雅礼纪中联合杂题选讲

2020.5.3

纪中：

第一题 One （邓晗）

第二题 （林立）

第三题 （林凯风）

第四题 [JOISC2017]Arranging Tickets（曹天佑）

第一题 One （邓晗）

Cold_Chair

「LibreOJ β Round #3」绯色 IOI（悬念）

- 有一个左边m个点右边n个点的二分图。
- 对于左边的每个点i，它连着右边的两个点，分别是$(a[i]+b[i]) mod\ n$和$(a[i]-b[i])\ mod\ n$，边有边权，数据保证左侧满匹配。
- q次修改，每次修改一条边的边权，每次修改后问最大权最大匹配。
- 强制在线。

数据范围与提示

对于所有数据，$1 \le m \le n \le 5 \times 10^5, n$ 是奇数 $, 0 \le a_i < n, 0 \le b_i \le \lfloor \frac{n}{2} \rfloor, 0 \le q \le 8 \times 10^5$，任何时刻每对关系的愉悦程度是 $[0, 1300]$ 中的整数。

数据保证存在一个匹配使得每个妹子都能找到自己的伙伴。

Subtask #	分值	m, n 的限制	q 的限制	T 的限制
1	7	$1 \le n \le 9$	$0 \le q \le 9$	$T = 0$
2	10	$1 \le n \le 18$	$0 \le q \le 18$	
3	17	$1 \le n \le 100$	$0 \le q \le 100$	
4	15	$1 \le n \le 5000$	$0 \le q \le 5000$	$T \in \{0, 1\}$
5	11	$1 \le n \le 2 \times 10^5$	$q = 0$	$T = 0$
6	13		$0 \le q \le 2 \times 10^5$	$T \in \{0, 1\}$
7	12	$1 \le n \le 5 \times 10^5$	$0 \le q \le 5 \times 10^5$	$T = 0$
8	15			$T \in \{0, 1\}$

小问题

- 怎么判断左边满配？

Hall定理？

- 由Hall定理，得左边任意一个点集S，连向右边的点集并T要满足$|T| \geq |S|$
- 反过来，对于右边的任意一个点集T，对于两条边右端点都在T的左端点集S，有$|T| \geq |S|$
- 如果左i连向右$u[i], v[i]$，视作有一条边$(u[i], v[i])$。
- 也就是对任意点集T，有T的导出子图的点数$\geq$ 边数。
- 不妨假设T是一个联通块（不是联通块可以划分成若干联通块，约束更紧），则约束说明T要么是树，要么是基环树。

- 现在一个匹配相当于给所有边定向，要求每个点入度不超过1，的最大权值和。
- 一次询问：我会树形dp

- 多次询问：我会ddp
- 有没有更简单的方法？

思考

- 对于一个基环树，树上的边的方向是确定的，环上只有两种方向。

- 对于一个树，一种定向方法相当于选择一个根，做外向树。
- 先随便选择一个点作根，讨论一条边的贡献，发现要么是对子树外的点作根有贡献，要么是对子树内的点作根有贡献。
- 用线段树维护dfs序上的区间加、区间最大值即可。

第二题 林立

对于一个01序列A，我们定义一次匹配(i,j)(i<j)为合法的，当且仅当

1）i为偶数，j为奇数

2）A[i]=A[j]

3）i和j没有被匹配

对于一个长度为2m的01序列P，我们称其为合法的，当且仅当尽可能多的匹配之后，没有被匹配的位置的数量<=2n。

求有多少个合法的序列，答案对998244353取模。

n,m<=3000

加强版 n,m<=100000

简单版Solution

考虑暴力，设F[i][j][k]为前i组，剩j个0,k个1。

假设一开始我们有n个可匹配的偶数位置(可0可1)，我们会发现我们转移过程中可供匹配的位置数量恒为n。

那么修改一下状态，F[i][j]表示前i组有j个0，那么就有n-j个1。

但是这样会算重，一个合法方案可能有很多种分配n个可匹配的位置的01的方案。

注意到一定有一种方案满足某个时刻可匹配的位置用完了。

再加一维0/1表示转移过程中是否经过j=0。

加强版Solution

考虑dp的转移，对于一个j的位置会转移到j-1, j, j+1

计算出不包含j到j的转移 转移i次的方案数，然后将剩下的m-i个j到j的转移插入到这些转移中就能得到总的方案数

枚举第一个经过j=0的位置，显然后一部分的转移是不能低于0超过n，而前一部分是不能低于1，所以计算方法是一样的。

如果能够得到前后部分长度为i时的值，那么卷积后就能得到答案。

延伸问题

$F[i][j]=F[i-1][j-1]*[j>0]+F[i-1][j+1]*[j<m]$

对于每一个$i(1<=i<=n)$，求$G[i]=\sum F[i][j]$。

可以将$F[i-1]$到$F[i]$的转移看成乘一个矩阵。

根据哈密顿-凯莱定理，G满足一个线性递推的关系。

既然G满足线性递推关系，那么只要求出递推式与初值就能使用多项式运算得到G。

初值

由于是n+1阶递推式，所以要求0~n位置的值，并且没有了上限。

那么对于i，就是求有多少个i长度的序列（值只有+-1）,前缀和都>=0的个数。

枚举j表示总和（j=i mod 2），用总减去不合法。

对于不合法的方案就将不合法的第一个位置前面的位置都翻转，那么总和一定为j+2。

从(0,0)到(i,j)的方案数为C(i,(i-j)/2)。

对于i总的方案数就是∑C(i,(i-j)/2)-C(i,(i-j-2)/2)=C(i,i/2)

递推式

根据哈密顿-凯莱定理，递推式就是矩阵的特征多项式。

而求矩阵A的特征多项式，就是求det(xI_n-A) （I_n是n*n的单位矩阵）

这个矩阵比较特殊A[i][i+1]=A[i][i+1]=1（1<=i<=n-1）

设F[i]表示这种类型i*i大小的矩阵的特征多项式。

F[i]=x*F[i-1]-F[i-2]

这是因为选(1,2)，就一定要选(2,1)，那么矩阵大小-2。

[x^k]F[n]=[n=k(mod 2)]C((n+k)/2,k)*(-1)^((n-k)/2)

考虑将这个F的下标看为第一维，多项式的指数看为第二维，转移是(+1,+1)，(+2,+0)。

每次一共+2，第一种一共有k次。

最后的最后

如果直接将原来的转移整个看成一个矩阵。

矩阵是(2n+1)*(2n+1)的大小，

A[i][i]=2,A[i][i+1]=A[i+1][i]=1,特殊的A[n+1][n]=0

递推式就是将前面的x代换成x-2，去掉A[n+1][n]的影响，而去掉A[n+1][n]的是剩下两个(n-1)*(n-1)矩阵。

初值可以使用前面的方法求。

剩下的就是齐次线性递推。

加加强版 m<=10^18,n<2^15

纪中第3题（林凯风）

N个点的竞赛图，求只包含前i个点的图有多少个强连通分量。

$n \le 2e5$

边以以下方式给出：

$(l,r,x,0)$表示对于点$i(l \le i \le r)$ i和x之间的边是从$(i->x)$

$(l,r,x,1)$即这些边的方向是$(x->i)$

其中$(l \le r < x)$

一共$m(m \le 2e6)$条这样的信息。

暴力：
前i个点组成的子图显然也是竞赛图。一个竞赛图缩点之后一定是一条链的形式。

按顺序加入每一个点，加入的点要么自成一个强连通分量，要么与一段连续的强连通分量合并成一个更大的强连通分量。

直接维护缩点后的拓扑序，找到这个点出边中拓扑序最小的和入边中拓扑序最大的，这个点与中间的会形成一个新的强连通分量。
暴力做的复杂度是n^2的

优化：
给边的方式是一段编号连续的区间与这个点的边方向相同，即只需要求出编号连续的区间里所在强连通分量拓扑序最大（最小）的点。

建一棵线段树分别维护最小和最大的。每次加点的时候只会影响新加的点与其他点的大小关系。
用一个平衡树维护拓扑序。可以在logn的时间内比较两个点的拓扑序关系。

时间是两个log的

离线优化：
对于区间[l,r]，当点r加入这张图的时候，拓扑序的最大（最小）的点就已经确定了。即找到一个在点r加入时[l,r]内拓扑序最大强连通分量内的任意一个点，这个点所在的强连通分量一定是拓扑序最大的强连通分量。

当点r加入的时候，可以直接查询所有以r为右端点的区间的答案。
维护一个单调栈，包含所有符合下面条件的点：所有编号比他大的点拓扑序都比他小。查询的时候就只需要在单调栈上二分就好了。加入一个点时跟栈顶比较直到栈顶拓扑序比他大。这样可以求出区间内拓扑序最大的点。

一共只需要n次比较，时间是一个log的

联合杂题

中山纪念中学 曹天佑

[JOISC2017]Arranging Tickets

- 在一个环形线路上有n个火车站，编号为1,2,3,…,n
- 有n种车票，第i种车票可以从i号车站前往i+1号车站，也可以从i+1号车站前往i号车站
- 有m类旅客，第i类旅客有c[i]人，要从车站a[i]前往车站b[i]
- 你可以为每一位旅客确定线路，最小化所有车票需要数目的最大值
- n<=2*10^5,m<=10^5

Solution

- 首先我们破环为链
- 问题可以变成这样：一个数轴上有n个整点，有m种区间。第i种区间覆盖了[a[i],b[i])的整点，有c[i]个。
- 你可以选择若干个区间翻转，[l,r)变为[1,l)∪[r,n]
- 最小化被覆盖最多的整点被覆盖的次数。
- 显然答案具有二分性，考虑二分答案并判定

Solution

- 通过观察我们可以得出一个结论，若区间[l,r)和区间[x,y)没有交，这两个区间不会同时翻转
- 这告诉我们所有被翻转的区间一定有交，也就是存在一个公共交点t，考虑枚举t
- 把区间按左端点排序，从1到t考虑每个点，如果该点被覆盖次数>mid就选择当前右端点最大的区间翻转
- 不过这样贪心后面翻转的区间可能导致前面的点的覆盖次数增加，我们需要预先知道总的翻转次数cnt

Solution

- 考虑当前点覆盖次数为y，后面剩余的翻转次数为T，我们需要满足y+T<=mid
- 否则考虑翻转，设翻转次数为x，需要满足y-x+T-x<=mid，取最小的合法的x即可
- 这样我们能保证1~t的点合法，并且最小化了t+1~n的覆盖次数，最后再check一次即可。
- 但好像我们需要枚举的东西有点多？

Solution

- 我们需要确定交点t和总翻转次数cnt，但是显然不能枚举
- 考虑a[i]为最初状态下每一个点被覆盖的次数，b[i]为翻转之后每个点被覆盖的次数

Solution

- 定理1：一定存在一种最优的翻转方法使得b[t]=max(b[i])或b[t]=max(b[i])-1

- 证明：
- 设最后所有翻转区间的交为[l,r)，我们取b[t]=max(b[l]~b[r-1])
- 考虑反证，即b[t]<=max(b[i])-2
- 我们选择右端点最左的和左端点最右的区间(可能是同一个)，取消他们的翻转
- 那么b[l]~b[r-1]会+2，max(b[i])不变或者-2，所有位置依旧合法
- 依次操作直到b[t]>=max(b[i])-1为止

Solution

- 又因为有解的情况下max(b[i])=mid
- 我们只需要检验cnt=a[t]-mid和cnt=a[t]-(mid-1)即可
- 这样子我们的复杂度就和c[i]无关了，可以做到O(n^2 log^2 n)

Solution

- 定理2：最优解满足a[t]=max(a[i])

- 证明：
- 考虑反证，也就是存在$x \notin [l,r)$，使得a[x]>=a[t]+1
- 既然这样一定存在一个翻转后的区间包含x不包含t
- 也就是b[x]-a[x]>=b[t]-a[t]+1
- 可以得到b[x]>=b[t]+2，和前一个矛盾

Solution

- 但现在还是有一个问题，如果有很多个t满足a[t]=max(a[i])，该选哪个？

- 定理3：t只需要取a[t]=max(a[i])的最左和最右即可

- 证明：
- 依旧考虑反证法，令a[l]和a[r]是最左和最右的，最优解选择了a[t]
- 我们需要证明所有翻转的区间都覆盖了a[l]或者a[r]
- 首先显然不存在翻转的区间[x,y)满足y<=l或者x>r

Solution

- 引理1：不存在翻转区间[x,y)满足l<x<y<=r

- 证明：
- 考虑b’为不翻转[x,y)得到的b数组，有b’[l]=b[l]-1,b’[t]=b[t]+1
- 然后无论什么时候都有b’[l]-a[l]>=b’[t]-a[t]，a[t]=a[l]得到b’[l]>=b’[t]
- 即b[l]>=b[t]+2，和前面的事实矛盾

Solution

- 引理2：
- 假设两个翻转区间[x1,y1)，[x2,y2)满足y2<=r且x1>l，同时取消翻转答案不会更劣

- 证明：
- 首先max(a[l]~a[r])=a[l]=a[r]，b[i]-a[i]的值越远离t越大
- 我们有max(b[l]~b[r])=max(b[l],b[r])
- 那么就有max(b[i])=max(b[j]),j<=l或j>=r
- 这个事实对任何时候的b都满足
- 那么翻转之后得到的max(b[i])不会增大

Solution

- 综上我们只需要取t=l，t=r，cnt=a[t]-mid，cnt=a[t]-mid+1分别检测即可
- 复杂度O(n log^2 n)

题目交流

Hirasawa Yui

2020.5.3

题目大意

给定一个序列，要求单点修改，查询出现超过 p 的数字，允许多输出至多 5 个

暴力

直接做，按照统计众数的方法

优化暴力

考虑暴力的做法，每次加入一个数字，就把他的次数增加 1，如果没有维护在队列中，就把其他的减少 1，那么，这个东西很容易发现是有结合律的，也就是说不依赖于顺序，那么可以套用一个线段树来维护，每次合并两个区间的信息，把其中一个集合暴力插入另一个集合，这样就可以得到最终出现次数最多的那些数字，由于一定出现了超过 p 次，所以这里只保留前 5 个一定是包含答案集合的

拓展一下这个做法，可以很容易的实现区间修改

改变？

不能允许多输出，必须恰好是答案集合？

蒙特卡洛

考虑这里的数字出现次数都不是很少，至少都有 $\frac{1}{5}$，那么是否可以考虑直接得到他们？

每次随机一个位置，查询这个位置的数字，并且查询他的出现次数，如果满足就直接加入答案，随意很多次，去命中答案集合，这个算法实现起来很简单

主要是考虑正确的概率，不放令 t 是最小出现频率 (p)，那么分析我们随机撒点 r 次，不命中的概率 $\frac{1}{(1-t)^{L}IM}$，在撒点 100 次时，这个概率小到了 $2e-10$，再带入询问次数 $(1-2e-10)^{10^5}$，这个概率是 0.99998 以上，也就是说我们操作这么多次，错误的概率已经可以忽略不计了，那么直接这样就可以轻松的对所有数据做到极高的正确概率通过，即使是在 Codeforces 上，也没有被 Hack

2019 百度之星复赛 1004 Maximum or Sum

给定长为 $n-1$ 的数组 $\{c\}$ 。定义正整数数组 $\{a\}$ 合法，当且仅当它长为 n ，且对任意 i 都有 $\max(a_i, a_{i+1}) = c_i$ 或者 $a_i + a_{i+1} = c_i$ 。

求合法 $\{a\}$ 的个数。

$n \leq 3 \times 10^3$

令 $d_i = \min(c_{i-1}, c_i)$ ，有 $a_i \in [1, d_i]$ 。

令 $d_i = \min(c_{i-1}, c_i)$ ，有 $a_i \in [1, d_i]$ 。

我们从前往后去确定 a_i 的值，可以发现 a_i 与 c_i 是否相等会对 a_{i+1} 产生两种完全不同的影响。

令 $d_i=\min(c_{i-1},c_i)$ ，有 $a_i\in[1,d_i]$ 。

我们从前往后去确定 a_i 的值，可以发现 a_i 与 c_i 是否相等会对 a_{i+1} 产生两种完全不同的影响。

- 若 $a_i=c_i$ ，这时 a_{i+1} 只会受到 $[1,d_{i+1}]$ 的取值范围限制。

令 $d_i = \min(c_{i-1}, c_i)$ ，有 $a_i \in [1, d_i]$ 。

我们从前往后去确定 a_i 的值，可以发现 a_i 与 c_i 是否相等会对 a_{i+1} 产生两种完全不同的影响。

- 若 $a_i = c_i$ ，这时 a_{i+1} 只会受到 $[1, d_{i+1}]$ 的取值范围限制。
- 若 $a_i \neq c_i$ ，这时在取值范围的限制下，必须有以下两者之一

令 $d_i = \min(c_{i-1}, c_i)$ ，有 $a_i \in [1, d_i]$ 。

我们从前往后去确定 a_i 的值，可以发现 a_i 与 c_i 是否相等会对 a_{i+1} 产生两种完全不同的影响。

- 若 $a_i = c_i$ ，这时 a_{i+1} 只会受到 $[1, d_{i+1}]$ 的取值范围限制。
- 若 $a_i \neq c_i$ ，这时在取值范围的限制下，必须有以下两者之一
  1. $a_{i+1} = c_i$
  2. $a_{i+1} = c_i - a_i$

令 $d_i = \min(c_{i-1}, c_i)$ ，有 $a_i \in [1, d_i]$ 。

我们从前往后去确定 a_i 的值，可以发现 a_i 与 c_i 是否相等会对 a_{i+1} 产生两种完全不同的影响。

- 若 $a_i = c_i$ ，这时 a_{i+1} 只会受到 $[1, d_{i+1}]$ 的取值范围限制。
- 若 $a_i \neq c_i$ ，这时在取值范围的限制下，必须有以下两者之一
  1. $a_{i+1} = c_i$
  2. $a_{i+1} = c_i - a_i$

初步设 f_i 表示 $a_i = c_i$ 时，$a\{i+1, n\}$ 的取值方案数；g_i 表示所有 $a_i \neq c_i$ 的情况下，$a\{i+1, n\}$ 的取值方案数之和。

令 $d_i = \min(c_{i-1}, c_i)$ ，有 $a_i \in [1, d_i]$ 。

我们从前往后去确定 a_i 的值，可以发现 a_i 与 c_i 是否相等会对 a_{i+1} 产生两种完全不同的影响。

- 若 $a_i = c_i$ ，这时 a_{i+1} 只会受到 $[1, d_{i+1}]$ 的取值范围限制。
- 若 $a_i \neq c_i$ ，这时在取值范围的限制下，必须有以下两者之一
    1. $a_{i+1} = c_i$
    2. $a_{i+1} = c_i - a_i$

初步设 f_i 表示 $a_i = c_i$ 时，$a\{i+1, n\}$ 的取值方案数；g_i 表示所有 $a_i \neq c_i$ 的情况下，$a\{i+1, n\}$ 的取值方案数之和。

如果能写出转移式，最终答案就是 $f_1 + g_1$ 。

根据之前的性质推导，我们有

根据之前的性质推导，我们有

- 若满足 $c_i = d_i$，取 $a_i = c_i$ ，有 $f_{i+1} + g_{i+1} \to f_i$

根据之前的性质推导，我们有

- 若满足 $c_i = d_i$，取 $a_i = c_i$ ，有 $f_{i+1} + g_{i+1} \rightarrow f_i$
- 用 $[l, r]$ 表示 $a_i \neq c_i$ 时，a_i 的取值范围，a_{i+1} 所对应的取值范围就应该是 $[l' = c_i - r, r' = c_i - l] \cap [1, d_{i+1}]$ ，而且是 a_{i+1} 决定 a_i

根据之前的性质推导，我们有

- 若满足 $c_i = d_i$ ，取 $a_i = c_i$ ，有 $f_{i+1} + g_{i+1} \to f_i$
- 用 $[l, r]$ 表示 $a_i \neq c_i$ 时，a_i 的取值范围，a_{i+1} 所对应的取值范围就应该是 $[l' = c_i - r, r' = c_i - l] \cap [1, d_{i+1}]$ ，而且是 a_{i+1} 决定 a_i
    1. 若满足 $c_i = d_{i+1}$ ，取 $a_{i+1} = c_i$ ，有 $f_{i+1} \to g_i$

根据之前的性质推导，我们有

- 若满足 $c_i = d_i$，取 $a_i = c_i$，有 $f_{i+1} + g_{i+1} \to f_i$
- 用 $[l, r]$ 表示 $a_i \neq c_i$ 时，a_i 的取值范围，a_{i+1} 所对应的取值范围就应该是 $[l' = c_i - r, r' = c_i - l] \cap [1, d_{i+1}]$，而且是 a_{i+1} 决定 a_i
  1. 若满足 $c_i = d_{i+1}$，取 $a_{i+1} = c_i$，有 $f_{i+1} \to g_i$
  2. 把 $c_i = d_{i+1}$ 的情况在 $[l', r']$ 中除去，有 a_{i+1} 取值 $[l', r']$ 时，$a\{i+2, n\}$ 的取值方案数之和 $\to g_i$

根据之前的性质推导，我们有

- 若满足 $c_i = d_i$，取 $a_i = c_i$，有 $f_{i+1} + g_{i+1} \to f_i$
- 用 $[l, r]$ 表示 $a_i \neq c_i$ 时，a_i 的取值范围，a_{i+1} 所对应的取值范围就应该是 $[l' = c_i - r, r' = c_i - l] \cap [1, d_{i+1}]$，而且是 a_{i+1} 决定 a_i
  1. 若满足 $c_i = d_{i+1}$，取 $a_{i+1} = c_i$，有 $f_{i+1} \to g_i$
  2. 把 $c_i = d_{i+1}$ 的情况在 $[l', r']$ 中除去，有 a_{i+1} 取值 $[l', r']$ 时，$a\{i + 2, n\}$ 的取值方案数之和 $\to g_i$

可以发现圈 2 与原问题的模式非常相似，于是传参 $[l, r]$ 递归求解即可。

根据之前的性质推导，我们有

- 若满足 $c_i = d_i$，取 $a_i = c_i$ ，有 $f_{i+1} + g_{i+1} \to f_i$
- 用 $[l, r]$ 表示 $a_i \neq c_i$ 时，a_i 的取值范围，a_{i+1} 所对应的取值范围就应该是 $[l' = c_i - r, r' = c_i - l] \cap [1, d_{i+1}]$，而且是 a_{i+1} 决定 a_i
    1. 若满足 $c_i = d_{i+1}$ ，取 $a_{i+1} = c_i$ ，有 $f_{i+1} \to g_i$
    2. 把 $c_i = d_{i+1}$ 的情况在 $[l', r']$ 中除去，有 a_{i+1} 取值 $[l', r']$ 时，$a\{i+2, n\}$ 的取值方案数之和 $\to g_i$

可以发现圈 2 与原问题的模式非常相似，于是传参 $[l, r]$ 递归求解即可。

复杂度是 $O(n^2)$ 的。

一道简单题

Caspian

May 2, 2020

雅礼中学

水题

「UR 19」通用测评号

有 n 个燃料舱，容量均为 a，初始时均为空。

每次等概率随机选择一个没有满的燃料舱，并在其中放入 1 单位燃料。当所有燃料舱均包含至少 b 单位的燃料时，停止该过程

求结束时期望有多少个燃料舱被填满了，答案对 998244353 取模

$1 \leq n \leq 250, 1 \leq b < a \leq 250$

Solution

原问题转化为：每个燃料舱容量无限，求在所有燃料舱 $\geq b$ 的时刻，燃料 $\geq a$ 的燃料舱个数的期望。

Solution

原问题转化为：每个燃料舱容量无限，求在所有燃料舱 $\geq b$ 的时刻，燃料 $\geq a$ 的燃料舱个数的期望。

根据期望的线性性，只需要对每个燃料舱考虑，计算在它燃料 $= a$ 的时刻存在其他燃料舱 $< b$ 的概率（在这个局面下，最终所有燃料舱 $\geq b$ 时，该燃料舱必然 $\geq a$），此时概率即为期望。

Solution

原问题转化为：每个燃料舱容量无限，求在所有燃料舱 $\geq b$ 的时刻，燃料 $\geq a$ 的燃料舱个数的期望。

根据期望的线性性，只需要对每个燃料舱考虑，计算在它燃料 $= a$ 的时刻存在其他燃料舱 $< b$ 的概率（在这个局面下，最终所有燃料舱 $\geq b$ 时，该燃料舱必然 $\geq a$），此时概率即为期望。

而所有燃料舱等价，那么对 1 号燃料舱进行计算，乘上 n 即为答案

Solution

考虑容斥，钦定有 p 个燃料舱燃料 $< b$，容斥系数为 $(-1)^{p-1}\binom{n-1}{p}$

Solution

考虑容斥，钦定有 p 个燃料舱燃料 $< b$，容斥系数为 $(-1)^{p-1}\binom{n-1}{p}$

问题转化为：有一个初始为空的序列，每次等概率随机添加一个 $[1, p+1]$ 的数至序列末尾，计算当序列中出现 a 个 1 时，$2 \sim p+1$ 的出现次数均 $< b$ 的概率。

Solution

考虑容斥，钦定有 p 个燃料舱燃料 $< b$，容斥系数为 $(-1)^{p-1}\binom{n-1}{p}$

问题转化为：有一个初始为空的序列，每次等概率随机添加一个 $[1, p+1]$ 的数至序列末尾，计算当序列中出现 a 个 1 时，$2 \sim p+1$ 的出现次数均 $< b$ 的概率。

可以枚举 x_i 表示数字 i 的出现次数，先钦定序列末尾一定是 1，再把剩下的 $a-1$ 个 1 以及 $2 \sim p+1$ 这些数用多重排列统计方案数。而每个元素被选到的概率都为 $\frac{1}{p+1}$，所以有

$$ans = \sum_{x_2, x_3, \ldots, x_{p+1} \in [0, b-1]} \frac{(a-1+\sum x_i)!}{(a-1)!(\prod x_i)!} \left(\frac{1}{p+1}\right)^{a+\sum x_i}$$

Solution

注意到答案只与 $\sum x_i$ 和 $(\prod x_i)!$ 有关，于是枚举 $j=\sum x_i$ 后可以写成 EGF 的形式。

$$ans=\sum_{j=0}^{p(b-1)}\frac{(a-1+j)!}{(a-1)!}\left(\frac{1}{p+1}\right)^{a+j}[x^j]\left(\sum_{i=0}^{b-1}\frac{x^i}{i!}\right)^p$$

Solution

注意到答案只与 $\sum x_i$ 和 $(\prod x_i)!$ 有关，于是枚举 $j=\sum x_i$ 后可以写成 EGF 的形式。

$$ans=\sum_{j=0}^{p(b-1)}\frac{(a-1+j)!}{(a-1)!}\left(\frac{1}{p+1}\right)^{a+j}[x^j]\left(\sum_{i=0}^{b-1}\frac{x^i}{i!}\right)^p$$

可以用 NTT 求出 $\left(\sum_{i=0}^{b-1}\frac{x^i}{i!}\right)^p$，时间复杂度为 $O(n^3\log n)$

Solution

进一步优化，令 $F(x)=\sum_{i=0}^{b-1}\frac{x^i}{i!}$，对 F 求导则有

$$F' = F - \frac{x^{b-1}}{(b-1)!}$$

$$(F^2)' = 2F^2 - 2F\frac{x^{b-1}}{(b-1)!}$$

$$\cdots$$

$$(F^i)' = iF^i - iF^{i-1}\frac{x^{b-1}}{(b-1)!}$$

$$\Rightarrow F^i = \frac{(F^i)'}{i} + F^{i-1}\frac{x^{b-1}}{(b-1)!}$$

Solution

进一步优化，令 $F(x)=\sum_{i=0}^{b-1}\frac{x^i}{i!}$，对 F 求导则有

$$
\begin{aligned}
F' &= F-\frac{x^{b-1}}{(b-1)!}\\
(F^2)' &= 2F^2-2F\frac{x^{b-1}}{(b-1)!}\\
&\cdots\\
(F^i)' &= iF^i-iF^{i-1}\frac{x^{b-1}}{(b-1)!}\\
\Rightarrow F^i &= \frac{(F^i)'}{i}+F^{i-1}\frac{x^{b-1}}{(b-1)!}
\end{aligned}
$$

从高位到低位依次确定 F^i 的系数即可，时间复杂度 $O(n^3)$

The End

Thanks

CF506E 解题报告

EndSaH

2020 年 4 月 30 日

Yali High School

题目描述

给定一个小写字符串 s 以及正整数 n，求在 s 中插入 n 个小写字符后使其回文的方案数。

另一种理解是：有多少个长度为 $n+|s|$ 的回文串，满足其包含 s 作为子序列。

答案对 10007 取模。

$n \leq 10^9, |s| \leq 200$。

初步思路——DP

默认下标从 1 开始。

先考虑 $n+|s|$ 为偶数的情况，奇数的不难由偶数导出。

设 $f_{i,l,r}$ 表示已经考虑好了该回文串的前 i 位和后 i 位，当前 s 的 $[l,r]$ 这段区间还未匹配上的串数。

另设 g_i 表示已经考虑好了该回文串的前 i 位和后 i 位，其已经包含了 s 为子序列的串数。转移如下：

初步思路——转移方程

无论什么情况，$g_{i+1} \leftarrow 26g_i$。

初步思路——转移方程

无论什么情况，$g_{i+1} \leftarrow 26g_i$。

1. $s_l = s_r$：此时有填 s_l 这个字符或填其他字符两种选择。

初步思路——转移方程

无论什么情况，$g_{i+1} \leftarrow 26g_i$。

1. $s_l = s_r$：此时有填 s_l 这个字符或填其他字符两种选择。
   1.1 $r - l \leq 1$:
   $f_{i+1,l,r} \leftarrow 25f_{i,l,r}$
   $g_{i+1} \leftarrow f_{i,l,r}$

初步思路——转移方程

无论什么情况，$g_{i+1} \leftarrow 26g_i$。

1. $s_l = s_r$：此时有填 s_l 这个字符或填其他字符两种选择。
   1.1 $r - l \leq 1$:
   $f_{i+1,l,r} \leftarrow 25f_{i,l,r}$
   $g_{i+1} \leftarrow f_{i,l,r}$
   1.2 $r - l > 1$:
   $f_{i+1,l,r} \leftarrow 25f_{i,l,r}$
   $f_{i+1,l+1,r-1} \leftarrow f_{i,l,r}$

初步思路——转移方程

无论什么情况，$g_{i+1} \leftarrow 26g_i$。

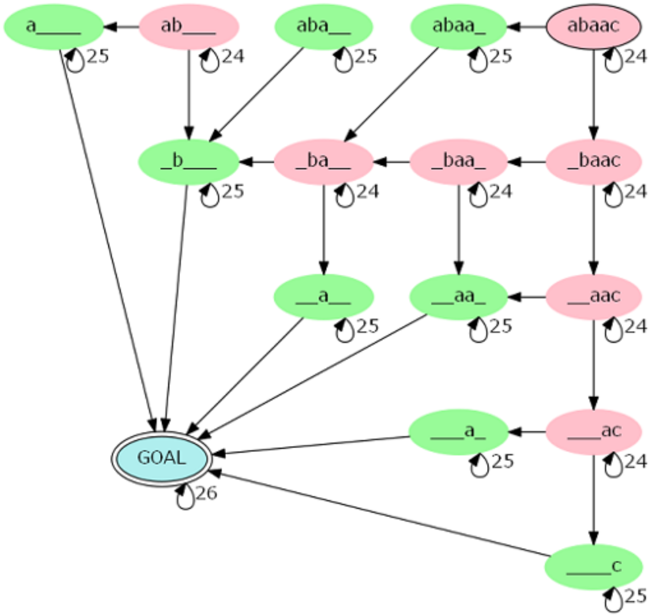
1. $s_l = s_r$：此时有填 s_l 这个字符或填其他字符两种选择。
   1.1 $r - l \leq 1$:
   $f_{i+1,l,r} \leftarrow 25f_{i,l,r}$
   $g_{i+1} \leftarrow f_{i,l,r}$
   1.2 $r - l > 1$:
   $f_{i+1,l,r} \leftarrow 25f_{i,l,r}$
   $f_{i+1,l+1,r-1} \leftarrow f_{i,l,r}$
2. $s_l \neq s_r$：此时有三种选择：填 s_l, s_r 和其他。注意这里不可能会转移到 g。
   $f_{i+1,l,r} \leftarrow 24f_{i,l,r}$
   $f_{i+1,l+1,r} \leftarrow f_{i,l,r}$
   $f_{i+1,l,r-1} \leftarrow f_{i,l,r}$

更进一步——状态优化

直接用上面的 DP，或者矩乘加速转移，复杂度分别是 $O(|s|^2 n)$ 与 $O(|s|^6 \log n)$，任重道远。

沿着矩乘的思路走，思考一下怎么压缩状态。忽略第一维，把每个状态放到图上表示成一个点，转移写成有向边，画个图出来：

更进一步——状态优化


a____
ab___
aba__
abaa_
abaac
25
24
25
25
24
_b___
_ba__
baa
_baac
25
24
24
24
__a__
__aa_
__aac
25
25
24
___a_
___ac
25
24
GOAL
26
____c
25

更进一步——状态优化

此处 $s=$ abaac。

图里面每个节点上的字符串表示还没有匹配上的区间所代表的字符串。绿点是 $s_l = s_r$ 的区间，红点与之相反。

问题转化成从起点（也即 $[1, n]$ 所代表的节点）走 $\frac{n+|s|}{2}$ 步走到终点的路径数。

更进一步——转移图的性质

注意，下面说的「路径」是指点集和边集，也即将从起点到终点具体走了哪些点和边表示出来，包括重边和自环；而「链」指的是某条路径对应的点集。

仔细观察这个图，可以发现一些性质：

更进一步——转移图的性质

注意，下面说的「路径」是指点集和边集，也即将从起点到终点具体走了哪些点和边表示出来，包括重边和自环；而「链」指的是某条路径对应的点集。

仔细观察这个图，可以发现一些性质：

1. 如果不算自环，这是一个 DAG。

更进一步——转移图的性质

注意，下面说的「路径」是指点集和边集，也即将从起点到终点具体走了哪些点和边表示出来，包括重边和自环；而「链」指的是某条路径对应的点集。

仔细观察这个图，可以发现一些性质：

1. 如果不算自环，这是一个 DAG。
2. 任意从起点到终点的**链**，终点的上一个点一定是绿点；且如果有 i 个红点，那么将会有 $\lceil\frac{|s|-i}{2}\rceil$ 绿点。

更进一步——转移图的性质

注意，下面说的「路径」是指点集和边集，也即将从起点到终点具体走了哪些点和边表示出来，包括重边和自环；而「链」指的是某条路径对应的点集。

仔细观察这个图，可以发现一些性质：

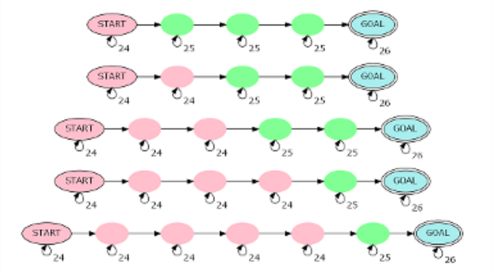
1. 如果不算自环，这是一个 DAG。
2. 任意从起点到终点的**链**，终点的上一个点一定是绿点；且如果有 i 个红点，那么将会有 $\lceil \frac{|s|-i}{2} \rceil$ 绿点。
3. 若两条从起点到终点的**路径**所对应的**链**的红点数量相同，那么其对答案的贡献相同。

更进一步——利用性质

由于只要红点数量确定，绿点数量也会确定，并且点的顺序不影响答案，那么可以将所有的起点到终点的**路径**按照其**链**上红点个数分类，总共只有 $O(n)$ 条本质不同的路径，如图所示。

更进一步——利用性质

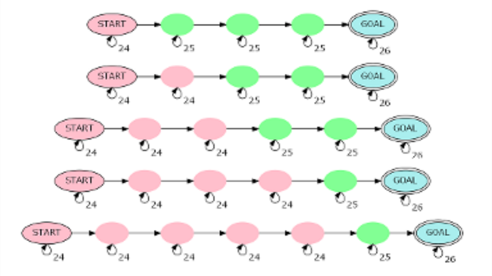
由于只要红点数量确定，绿点数量也会确定，并且点的顺序不影响答案，那么可以将所有的起点到终点的**路径**按照其**链**上红点个数分类，总共只有 $O(n)$ 条本质不同的路径，如图所示。



图里面默认起点是个红点。

更进一步——利用性质

由于只要红点数量确定，绿点数量也会确定，并且点的顺序不影响答案，那么可以将所有的起点到终点的**路径**按照其**链**上红点个数分类，总共只有 $O(n)$ 条本质不同的路径，如图所示。



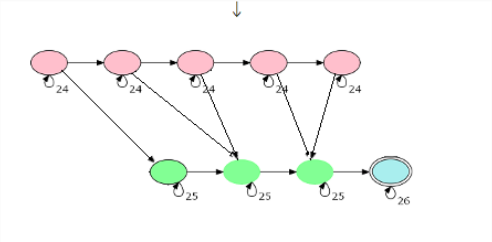
图里面默认起点是个红点。

此时可以考虑求出从起点到终点的有 i 个红点的链的数量（这个可以用记忆化搜索做到 $O(|s|^3)$），然后对所有 i 个红点的链分别做一次矩乘，再乘上 i 个红点的链的数量，可以做到 $O(|s|^4 \log n)$。

正解

现在已经离正解十分接近了。考虑怎么把这些链全部压成一条。

如果把红点和绿点排成两行，第 i 个红点向下面对应的绿点连上权值为 i 个红点的链的数量的边，则可以构造出一次矩乘求出答案的图：

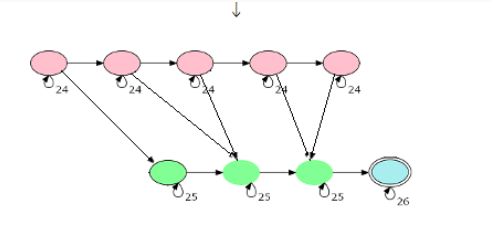


此时做到了 $O(|s|^3 \log n)$，可以通过本题。（实际上会带上一个 $(\frac{3}{2})^3 = \frac{27}{8}$ 的常数，因为这时点数是 $O(\frac{3}{2}|s|)$ 的）

正解

现在已经离正解十分接近了。考虑怎么把这些链全部压成一条。

如果把红点和绿点排成两行，第 i 个红点向下面对应的绿点连上权值为 i 个红点的链的数量的边，则可以构造出一次矩乘求出答案的图：



此时做到了 $O(|s|^3 \log n)$，可以通过本题。（实际上会带上一个 $(\frac{3}{2})^3 = \frac{27}{8}$ 的常数，因为这时点数是 $O(\frac{3}{2}|s|)$ 的）

还遗留了一个问题：$n + |s|$ 是奇数怎么办?

正解——奇数的处理办法

观察转移方程，可以发现，奇数与偶数唯一的不同，就是在**最后一步**形如 $[i, i+1]$ 的绿点不能转移到终点。

考虑计算这些非法的方案数，拿原来算出来的减一下就可以得到真正的答案了。

在转移图上看，也就相当于是求在 $\frac{n+|s|-1}{2}$ 步后到达了某个形如 $[i, i+1]$ 的绿点的方案数。

若把所有这样的路径后面都添加一个终点，就相当于求从起点到终点的方案数，和上面类似建图再矩乘即可。需要注意终点要去掉 26 的自环，并且要多乘一次。

总结

到此，这道题的大致解法已经讲完了。

这道题本质上是在观察转移图得到启发后，利用转移的特殊性质尽可能的压缩了状态量，减少了冗余的转移，从而得到了一个较优的复杂度，是一道不可多得的好题。

实际上这道题有另外的做法，并且可以做到更优的复杂度 $O(|s|^3+\log n)$，这里不再展开，有兴趣的同学可以看一下附件里面的题解。

感谢倾听！

IOI2018 中国国家集训队第一阶段作业
解题报告

广州市第六中学 董炜隽

2017 年 12 月 27 日

1 Division into Two(AGC009_C)

1.1 题目大意

有 N 个互不相同的整数，其中第 i 小的整数为 S_i。问有多少种方案把它们划分成两个集合 X,Y，使得 X 中相邻两个数的差不小于 A，且 Y 中相邻两个数的差不小于 B。

1.2 解法

考虑把序列划分成若干段，之后交替地将每一段放入集合 X,Y 中。

设 f_i 表示已经划分好前 i 个数且 i 为某个放入 X 中的段的结尾的方案数，g_i 表示已经划分好前 i 个数且 i 为某个放入 Y 中的段的结尾的方案数。考虑如何用 f 更新 g（g 更新 f 同理），f_i 可以转移到 $g_j(j > i)$ 的条件为

1. $S_{j+1} - S_i \geq A$
2. $S_{i+1..j}$ 中任意两个相邻的数的差均不小于 B

令 $[l_i, r_i]$ 表示 f_i 可以转移到 g_j 的 j 的范围。显然 l_i 由条件 1 决定，r_i 由条件 2 决定，并且他们都是关于 i 单调非降的，因此可以在 DP 的时候顺带维护一下 l_i, r_i。每次转移时相当于把 $g_{l_i..r_i}$ 都加上 f_i，可以先把 g 差分一下，即把 g_{l_i} 加上 f_i、g_{r_i+1} 减去 f_i，之后用 g 转移时再对其求一下前缀和就能得到实际的值了。

总时间复杂度 $O(N)$。

2 Rotated Palindromes(ARC064_F)

2.1 题目大意

问有多少个长度为 N 且字符集大小为 K 的字符串可以通过回文串旋转（把第一个字符移到最后）若干次得到。

2.2 解法

设 f_i 表示长度为 N，字符集大小为 K，且周期为 i 的回文串的个数。可得

$$f_i = k^{\lceil \frac{i}{2} \rceil} - \sum_{j|i} f_j$$

显然，一个周期为 d 的回文串通过若干次旋转可以得到共 d 个不同的字符串。但是，两个不同的回文串也可能通过旋转得到同一个串，因此我们需要考虑每个串被重复计数了多少次。

对于一个周期为 d 的回文串 x，考虑有哪些与它循环同构的串也是回文的。设 x 旋转 $t(1 \le t < d)$ 次之后能够得到回文串 y，则

x 旋转 $d-t$ 次 $= x$ 逆旋转 t 次 $= x$ 旋转 t 次的反串 $= y$ 的反串 $= y$

由于 x 旋转 $1..d-1$ 次得到的串互不相同，所以有 $t = d - t$，即 $t = \frac{d}{2}$。因此，周期为奇数的串只被计算了一次，而周期为偶数的串会被计算两次，也就是说

$$Ans = \sum_{d|N, d\ is\ odd} f_d \cdot d + \frac{1}{2} \sum_{d|N, d\ is\ even} f_d \cdot d$$

在计算 f 时可以先把 N 分解质因数，然后搜出所有满足 $j \mid i, i \mid N$ 的数对 (i, j) 来计算。令 $d(n)$ 表示 n 的约数个数，则总时间复杂度为 $O(\sum_{i|N} d(i)) \le O(d(N)^2)$。在 $N \le 10^9$ 内 $d(N)$ 的最大值为 $d(735134400) = 1344$，因此该算法可以通过所有数据。

3 Mr.Kitayuta's Gift(Codeforces 506E)

3.1 题目大意

给一个仅包含小写字母的字符串 s。你需要选择恰好 n 个小写字母，并把他们分别插入到 s 的任意位置。问通过这种方式能生成出多少种回文串。

3.2 解法一

暴搜所有方案并计算能得到多少种不同的回文串。

期望得分 10 分。

3.3 解法二

题意相当于问有多少个回文串包含 s 这个子序列。考虑如何判断一个回文串 a 中是否包含子序列 s，我们可以从小到大枚举 i，分别用 a_i 和 a_{n-i+1} 去匹配当前 s 两端的字母，并将 s 中被成功匹配的字符删去。我们可以据此进行 DP，令 $f_{i,l,r}$ 表示枚举到 i 时 $s[l..r]$ 未被匹配的方案数，转移时分 $s_l = s_r$ 与 $s_l \neq s_r$ 讨论即可。时间复杂度为 $O(|s|^2 n)$。

期望得分 30 分。

3.4 解法三

解法二中 DP 每一层的转移是相同的，因此可以用矩阵乘法进行优化。时间复杂度为 $O(|s|^6 \log n)$。

期望得分 40 分。

3.5 解法四

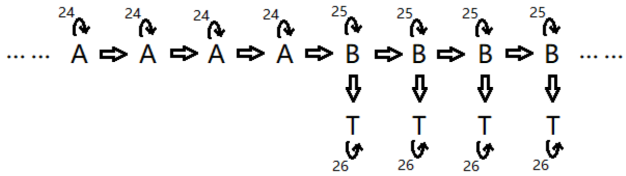
考虑把 DP 转移的过程建成一个自动机，每个节点代表 s 的一个区间 $[l, r]$。我们把 $s_l \neq s_r$ 的节点称为 A 类节点，$s_l = s_r$ 的称为 B 类节点，那么每个 A 类节点有 24 个自环，B 类节点则有 25 个自环。设 a, b 分别为路径上经过的 A 类和 B 类节点的个数。注意到 $b = \lceil \frac{n-a}{2} \rceil$，即合法的数对 (a, b) 仅有 $O(|s|)$ 对。因此我们可以把 (a, b) 相等的所有路径一并计算，先 DP 出经过了 a 个 A 类节点和 b 个 B 类节点的路径数，再通过矩阵快速幂求出把剩下的自环插入到这样的路径中的方案数。由于我们要对每对 (a, b) 做一次矩阵快速幂，因此总时间复杂度为 $O(|s|^4 \log n)$。

注意，当 $|s|+n$ 为奇数时回文串的中心仅有一个字符，需要特殊处理。我们可以分加入中心字符前是否已经匹配完 s 来讨论，两种情况均可以使用之前所说的做法解决。

期望得分 60 分。

3.6 解法五

考虑继续优化解法四。我们可以像这样把所有 (a,b) 的自动机组合在一起：



可以发现，上一个解法中每个需要求的方案数都对应这个自动机中某个 A 类节点走到某个终止节点的长度为 k 的路径数。因此我们只需要对这个自动机做一遍矩阵快速幂就可以得到所有需要求的值。总时间复杂度为 $O(|s|^3\log n)$。另外可以把矩阵写成上三角的形式，这样能把常数将至约 $\frac{1}{6}$。

期望得分 80 分。

3.7 解法六

使用 Cayley-Hamilton 定理优化解法五的矩阵乘法，时间复杂度为 $O(|s|^3+|s|^2\log n)$，期望得分 100 分。

3.8 解法七

设 $m=\lceil\frac{|s|+n}{2}\rceil-a-b$，那么把自环插入到经过 a 个 A 类节点和 b 个 B 类节点的路径上的方案数为

$$\sum_{i=0}^{m}\sum_{j=0}^{m-i}\binom{i+a-1}{a-1}\binom{j+b-1}{b-1}24^i25^j26^{m-i-j}$$

引理 设 $S_{x,y}(n)=\sum_{i=0}^{n}i^xy^i(y\neq 1)$，那么 $S_{x,y}(n)$ 可以表示为 $P_x(n)+w$ 的形式。（下文中所有 P_k 均表示某个 k 次多项式，w 均表示某个常数）

证明

$$
\begin{aligned}
S_{x,y}(n)+(n+1)^x y^{n+1} &= 0^x+\sum_{i=0}^{n}(i+1)^x y^{i+1}\\
&= 0^x+\sum_{i=0}^{n} y^{i+1}\sum_{j=0}^{x}\binom{x}{j} i^j\\
&= 0^x+\sum_{j=0}^{x} y\binom{x}{j} S_{j,y}(n)
\end{aligned}
$$

那么我们可以得到 $S_{x,y}$ 的递推式

$$
(y-1)S_{x,y}(n)=(n+1)^x y^{n+1}-0^x-\sum_{i=0}^{x-1} y\binom{x}{j} S_{j,b}(n)
$$

观察递推式，可以发现 $S_{x,y}(n)$ 就是由若干个 $(n+1)^x y^{n+1}$ 及 0^x 乘上一些由 x,y 决定的系数后相加得到的，因此必然可以表示成 $P_a(n)+w$ 的形式。

考虑使用上面的引理来化简我们需要求的式子：

$$
\begin{aligned}
&\sum_{i=0}^{m}\sum_{j=0}^{m-i}\binom{i+a-1}{a-1}\binom{j+b-1}{b-1}24^i 25^j 26^{m-i-j}\\
=&\sum_{i=0}^{m}\binom{i+a-1}{a-1}24^i 26^{m-i}\sum_{j=0}^{m-i}\binom{j+b-1}{b-1}(\frac{25}{26})^j\\
=&\sum_{i=0}^{m}\binom{i+a-1}{a-1}24^i 26^{m-i}[P_{b-1}(m-i)(\frac{25}{26})^{m-i}+w]\\
=&\sum_{i=0}^{m}\binom{i+a-1}{a-1}24^i[P_{b-1}(m-i)25^{m-i}+w26^{m-i}]\\
=&\sum_{i=0}^{m}25^m P_{a+b-2}(m,i)(\frac{24}{25})^i+w26^m\binom{i+a-1}{a-1}(\frac{24}{26})^i\\
=&25^m[P_{a+b-2}(m)(\frac{24}{25})^m+P_{b-1}(m)]+w26^m[P_{a-1}(m)(\frac{24}{26})^m+w]\\
=&P_{a+b-2}(m)24^m+P_{b-1}(m)25^m+w26^m
\end{aligned}
$$

这里的形式看起来还不够优美，直觉告诉我们 24^m 前面的多项式应该是 $a-1$ 次的（实际测试也是如此）。这里我想到的解释是，如果在化简的过程中交换 i,j，最终得到的结果中 24^m 和 25^m 前面的多项式的次数会反过来变成 $a-1$ 和 $a+b-2$。而 24^m 和 25^m 应当是无法互相转化的（因为实际上这个结论可以把 24,25 替换成其他任意常数），所以 $24^m,25^m$ 前的多项式应该分别是 $a-1$ 次和 $b-1$ 次的。

由于最终的答案是由若干对 (a,b) 的方案数相加得到的，设 $t=\lceil\frac{|s|+n}{2}\rceil$，答案必然可以表示为 $P_{n-1}(t)24^t+P_{\lceil\frac{n}{2}\rceil}(t)25^t+w26^n$ 的形式。我们可以暴力求出 n 较小时的答案，再通过高斯消元解方程组求出式子中的各项系数。总时间复杂度为 $O(|s|^3+\log n)$。

期望得分 100 分。

事实上原题的出题人在题解中只是粗略提到存在不使用矩乘的做法，我不太确定他的做法与我的是否相同。如果有同学会其他做法的话欢迎与我交流。