1001

用set记录一下每个点当前的最小花费是由哪几种情况转移而来 跑一边最短路即可。

1002

AraBellaC

把 ‘A'字符的出现位置集合记录为 a_i ,类似的记录'B', 'C'字符的出现位置 `ParseError: KaTeX parse error: $ within math mode` 。如果假定循环节长度为 len ， 那么 a,b,c 有解当且仅当任意 i,j,k 对如下公式成立

$$\begin{cases} max\{(a_i-1)\ mod\ len\} < min\{(b_j-1)\ mod\ len\} \\ max\{(b_j-1)\ mod\ len\} < min\{(c_k-1)\ mod\ len\} \end{cases}$$

如果此方程满足了，则可以更新 a,b,c 的最优解为
$max\{(a_i-1)\ mod\ len\}, max\{(b_j-1)\ mod\ len\} - max\{(a_i-1)\ mod\ len\}, len - max\{(b_j-1)\ mod\ len\}$ 。

但是如果直接暴力，复杂度将会达到 $T*N*M$,其中 N,M 分别表示已知的点的个数与最大循环节长度。

考虑进行优化： 枚举循环节长度，并在当前循环节长度下枚举每个循环节，二分查找这个循环节内的
$max\{(a_i-1)\ mod\ len\}, max\{(b_j-1)\ mod\ len\}, max\{(c_k-1)\ mod\ len\}$ ， 复杂度为 $T*M*log(M)*log(N)$

可以通过 ST 表优化到 $T*m*log(m)$ 。

1003

YJJ’s Stack题解

一种可行解法：

先考虑没有 `pop` 的情况，因为 v 值最多五个，我们完全可以针对每一个的 v 值维护一个栈，每次查询的时候将每个栈的栈顶元素的插入时间 t 进行比较， t 更大的那个就是总栈的栈顶。

所以只用考虑维护一个栈的时候，我们把 t 离散化。当 `query T` 时，我们需要找到一段区间最短的区间 $[x,T]$ ，使得 $[x,T]$ 这段区间里 `push` 和 `delete` 操作抵消后，只剩一个 `push` 的元素，这个元素的插入时间就是 `x`。

找到 x 值的方式多种多样，其中一种考虑权值线段树， `push` 操作在线段树 t 上+1， `delete` 操作-1，所以只用找到最短的区间，使得 $sum[x,T]=1$ 即可。具体可以用区间修改，令 val[i] 表示 $[i,N]$,即以i为后缀的区间的和，所以 $val[x]-val[T]$ 即可表示 $[x,T-1]$ 区间的和，要找到栈顶，等价于找到最小的 $x<T$,使得 $val[x]>val[T]$ 即可，改一改线段树的查找写法，优先处理右区间即可。

我们把没有 `pop` 时的 `query` 操作命名为 `query0` ，再考虑有 `pop` 时，因为 `pop` 数量只有 5，每次 `query` 时，需把 `pop` 替换成相应等价的 `delete v` ，所以就需要在 `pop` 的地方进行子操作 `query0` ,获得 `pop` 对应删除的是哪个 v，替换成 `delete` 后再在 T 处进行一个 `query0` 操作即可。

总复杂 $O(c(k+1)n\log(n))$ ，c 是 v 的值域， k 是 `pop` 的数量。

1004

引理：n个在0到A上均匀分布的随机变量X1，X2，X3....X1，X2，Xn，若互相独立，则随机变量Z=min(X1，X2，X3...Xn)的期望E（Z）=A/（n+1）

该引理可较简单由观察发现或通过概率论证明，证明思路：求分布函数相乘再还原概率密度函数，由期望定义求得。

核心思想：由于来车的概率刚好是L/M ，所以每辆车都能看作一个在0到M上均匀分布的随机变量Xi，如果Xi<=Li表示车来了的，Xi>Li表示车没来。

首先把所有Li增序排序。对于策略点时间点X 总期望的贡献由三部分构成：

Li<X的部分

先依次枚举最小的Li是哪个。

比如枚举到第i个，所有j>i的车对Li来说都是等价的（因为i一定会来，等车时间一定<Li，而Lj>Li）此时再枚举标号比i大的车有几辆来的时间早于Li，若来了k辆，由引理此时期望为Li/（K+2)，来K辆车的方案有C（n-i，k）种，概率为 $P_i^k * (1-P_i)^{n-i-k}$.因此总贡献为P（比i小的车都没来）×P（i一定来）× $\sum_1^{n-i}(C(n-i,k)*Pi^k*(1-Pi)^{(n-i-k)}*(\frac{Li}{(K+2)}+A))$ 把每个i的贡献都加进答案即可

Li>X的部分

如果Li>X,其实对X来说都是等价的（核心思想）。 应此枚举有几辆车来的时间早于X，接下来和第一部分相似处理。 这部分期望不要忘了乘上P（Li<X的车都没来）。

一辆车都没来的情况

简单出统计概率乘（X+B）加到答案里即可

第一部分其实和X的具体值无关，只与有几个Li<X有关，可以预处理。 第二部分接近O（N) 第二部O（1） 总复杂度O（N×Q）

1005

题意:给定 n , m , p ,求

$$(\sum_{a=1}^{m}\sum_{b=1}^{n}\frac{\phi(ab)}{\phi(a)\phi(b)}) \pmod p$$

解法:通过观察,容易得到

$$\frac{\phi(ab)}{\phi(a)\phi(b)}=\frac{\gcd(a,b)}{\phi(\gcd(a,b))}$$

故原式等价于

$$\sum_{a=1}^{n}\sum_{b=1}^{m}\sum_{k=1}^{\min(n,m)}[k==\gcd(a,b)]\frac{k}{\phi(k)}=\sum_{k=1}^{\min(n,m)}\frac{k}{\phi(k)}\sum_{a=1}^{n}\sum_{b=1}^{m}[k==\gcd(a,b)]$$

右式等价于统计 $a=1-n$ 与 $b=1-m$ 中最大公因数为 k 的个数,解法很多(如Mobius).

题目保证质数比 n 与 m 大,故直接求逆元最后乘起来即可.

标程复杂度: $O(n\log n+Tn\log n)$

1006

根据异或的性质, $X_0\oplus X_1\oplus X_2\oplus X_3\oplus .....\oplus X(M-1)=S\oplus T$ 计算出S⊕T中一共有cnt个1，将这个二进制数标准化,使它成为前面N-cnt个0，后cnt个1的二进制数，答案保持不变，问题变为:用K个N位且有3个1的不同二进制数进行异或，最终得到前面N-cnt个0，后面cnt个1这个二进制数，有多少种方案？ 递推。设d[K][M]表示用K个N位且有3个1的不同二进制数进行异或，最终得到前面N-cnt个0，后面cnt个1这个二进制数的方案数。 递推方程：

$$d[i][j]=d[i-1][j+1]*C[j][1]*C[n-j][2]+d[i-1][j+3]*C[n-j][3]+d[i-1][j-1]*C[j][2]*C[n-j][1]$$

此时得到的d[i][j]未去除加入数字的先后顺序和加入重复串带来的影响 考虑加入重复串的影响，$d[i][j]-=(d[i-2][j]*(C[n][3]-i+2)$; 表示去除加入i-2个数字并且后面有j个1这个二进制数但后来又异或上两个之前没用过的相同的数的方案数， 考虑二进制数加入的先后顺序给答案带来的额外贡献,d[i][j]=d[i][j]*inv[i]%mod 即d[i][j]/i，其中inv[i]为数字i关于模mod的逆元.

1007

题意：给定一个 N*N 的染有黑白的矩阵，矩阵的左边和右边相连（可以想象成一个圆 柱面）。每次操作可以是将一个格子的颜色翻转，也可以是将一列的格子的颜色全部翻转。 对于每次操作后回答整个矩阵有多少个黑连通块和白连通块。 题解：横向建立一个线段树，线段树的每个叶子节点表示一列格子的颜色状态。之后对 于线段树上的每个节点，维护一个并查集表示对于该节点所表示的区间，其左右边上的节点 哪些属于同一个块，合并的时候暴力向上合并。最后维护下线段树的根节点左右边上的并查 集，处理下环的问题即可。 时间复杂度 O（QNlogN）

1008

题意：给定一棵边带权的树（加一条额外的边），支持两种操作：修改某条边的权值，询问两点间的最短路。

把边分成两种分别处理：对于环上的边，用一个单点修改，前缀求和的树状数组维护；对于不在环上的边，把环上任意一条边删去，并且将环上的任意一个点当做根，就变成了一棵普通的树，求两点间距离可以结合 LCA ,用树状数组的区间修改，点查询来完成。询问的时候，如果路径通过环，就把两部分答案结合，否则只需直接使用树上两点距离即可。复杂度 $O(nlogn)$ 。

1009

首先树分块，我们维护这几个东西 $a[i]$ 就是题目中的 $a[i]$ $cnt[i]$ 当前节点跳多少步才能到下个块 $b[i]$ 当前节点跳 $cnt[i]$ 步后到哪个节点，就是下一块的第一个节点是什么 修改的时候，暴力扫自己块里面所有节点，如果这个节点要经过 x 的话就直接修改 cnt 然后修改 $cnt[x]$ 和 $a[x]$ ，就维护完了所有东西 当然这道题也可以LCT，不过常数较大，实际运算速度与分块差别不大。

1010

$\lfloor\frac{P}{i}\rfloor$ 不同的取值在 $\sqrt{P}$ 级别，然后按此分段用矩乘快速幂递推。注意 P 和 n 的大小。

1011

解法 1

对于 k 种防御属性，对于每一种属性建 1 个以该属性为关键字的堆，一开始将所有 monster 放入第一个堆。依次检查每一个堆，对于第 i 个堆的堆顶 monster，若其第 i 个属性满足被杀死的条件，则将其弹出并放入第 i+1 个堆，循环往复直至无法移动 monster。每个怪兽最多进堆 k 次，出堆 k 次，时间复杂度 O(kn log n)

解法 2

对于 k 种防御属性，分开进行从小到大排序，设立 k 个指针从最小处开始往最大处移动，对满足被杀死的条件的属性进行标记，当某只 monster 的所有防御属性都被标记时，更新剑士的魔法属性同时更新指针往后移动。时间复杂度 O(kn log n)