7-25 多校题解

咕咕咕，咕咕咕

Absolute

做法一：设 $f(i,x)$ 表示|前i个变量的和+x|的期望，显然这是一个分段多项式函数，每一段可以通过对 $f(i-1)$ 积分得到。$f(i)$ 的段点为 $f(i-1)$ 的段点减去 l_i 和减去 r_i。时间复杂度 $O(\sum_{i=1}^{n}(2^i*i^2))=O(2^n*n^2)$。

做法二：$f(i,x)$ 是对 $f(i-1,x+l_i)$ 到 $f(i-1,x+r_i)$ 进行积分，那么相当于不定积分后两个位置相减。那么 $f(n,0)$ 就是将 $f(0,x)$ 不定积分 n 次之后，在 2^n 个位置求值。（类似容斥，令 $x=\sum_{i=1}^{n}(l_i\ or\ r_i)$，系数为 $(-1)^{l_i\text{个数}}$），时间复杂度 $O(2^n*\log(n))$。

Counting Permutations

设 $g_n=\max_{\{p_1,p_2...p_n\}=\{1,2,...,n\}}\sum_{i=1}^{n}\min(i-l_i,r_i-i)$

我们枚举 $s+1$ 左边有 i 个数，$g_{s+1}=\max_{i=0}^{s}(g_i+g_{s-i}+\min(i,s-i))$。

可以证明 $g_n=O(n\log(n))$。

设 $h_n=\sum_{\{p_1,p_2...p_n\}=\{1,2,...,n\}}x^{\sum_{i=1}^{n}min(i-l_i,r_i-i)}$。

那么容易看出 $h_0=1$，$h_{s+1}=\sum_{i=0}^{s}x^{\min(i,s-i)}h_ih_{s-i}C_s^i$。

暴力算这个好像能过，不过存在更好的做法。

由于 h_x 是一个 g_x 次的多项式，那么我们可以带入 $x=0...g_n$ 算出 $h_0...h_n$ 在每个 x 时的值，那么我们就可以用多项式插值还原出 h_n。

复杂度 $O(n^3\log(n)+Tn^2)$，也可以做到更低。

Cover

每个连通块显然是独立的。对于一个连通块（除了单个点的），如果奇度数点个数为 k，那么至少需要 $\max(k/2,1)$ 条路径。我们将奇度数点两两配对连边，求出欧拉回路，然后把这些边删掉，就可以变成恰好 $\max(k/2,1)$ 条路径。

复杂度 $O(n+m)$。

Game

考虑将游戏变成初始时只有2~n，如果先手必胜的话，那么先手第一步按这样取就获胜了；如果后手必胜的话，那么先手第一步取走1就获胜了。所以全输出Yes就行了。

时间复杂度 $O(1)$。

Hack It

取质数 p 使得 $n=p^2$，考虑构造 p^2 个有比较多1的01序列使得没有任意两个有超过一个公共1。

对于每个 $k,b\in[0,p)$ 取 $ip+(ki+b) \bmod p\ (i\in[0,p))$ 为1即可。

Matrix

考虑对行的条件进行容斥，那么 $ans=\sum_{j=a}^{n} f_{a,j}C_n^j\times$选出的 j 行全是黑格，有至少 b 列是黑格的方案数。f 是某个神秘容斥系数。

考虑怎么求 f，注意到我们如果枚举 $j=a\dots n$，那么恰好 j 行为黑格，至少 b 列为黑格的方案数就被算了 $\sum_{k=a}^{j-1} C_j^k$ 次，那么 $f_{a,j}=1-\sum_{k=a}^{j-1} C_j^k f_{a,k}$。

这个容斥系数还能再在列上用一次，那么 $ans=\sum_{j=a}^{n} f_{a,j}C_n^j \sum_{k=b}^{m} f_{b,k}C_m^k\times$ 选出的j行k列是黑格的方案数，这个显然是 $2^{(n-j)(m-k)}$。

预处理一下2的次幂就行了。复杂度 $O(n^2+m^2+nm)$。

Naive Operations

本来的std挺傻吊的，大家就当无事发生过。

比较靠谱的做法是这样的，考虑维护 $\lfloor (a_i+t_i)/b_i\rfloor > \lfloor a_i/b_i\rfloor$ 的这样的最小的 t_i，每次 a_i 加一的时候 t_i 就减一，一旦 t_i 变成 0 了那么就需要把 $\lfloor a_i/b_i\rfloor$ 加一，这样两个线段树维护一下就行了。

注意到 $\sum_{i=1}^{n}\lfloor n/b_i\rfloor$ 由于 b 是排列是 $O(n\log(n))$ 的，那么复杂度就是 $O(n\log^2(n))$ 。

Odd Shops

我们需要算 $(1+\sum_{i=1}^{10} a_ix^i)^n$ 有多少项系数为奇数。

考虑一个更广的问题：给 $f(x),g(x)$，求 $f(x)^ng(x)$ 有多少项系数为奇数。

注意到 $(\sum_{i=0}^{\infty} a_ix^i)^2\equiv\sum_{i=0}^{\infty} a_ix^{2i}(mod\ 2)$（freshman's dream）。

考虑 $f(x)^{2k}g(x)$，我们把 $g(x)$ 划分成奇次项和偶次项：

$h(x)=\sum_{i=0}^{\infty} a_ix^{2i}, r(x)=\sum_{i=0}^{\infty} a_ix^{2i+1}, g(x)=h(x)+r(x)$。

注意到 $f(x)^{2k}=(f(x)^k)^2$，那么 $(f(x)^k)^2 \bmod 2$ 只有偶次项系数非零，那么 $f(x)^{2k}h(x)$ 和 $f(x)^{2k}r(x)$ 彼此无关。

注意到 $f(x)^{2k}h(x)=(f(x)^kh'(x))^2$、$f(x)^{2k}r(x)=(f(x)^kr'(x))^2x$ （反用freshman's fream）, 我们就可以递归算 $f(x)^kh'(x)$ 和 $f(x)^kr'(x)$ 奇次项的个数。

注意到 g 的度数不超过 10，那么总共的合法状态数不超过 $2^{11}\log(n)$，用一个map就行了。$O(2^{11}\log^2(n))$。

Segment

首先我们考虑一个 $O(n^2+q)$ 的做法。

假设树已经确定了，那么对于一个询问，ans=相交数量-父亲被包含数量。那么对于相交且不被包含的区间，贡献为1；对于被包含的非叶区间（即l<r的区间），贡献为-1；对于被包含的叶子区间，贡献为1。

dp求出每个区间出现的概率，然后可以轻松地预处理出询问每个区间的答案。

接下来我们发现，随机出这棵树的过程等价于：随机一个1~n-1的排列，每次需要mid时取走排列里最靠前的一个合法的。将这个过程倒过来就是，一开始有n个叶子结点（也就是单点区间），每次将排列里最后一个数左右两侧的区间合并起来。

那么一个区间[l,r]出现当且仅当排列中l-1和r+1在l~r之前，那么所有长度相同的区间出现的概率是一样的（除了l=1和r=n的）。那么只需要几次前缀和就可以求出各种区间的贡献了。

时间复杂度 $O(n)$。

Swaps and Inversions

注意到逆序对=交换相邻需要交换的次数，那么输出 $\min(x,y)\times$ 逆序对个数 即可。