^k$ 的期望。
数据范围：$1 \leq n \leq 10^5, 1 \leq k \leq 10^9$。

loj3070

问题转化为求距离为 l 的有多少点对。

loj3070

问题转化为求距离为 l 的有多少点对。
如果是一棵树，通过点分治+NTT解决。

loj3070

问题转化为求距离为 l 的有多少点对。
如果是一棵树，通过点分治+NTT解决。
题目中的条件不难得出是一个环套树。先计算不用经过环上边的答案，与树计算方法一样。再计算经过环上边的情况。

loj3070

问题转化为求距离为 l 的有多少点对。
如果是一棵树，通过点分治+NTT解决。
题目中的条件不难得出是一个环套树。先计算不用经过环上边的答案，与树计算方法一样。再计算经过环上边的情况。
给环上节点标号 $(0,\ 1,\ 2,\ \ldots,\ m\ -\ 1)$，取中点 mid，$[0,\ mid]$ 区间内的点对以及 $[mid\ +\ 1,\ m\ -\ 1]$ 的点对距离也很好算。

loj3070

问题转化为求距离为 l 的有多少点对。
如果是一棵树，通过点分治+NTT解决。
题目中的条件不难得出是一个环套树。先计算不用经过环上边的答案，与树计算方法一样。再计算经过环上边的情况。
给环上节点标号 $(0,\ 1,\ 2,\ \ldots,\ m\ -\ 1)$，取中点 mid，$[0,\ mid]$ 区间内的点对以及 $[mid\ +\ 1,\ m\ -\ 1]$ 的点对距离也很好算。
考虑实现函数 $work(l1,\ r1,\ l2,\ r2)$ 来计算 $u\ \in\ [l1,\ r1],\ v\ \in\ [l2,\ r2]$ 的距离。

loj3070

考虑实现函数 $work(l1,\ r1,\ l2,\ r2)$ 来计算 $u \in [l1,\ r1],\ v \in [l2,\ r2]$ 的距离。

loj3070

考虑实现函数 $work(l1,\ r1,\ l2,\ r2)$ 来计算
$u \in [l1,\ r1],\ v \in [l2,\ r2]$ 的距离。
取 $[l1,\ r1]$ 中点 mid，计算出 $[l1,\ mid]$ 都能按照顺（逆）时针方向到达的 $[l2,\ r2]$ 区间 $[l2,\ a]$（$[b,\ r2]$），再递归调用 $work(l1,\ mid,\ a+1,\ b-1)$ 即可。

loj3070

考虑实现函数 $work(l1,\ r1,\ l2,\ r2)$ 来计算
$u \in [l1,\ r1],\ v \in [l2,\ r2]$ 的距离。
取 $[l1,\ r1]$ 中点 mid，计算出 $[l1,\ mid]$ 都能按照顺（逆）时针方向到达的 $[l2,\ r2]$ 区间 $[l2,\ a]$（$[b,\ r2]$），再递归调用 $work(l1,\ mid,\ a+1,\ b-1)$ 即可。
对 $[mid+1,\ r1]$ 同样处理。

loj3070

考虑实现函数 $work(l1,\ r1,\ l2,\ r2)$ 来计算
$u \in [l1,\ r1],\ v \in [l2,\ r2]$ 的距离。
取 $[l1,\ r1]$ 中点 mid，计算出 $[l1,\ mid]$ 都能按照顺（逆）时针方向到达的 $[l2,\ r2]$ 区间 $[l2,\ a]$（$[b,\ r2]$），再递归调用 $work(l1,\ mid,\ a+1,\ b-1)$ 即可。
对 $[mid+1,\ r1]$ 同样处理。
其本质就是分治ntt，故复杂度 $O(n\ log^2\ n)$。

loj3075

定义 $f(j) \equiv \sum_{i=0}^{n-1} C_{i\cdot d}^{j} \pmod M$, $0 \le f(j) < M$，其中 n, d, M 为给定值。
现在给定 m，输出 $f(0) \operatorname{xor} f(1) \operatorname{xor} f(2) \operatorname{xor} \cdots \operatorname{xor} f(m-1)$ 的值。
其中 C_n^m 为组合数 n 选 m，即

$$C_n^m = \begin{cases} \frac{n!}{m!\cdot(n-m)!} & ,0 \le m \le n \\ 0 & ,\text{otherwise} \end{cases}$$

。xor 表示异或和。
数据范围：$1 \le d \le 100$, $1 \le m \cdot d \le 3 \times 10^6$, $1 \le n \cdot d \le 10^9$, $10^8 \le M \le 10^9$。

loj3075

令 $C(x) = \sum_{i=0}^{n-1} (x+1)^{i \cdot d}$，则 $f(j) \equiv [x^j]C(x) \pmod{M}$。于是问题变成如何求 $C(x)$ 的前 m 项。

loj3075

令 $C(x) = \sum_{i=0}^{n-1} (x+1)^{i \cdot d}$，则 $f(j) \equiv [x^j]C(x) \pmod M$。于是问题变成如何求 $C(x)$ 的前 m 项。

类似等比数列求和，有：$C(x) = \frac{(x+1)^{n \cdot d} - 1}{(x+1)^d - 1}$。

令 $A(x) = \sum_{i=1}^{n \cdot d} \binom{n \cdot d}{i} \cdot x^{i-1}$，$B(x) = \sum_{i=1}^{d} \binom{d}{i} \cdot x^{i-1}$，那么

$$C(x) = \frac{A(x)}{B(x)},\ B(x) \cdot C(x) = A(x)$$

loj3075

$gcd(d,\ M)\ =\ 1$ 时的做法

loj3075

$gcd(d,\ M)\ =\ 1$ 时的做法

有

$$[x^i]C(x)\ =\ \frac{[x^i]A(x)\ -\ \sum_{j=1}^{d-1}\ ([x^j]B(x))\cdot([x^{i-j}]C(x))}{[x^0]B(x)}$$

由于 $[x^0]B(x)\ =\ \binom{d}{1}\ =\ d$，并且 $gcd(d,\ M)\ =\ 1$，所以 $[x^0]B(x)$ 在对 M 取模意义下的逆元存在，因此直接按照 i 从小到大的顺序对 $[x^i]C(x)$ 进行求解即可。

loj3075

$d = 2$ 时的做法

loj3075

$d = 2$ 时的做法

将 M 分解成 $2^a \cdot b$ 的形式，其中 $b \equiv 1 \pmod 2$。对每个 $f(j)$ 求出其对 2^a 取模意义下的值和对 b 取模意义下的值之后再使用中国剩余定理(CRT)算法就能求出 $f(j)$ 对 M 取模意义下的值。

loj3075

$d = 2$ 时的做法

将 M 分解成 $2^a \cdot b$ 的形式，其中 $b \equiv 1 \pmod 2$。对每个 $f(j)$ 求出其对 2^a 取模意义下的值和对 b 取模意义下的值之后再使用中国剩余定理(CRT)算法就能求出 $f(j)$ 对 M 取模意义下的值。计算对 b 取模的值可用之前算法，问题变成如何求对 2^a 取模下的值。

loj3075

$d = 2$ 时的做法

将 M 分解成 $2^a \cdot b$ 的形式，其中 $b \equiv 1 \pmod 2$。对每个 $f(j)$ 求出其对 2^a 取模意义下的值和对 b 取模意义下的值之后再使用中国剩余定理(CRT)算法就能求出 $f(j)$ 对 M 取模意义下的值。计算对 b 取模的值可用之前算法，问题变成如何求对 2^a 取模下的值。

当 $d = 2$ 时，$B(x) = x + 2$，有：

$2 \cdot [x^i]C(x) + [x^{i-1}]C(x) = [x^i]A(x)\ ,\ \forall\ i \geq 0$，即：

$[x^i]C(x) = [x^{i+1}]A(x) - 2 \cdot [x^{i+1}]C(x)$。

loj3075

$d = 2$ 时的做法

将 M 分解成 $2^a \cdot b$ 的形式，其中 $b \equiv 1 \pmod 2$。对每个 $f(j)$ 求出其对 2^a 取模意义下的值和对 b 取模意义下的值之后再使用中国剩余定理(CRT)算法就能求出 $f(j)$ 对 M 取模意义下的值。计算对 b 取模的值可用之前算法，问题变成如何求对 2^a 取模下的值。

当 $d = 2$ 时，$B(x) = x + 2$，有：

$2 \cdot [x^i]C(x) + [x^{i-1}]C(x) = [x^i]A(x)\ ,\ \forall\ i \geq 0$，即：

$[x^i]C(x) = [x^{i+1}]A(x) - 2 \cdot [x^{i+1}]C(x)$。

将右侧 $C(x)$ 的项不断用其带入可得

$[x^i]C(x) \equiv \sum_{j \geq 0} (-2)^j \cdot [x^{i+j+1}]A(x) \pmod{2^a}$，当 $j \geq a$ 时，$(-2)^j \cdot [x^{i+j+1}]A(x) \equiv 0 \pmod{2^a}$，故：

计算一项所需时间为 $O(a)$ 即 $O(log_2\ M)$。结合计算对 b 取模的部分，总时间复杂度为 $O(m \cdot (log_2 N + log_2 M + d))$。

loj3075

$d = 4$ 时的做法

loj3075

$d = 4$ 时的做法

与之前一样，将 M 分解成 $2^a \cdot b$ 的形式，考虑怎么算对 2^a 取模意义下的值。

loj3075

$d = 4$ 时的做法

与之前一样，将 M 分解成 $2^a \cdot b$ 的形式，考虑怎么算对 2^a 取模意义下的值。

当 $d = 4$ 时，$B(x) = 4 + 6x + 4x^2 + x^3$，有 $[x^i]C(x) = [x^{i+3}]A(x) - 4 \cdot [x^{i+3}]C(x) - 6 \cdot [x^{i+2}]C(x) - 4 \cdot [x^{i+1}]C(x)$。

loj3075

$d = 4$ 时的做法

与之前一样，将 M 分解成 $2^a \cdot b$ 的形式，考虑怎么算对 2^a 取模意义下的值。

当 $d = 4$ 时，$B(x) = 4 + 6x + 4x^2 + x^3$，有 $[x^i]C(x) = [x^{i+3}]A(x) - 4 \cdot [x^{i+3}]C(x) - 6 \cdot [x^{i+2}]C(x) - 4 \cdot [x^{i+1}]C(x)$。

可以发现对于上式的右侧来说，$C(x)$ 的每一项系数 $\equiv 0 \pmod 2$。与之前类似，我们称一轮操作为将式子右侧的每个 $C(x)$ 项用上式带入。如果操作前右侧式子中 $C(x)$ 每一项系数都是 2^i 的倍数，那么一轮操作后式子右侧 $C(x)$ 的每一项系数都将是 2^{i+1} 的倍数。如果进行 $a - 1$ 轮操作，$C(x)$ 的每一项系数将会 $\equiv 0 \pmod{2^a}$，此时，式子右侧就是若干 $A(x)$ 项的系数的和，问题就得到了解决。

loj3075

$d = 4$ 时的做法

迭代操作会进行 a 轮，迭代后的式子长度可以达到 $O(ad)$，对每一项展开需要 $O(d)$ 的复杂度，故计算迭代的式子需要复杂度为 $O(a^2d^2)$，即 $O((log_2 M)^2 \cdot d^2)$。之后计算 $[x^i]C(x)$ 每一项需要时间复杂度为 $O(log_2 M \cdot d)$，再加上计算对 b 取模意义下的结果，总时间复杂度为 $O(m \cdot (log_2 N + log_2 M \cdot d) + (log_2 M)^2 \cdot d^2)$。

loj3075

d 是质数时的做法

loj3075

d 是质数时的做法

与之前一样，将 M 分解成 $d^a \cdot b$ 的形式，其中 $gcd(b,\ d) = 1$，对模 d^a 和模 b 意义下分别求解。考虑对 d^a 取模怎么做。

loj3075

d 是质数时的做法

与之前一样，将 M 分解成 $d^a \cdot b$ 的形式，其中 $gcd(b,\ d) = 1$，对模 d^a 和模 b 意义下分别求解。考虑对 d^a 取模怎么做。

当 d 是质数时，有 $([x^{d-1}]B(x)) \cdot ([x^i]C(x)) = [x^{i+d-1}]A(x) - \sum_{j=0}^{d-2} ([x^j]B(x)) \cdot ([x^{i+d-1-j}]C(x))$。

loj3075

d 是质数时的做法

与之前一样，将 M 分解成 $d^a \cdot b$ 的形式，其中 $gcd(b,\ d) = 1$，对模 d^a 和模 b 意义下分别求解。考虑对 d^a 取模怎么做。

当 d 是质数时，有 $([x^{d-1}]B(x)) \cdot ([x^i]C(x)) = [x^{i+d-1}]A(x) - \sum_{j=0}^{d-2} ([x^j]B(x)) \cdot ([x^{i+d-1-j}]C(x))$。

观察前两个算法，都是对式子右侧的每一项 $C(x)$ 不断地用自己迭代下去，直到系数在模意义下都为 0 为止。那么对于 d 是质数的情况，只要能保证有限轮迭代之后能使右式在对 d^a 取模意义下只剩下若干个 $A(x)$ 项求和，那么就可以类似地做。

loj3075

d 是质数时的做法

由卢卡斯定理可知，
$\forall\ 0 < i < d,\ \binom{d}{i} \equiv \binom{1}{0} \cdot \binom{0}{i} \equiv 0 \pmod{d}$，即 $B(x)$ 的系数除了 $[x^{d-1}]B(x)$ 以外其余均是 d 的倍数。那么若迭代前 $C(x)$ 的系数均为 d^i 的倍数的话，迭代后的系数将会为 d^{i+1} 的倍数，因此经过 $a - 1$ 轮迭代后 $C(x)$ 的系数都会为 0。
总时间复杂度计算与上一个算法类似，为
$O(m \cdot (log_2 N + log_2 M \cdot d) + (log_2 M)^2 \cdot d^2)$。

loj3075

满分做法

loj3075

满分做法

将 M 分解质因数得 $M = \Pi_{i=1}^{k} p_i^{q_i}$，对每个 $p_i^{q_i}$ 计算取模意义下的值。问题变成如何对一个质数的次幂 p^q 进行求解。

loj3075

满分做法

将 M 分解质因数得 $M = \Pi_{i=1}^{k} p_i^{q_i}$，对每个 $p_i^{q_i}$ 计算取模意义下的值。问题变成如何对一个质数的次幂 p^q 进行求解。

当 $gcd(d, p) = 1$ 时，直接用逆元解。

loj3075

满分做法

将 M 分解质因数得 $M = \Pi_{i=1}^{k} p_i^{q_i}$，对每个 $p_i^{q_i}$ 计算取模意义下的值。问题变成如何对一个质数的次幂 p^q 进行求解。

当 $gcd(d, p) = 1$ 时，直接用逆元解。

当 $gcd(d, p) > 1$ 即 $d \equiv 0 \pmod{p}$ 时，仍然是想通过迭代使得右式 $C(x)$ 的系数在取模意义下为 0。当 $i \equiv 0 \pmod{p}$ 时，由卢卡斯定理 $\binom{d}{i} \equiv \binom{\frac{d}{p}}{\frac{i}{p}} \cdot \binom{0}{0} \equiv \binom{\frac{d}{p}}{\frac{i}{p}} \pmod{p}$，其结果不一定为 0，因此不能直接套用之前的迭代解法了。

loj3075

满分做法

迭代失败的原因主要是其迭代式子存在一个次数比 i 高的 $C(x)$ 项，其系数不一定为 p 的倍数，导致迭代后，新产生的系数不一定能使得所有系数都是 p 的更高次幂的倍数。如果我们选择系数不是 p 的倍数的 $C(x)$ 项中次数最高的那一项放在等式左侧，即找到最小的 t 使得 $[x^t]B(x) \not\equiv 0 \pmod p$，那么有：

$$\begin{aligned}([x^t]B(x)) \cdot ([x^i]C(x)) &= [x^{i+t}]A(x) \\ &- \sum_{j=0}^{t-1} ([x^j]B(x)) \cdot ([x^{i+t-j}]C(x)) \\ &- \sum_{j=t+1}^{d-1} ([x^j]B(x)) \cdot ([x^{i+t-j}]C(x))\end{aligned} \quad , \forall\ i \geq 0$$

其中当 $j < t$ 时 $[x^j]B(x)$ 为 p 的倍数，当 $j > t$ 时则不一定。

loj3075

满分做法

只要把上式中次数比 i 高的 $C(x)$ 项的系数在模 p^q 意义下变成 0 就行了。因此，只需要对次数比 i 高的 $C(x)$ 项迭代。如果直接迭代，在对次数为 j 的项进行迭代的时候，有可能向次数为 $k,\ (i < k < j)$ 的项产生一个非 p 倍数的系数，就不一定能保证有限轮迭代之后结束，而这样的问题只会向次数较低的项产生，那么很自然的想法是：假设当前次数比 i 高的每一个 $C(x)$ 项的系数都是 p^a 的倍数，每次找到次数最高的系数不是 p^{a+1} 的倍数的 $C(x)$ 项，对其进行迭代，直到所有 $C(x)$ 项的系数都是 p^{a+1} 的倍数为止，称这个过程叫做一轮，不难发现，一轮操作中需要迭代的项的次数是单调变低的，故一轮时间复杂度为 $O(l)$,其中 l 为操作前式子右侧 $C(x)$ 项个数。进行 $q-1$ 轮操作后，次数比 i 高的项的系数都变成了 0。在迭代中 $[x^i]C(x)$ 由于每次加上的都是一个为 p 的倍数的数，所以其仍然与 p^q 互质，逆元仍然存在，只用将等式两边同时乘上这个逆元就做完了。

祝大家 NOI 取得好成绩!