中超第二场题解

1001 I love cube

容易发现这题是签到题

发现所有可能满足条件的等边三角形都是形如$(0,0,0)$,$(1,1,0)$,$(0,1,1)$的

考虑对于每个边长分别计算

对于边长为i的，答案为$8*(n-i)^3$

所以最后的答案为$\sum_{i=1}^{n-1} 8*(n-i)^3 = 8*\sum_{i=1}^{n-1} i^3 = 2*(n(n-1))^2$

注意n要先对mod取模或者使用$int128$

1002 I love tree

因为还有一道线段树题所以这题维护的线段树标记比较简单

方法1

考虑对于一条链$< a,b >$树剖之后对应到log段连续的区间

于是问题可以转化为给区间加二次函数

不妨设对$[h,t]$这一段区间增加$(x-h)^2$

对$(x-h)^2$展开之后是$x^2 + h^2 - 2*x*h$

容易发现对三个标记分开维护就可以了

时间复杂度 O($nlog^2 n$)

方法2

考虑对于操作分块重建

每次询问相当于查询重建之前的修改和未被更新的修改的和

对于未被修改的和，每次需要判断一个点是否在一条链上

这个我们可以利用$rmq-lca\ O(1)$完成

对于重建的操作，我们考虑利用一下差分

这样子操作就变成区间加等差数列 我们可以O(块大小)修改 $O(n)$维护区间和

于是我们将块大小取根号

总时间复杂度O($n\sqrt{n}$)

1003 I love playing games

在验题的时候这道题有比较多错误的$O(n)$做法，所以构造了比较多分层图的数据来卡掉这些做法

我们可以先对z点做一遍最短路

如果$dis[x] < dis[y]$ 先手必胜

如果$dis[x] > dis[y]$ 后手必胜

如果$dis[x]=dis[y]$ 我们利用dp来判断一下

令$dp[xx][yy][0/1]$代表当前x在xx点,y在yy点,当前轮到谁走,dp值代表必胜/平局/必输

转移枚举最短路图的边暴力转移即可

时间复杂度$O(n^2)$

1004 I love counting

这道题和上一场的1006和1010的$idea$有点撞。。。~~就当是上一场的复习题好了~~。

做法有一大堆，std写的是对于所有数建01$trie$， $trie$上的每个节点记录一下子树内的所有点，对于一个询问，它所包含的是$trie$上的最多log个子树，将它们挂在这些子树上，最后对每颗子树做一遍二维数点即可(方法如同上一场，区间数颜色转化为前驱<询问左端点，查询点在区间内)。

1005 I love string

容易发现，假设我要模拟出字典序最小的序列，只有最前面一段相同的字符才可以有两种选择，令最初一段相同的字符长度为x，答案就是2^{x-1}。

1006 I love sequences

因为$\sum_{p=1}^{n} n/p = nlogn$

所以对于每一个p分别计算卷积即可

FWT的经典套路是考虑构造矩阵使$(T1A)*(T2B)=T3C$

将其按照最高位拆开之后，即$A=[A0,A1,A2],B=[B0,B1,B2],C=[C0,C1,C2]$

即要求$T1A(x)*T2B(x)=T3C(x)$

按照这题的要求 $C[0]=A[0]*B[0]$ $C[1]=(A[0]+A[1]+A[2])*(B[0]+B[1]+B[2])-C[2]$ $C[2]=(A[0]+A[2])*(B[0]+B[2])-C[0]$

代码比较容易实现

时间复杂度$nlog^2n$

1007 I love data structure

~~题面简单易懂~~。

二三操作可以用矩阵$\begin{bmatrix} 3 & 3 \\ 2 & -2 \end{bmatrix}$和$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$来表示。

在线段树的每个节点上维护矩阵乘法标记和加法标记，在下放乘法标记时候用标记更新加法标记和要求的和，与此同时还需要维护$\sum_{i=1}^{n} a_i, \sum_{i=1}^{n} b_i, \sum_{i=1}^{n} a^2, \sum_{i=1}^{n} b^2, \sum_{i=1}^{n} a_i * b_i$，下放的时候更新即可。

本来要开3s的，结果不小心开成了5s，维护6*6矩阵的做法也放过去了。

1008 I love exam

入门dp。

对于每门课，我们可以背包算出在$f(i)$表示在花费i天的情况下，最多能得多少分。

合并每门课的时候，记录$g(j,k)$表示花费j天，挂了k门课的时候得到的最多分数。

最终转移是$O(n*t^2*p)$的。

1009 I love triples

将a_i中的平方因数消去并离散化，于是问题就是要算有多少(i,j,k)满足$1 \le i \le j \le k \le V(V=100000)$且$ijk$是平方数。

容易发现(i,j,k)一定满足以下两种情况的其中一种：

1.i,j,k均不包含大于$\sqrt{V}$的因数。

2.i,j,k中的某两个包含相同的大于$\sqrt{V}$的因数。

对于第一类，经过搜索会发现这种三元组的数量很少，爆搜即可。

对于第二类，枚举两个含有大于$\sqrt{V}$的因数，然后用hash查找另一个数。因为含有$P(P>\sqrt{V})$的数不超过$\sqrt{V}$个，枚举的总效率是$O(V\sqrt{V})$。

1010 I love permutation

得到的数列显然是一个长度为$P-1$的排列，求排列的逆序对数对2取模的值，相当于求这个排列的奇偶性。

令这个排列为π,排列的第i个数为$\pi(i)$,排列的逆序对数为$n(\pi)$，$sgn(\pi)=(-1)^{n(\pi)}$。

$$sgn(\pi)=\frac{\prod_{0<j<i\le p-1}\pi(i)-\pi(j)}{\prod_{0<j<i\le p-1}i-j}=\prod_{0<j<i\le p-1}\frac{\pi(i)-\pi(j)}{i-j}\equiv\prod_{0<j<i\le p-1}\frac{ia-ja}{i-j}=a^{\frac{P(P-1)}{2}}\equiv a^{\frac{P-1}{2}}(modP)$$

于是直接计算$a^{\frac{P-1}{2}}(modP)$的值，就能知道$sgn(\pi)$的值，进而算出$n(\pi)$的奇偶性。

1011 I love max and multiply

要求$C_k=max(A_iB_j)$并满足$i\&j\ge k$。

可以考虑求出所有的$D_k=max(A_iB_j)$并满足$i\&j=k$，然后再从后往前取max。

然而$i\&j=k$也不太好做，可以改为求$D_k=max(A_iB_j)$并满足$k\in i\&j$的。

对于A和B分别求满足$k\in i,j$时的$max(A_i),min(A_i),max(B_j),min(B_j)$。然后分别相乘取最大的即可得到$D_k$。