Problem A

20联赛前冲刺：http://zhengruioi.com/contest/745/problem/1670

长度为n初始全0的序列，两人轮流在上面选择连续的一段长度不超过k的0变成一段1，如无法选择则输，问先手是否必胜

$1 \leq k \leq n \leq 1e9$

$hint$:这是problem A

如果先手能从中间选择一段把序列分成两段相等长度的，则只需要跟对手镜像走一定可以获胜；$k \geq 2$一定可以，否则只有奇数可以

Problem B

20联赛前冲刺：http://zhengruioi.com/contest/716/problem/1593

给定两个字符串s, t，取s的一个前缀接上t的一个后缀构成新字符串(可以为空串)，求有多少种不同的新字符串

$1 \leq |S|, |T| \leq 1e7$

假设已经选定$s[1,i]$作为前缀，那么$t[j,n]$以前不能有$s[1,i]$的后缀，而只要限定$t[j-1]!=s[i]$就一定满足条件，否则这一组贡献一定被计算过

Problem C

20联赛前冲刺：http://zhengruioi.com/contest/719/problem/1612

给定一字符串s，求最多可以选出多少子串(连续)使得他们彼此不是后缀

$|S| \leq 1e6$

以i为最后一个点最多只能贡献1，所以我们只要看它能不能贡献，考虑如果$j < i$且以两者为终点无论怎么选都会冲突，一定有$s[1, j]$是$s[1, i]$的后缀，而且为了后面考虑我们舍弃j保留i；否则一定存在一种起点的选法使两串不冲突；所以我们KMP的时候每次把$nxt[i]$的位置标成$false$，最后统计没有被标的位置的个数就是答案

Problem D

20秋季提高十连测：http://zhengruioi.com/contest/696/problem/1491

T组数据，每次给定一串01串s，每次操作可以将连续的 长度为k的颜色相同串颜色反转；求经过若干次操作后a能否变成b

$T \leq 1e3;\ |s|, k \leq 1e6$

在给定某些操作 求字符串a是否可以转换成b时 当操作可逆时 一种思路是尝试把a,b向另外一个状态(比如最小字典序)转移

本题中注意到$k\times'0'+'1'$可以变成$'1'+k\times'0'$，$k\times'1'+'0'$可以变成$'0'+k\times'1'$，这意味着k个连续的$'0'$或$'1'$是可以自由移动的，且移动后其他字符的相对位置不变；所以我们求最大字典序时可以把u个$k\times'0'$扔到后面，把v个$k\times'1'$扔到前面去，记录下a,b的u,v个数，以及卡在中间无法转移的字符串，都一样才可以从a变成b，否则不行

Problem E

20联赛前冲刺：http://zhengruioi.com/contest/717/problem/1598

给定一$a[1,n]$排列，求$p[1,n-1]$的操作排列数量，每个操作表示交换原排列中的$a_{p_i},a_{p_{i+1}}$，使得操作后原排列升序

$n \leq 5e3$

观察到如果$a[i] == i$一定无解，如果$a[i] < i$说明要一路移动到更左边，所以这一路上的点一定是先做(相对它左边相邻的点)，两端点后做；否则一路上的点一定是后做的，两端点先做；这样就转化为了求排列数量满足$b_{i-1} < b_i$或$b_{i-1} > b_i$，$dp[i][j]$表示考虑了前i个数，最后一个数排名(从小到大)为j时的方案数，当要求$b_{i-1} < b_i$时$dp[i][j] = \sum_{k=1}^{j-1} dp[i-1][k]$，要求$b_{i-1} > b_i$时$dp[i][j] = \sum_{k=j}^{i-1} dp[i-1][k]$

Problem F

20联赛前冲刺：http://zhengruioi.com/contest/725/problem/1631

将给定数n划分成若干个$\in [x,y]$的数的和，求方案数；只是顺序不同的划分算一种，答案对P取模

$1 \le x \le y \le n \le 1e5;\ 1 \le P \le 1e9$

有两种划分数dp：$f[i][j]$表示当前划分的最大数字是j，和为i的方案数，这是个完全背包$f[i]+=f[i-j]$；$g[i][j]$表示将i分成j个数的方案数$g[i][j]=g[i][j-1]+g[i-j][j]$，可以将方案分成两种：包含0的用$g[i][j-1]$转移，不含0的一定唯一对应一种所有数都$-1$的划分方案即$g[i-j][j]$，用$f$转移划分出来的数$<\sqrt{n}$的方案，用$g$转移划分出来的数$\geq\sqrt{n}$的方案，后者最多划出$\sqrt{n}$个数，故总复杂度是$O(n\sqrt{n})$的

至于$\in[x,y]$的限制，可以把他们拆成两个$\geq x\&\geq y$的限制，分别求解后差分即可

Problem G

20联赛前冲刺：http://zhengruioi.com/contest/754/problem/1674

给定一字符串s，求有多少长度为6的子序列满足相等/不相等关系形如114514

$|s| \leq 5e5$

枚举5的位置，我们需要知道前缀114的数量和后缀14的数量，通过dp

$dp0[i][x]$表示$[1,i]$中x的个数，$dp0[i][x]=dp0[i-1][x]+[a[i]==x]$

$dp1[i][x]$表示$[1,i]$中xx的个数，
$dp1[i][x]=dp1[i-1][x]+[a[i]==x]\times dp0[i-1][x]$

$dp2[i][x][y]$表示$[1,i]$中xxy的个数，
$dp2[i][x][y]=dp2[i-1][x][y]+[a[i]==y]\times dp1[i-1][x]$

$dp3[i][x][y]$表示$[i,n]$中xy的个数，
$dp3[i][x][y]=dp3[i+1][x][y]+[a[i]==x]\times(dp0[n][y]-dp0[i][y])$

$$ans=\sum_{x!=a[i]}\sum_{y!=a[i]\ \&\ x!=y}dp2[i-1][x][y]\times dp3[i+1][x][y]$$

卡了空间，不能开这么多数组，先把$i==1$时$dp3[x][y]$算出来，这样向后推的时候四个式子都能动态维护省掉第一维

Problem H

20联赛前冲刺：http://zhengruioi.com/contest/730/problem/1642

给定一个初始的01串s，一次操作把原来所有相邻的前0后1进行交换构成新串，求t次操作后的字符串串，以及每一次操作后的逆序对数量

$|s|, t \leq 5e6$

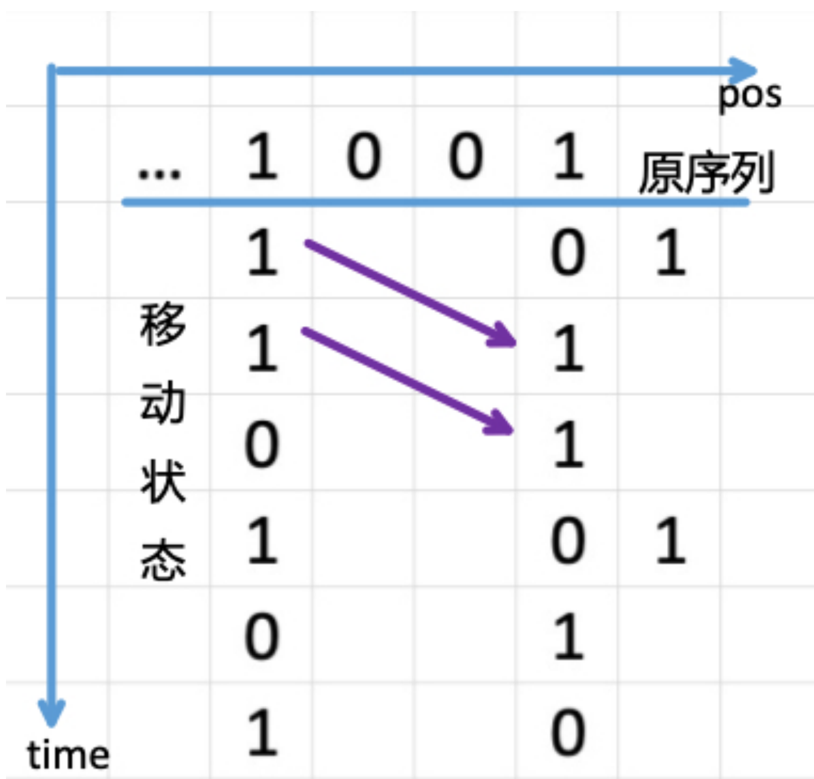
对每一个字符1求其在第i次操作中能否向前交换，能1不能0；

如果当前字符的前一位也是1，那么状态就会比前一位整体慢一步，并且第一步也是0；否则如果和上一个1之间隔了k个0，可以先认为两个1是相邻的计算移动状态，每次当前的1被卡位了就无中生有一个0让它能继续移动，所以就将状态前k个0变成1；

维护状态可以通过deque维护移动状态中所有0的位置并打全局tag；

第i位字符1的移动状态中有j个0就意味着阻止了t次移动中的j次，所以这一位的1最终的位置在$i-t+j$；

每一个1每移动一次就会增加一个逆序对个数，所以操作一次的逆序对变化量就是当前操作时间上所有字符1的移动数量，而某一位字符的移动状态中的一个0变成1，会使其之后的所有字符1的移动状态上慢1,2,3...位也从0变成1；所以记录当前操作位是否变化是移动数量的差分值，求和之后就是逆序对的变化量了



Problem I

20秋季提高十连测：http://zhengruioi.com/contest/684/problem/1533

给出一字符串s，求形如AA^*A的子串(连续)的数量，其中A^*是A的倒序

$|s| \le 2e6$

$hint: manacher$

首先可以通过manacher O(n)算出以每个点为中心的最长回文串AA^*半径$len[i]$

考虑枚举所求字串前1/3位置i，则能对该点产生1贡献的后1/3位置x满足$x \in [i+1, i+len[i]] \&\& x-len[x] \le i$，这样倒着做的时候后一个条件右侧单调减，可以用优先队列边走边删除/加入，对前一个条件用树状数组查询一下即可

Problem J

20联赛前冲刺：http://zhengruioi.com/contest/725/problem/1632

给定一字符串s，将其分为五部分$A+B+C+D+E=S$，可能有空串，并要求$A+C+E$为回文串，求$|A|+|C|+|E|$的最大值

$|s| \le 5e6$

首先可以将首位相同的的字母逐个删去，这样问题就变成了$A+B+C+D$中最大化回文串$A+C$或$B+C+D+E$中最大化回文串$C+E$，第二个问题反过来和第一个等价，这里只讨论第一个

假设$A+C$的中心点在A中，枚举这个中心点i，通过manacher求出其回文半径为t，左边界为$l=i-t$，右边界为$r=i+t$，问题变成了r右边是否存在后缀是$s[l-1,1]$，考虑如果问的是后缀是$[1,l-1]$那就直接拿KMP绝杀，这里我们可以把文本串倒过来，用s^*作为文本串和s做匹配，其中s^*是s的倒序，记$f[i]$为$s^*[i]$的匹配数，$g[i]$为$s^*[i]$的前缀最大匹配数，则$[g[n-r]>l-1]$即为所求

假设$A+C$的中心点在C中，则问题变成了$k-r$使得$s[k,r]=s[1,k-r+1]$，发现这个东西就是上述$f[r]$

实际做的时候要倒过来，先预处理manacher，KMP，再枚举中心点，同时讨论奇偶性

Problem K

20联赛前冲刺：http://zhengruioi.com/contest/757/problem/1684

定义两字符串是相似的，当且仅当它们的最长公共子序列的长度$\geq n-2$；求在字符集大小为R的情况下，有序对(A,B)的个数，其中$|A|=|B|=n$，且A,B是相似的

$n \leq 1e18;\ 2 \leq R \leq 1e9$

LCS的转移是
$dp[i][j] = max\{dp[i-1][j], dp[i][j-1], [a[i] == b[j]] \times (dp[i-1][j-1]+1)\}$
，既然要求LCS的长度$\geq n-2$，说明每次最多从$i-2, j-2$的地方转移过来

考虑每次加入一个字符，从第i位转移到第$i+1$位时需要的状态有：
$dp[i-2][i], dp[i-1][i], dp[i][i], dp[i][i-1], dp[i][i-2], A[i-1], A[i], B[i-1], B[i]$
；考虑这9个数的本质，其中dp值变成$i - dp[*][*] \leq 2$，字符只需要用$1, 2, 3, 4$记录相等关系

dfs一下，每次枚举$A[i+1], B[i+1]$与$A[i-1], A[i], B[i-1], B[i]$之间的相等关系，便可以找到所有的状态及他们之间的转移$dp[i+1][S'] = dp[i][S] \times A[S'][S]$
，发现可能的状态只有123种

设初始矩阵$F[i][1]$表示初始状态为i的方案数，单位矩阵$A[i][j]$表示一步从状态j转移到状态i的方案数，这样每多一个字符$F = A \times F$，矩阵快速幂一下即可