水题选讲

dy0607

September 8, 2017

Overview

今天的主题是递推，分治，二(三)分。
不过一般认为递推和DP没有什么区别，所以递推的部分就随便选了几道计数题。

Overview

今天的主题是递推，分治，二(三)分。

不过一般认为递推和DP没有什么区别，所以递推的部分就随便选了几道计数题。

主要是一些NOIP或者稍高于NOIP难度的题。

HAOI2009逆序对数列

求逆序对数为k长度为n的排列个数，对10000取模。

HAOI2009逆序对数列

求逆序对数为k长度为n的排列个数，对10000取模。

$n, k \leq 10^3$

n, k都很小，直接开两维递推。

n, k都很小，直接开两维递推。
从小到大将数加入排列，$f[i][j]$表示前i个数产生了j个逆序对，考虑第$i+1$个数放置位置。

n, k都很小，直接开两维递推。
从小到大将数加入排列，$f[i][j]$表示前i个数产生了j个逆序对，考虑第$i+1$个数放置位置。

$$f[i][j] = f[i-1][j-i+1..j]$$

n, k都很小，直接开两维递推。

从小到大将数加入排列，$f[i][j]$表示前i个数产生了j个逆序对，考虑第$i+1$个数放置位置。

$$f[i][j] = f[i-1][j-i+1...j]$$

前缀和优化一下即可，$O(nk)$。

HNOI2011数学作业

求将1到n的整数串起来得到的数模m的值。

HNOI2011数学作业

求将1到n的整数串起来得到的数模m的值。

$n \leq 10^{18}, m \leq 10^9$

$f[i] = f[i-1] * 10^x + i$
其中x是i的位数。

$f[i] = f[i-1] * 10^x + i$

其中x是i的位数。

于是只要按x分开算，然后就可以上矩阵了。$O(\log^2 n)$

NOI2009管道取珠

两个长度分别为n, m的0/1串，现在可以从某个个串开端取出一个字符，重复$n+m$次可以得到一个长$n+m$的新串。
设有a_i种取出字符的方案可以得到同一个串，则$\sum a_i = \binom{n+m}{n}$，现在你需要求$\sum {a_i}^2$。

NOI2009管道取珠

两个长度分别为n, m的0/1串，现在可以从某个个串开端取出一个字符，重复$n+m$次可以得到一个长$n+m$的新串。

设有a_i种取出字符的方案可以得到同一个串，则$\sum a_i = \binom{n+m}{n}$，现在你需要求$\sum {a_i}^2$。

$n, m \leq 500$

由于是$\sum {a_i}^2$，可以看成取两次得到了相同的串的方案数。

由于是$\sum a_i^2$，可以看成取两次得到了相同的串的方案数。
于是设$f[i][j][k]$表示现在取了i个字符，第一次取时已经取了第一个串的前j个字符，第二次已经取了第一个串的前k个字符，且两次取出来的串相同的方案数。

由于是$\sum a_i{}^2$，可以看成取两次得到了相同的串的方案数。
于是设$f[i][j][k]$表示现在取了i个字符，第一次取时已经取了第一个串的前j个字符，第二次已经取了第一个串的前k个字符，且两次取出来的串相同的方案数。
转移只需要判一下对应位置是否相等，$O(n^3)$。

CF418 Div2 E

给出每个点的度数d_i，求n个点的简单无向图个数，满足$dis(1, i-1) \leq dis(1, i)$，$dis$表示两点之间的最短路（所有边权均为1），且最短路唯一。

CF418 Div2 E

给出每个点的度数d_i，求n个点的简单无向图个数，满足$dis(1, i-1) \le dis(1, i)$，$dis$表示两点之间的最短路（所有边权均为1），且最短路唯一。

$d_i \in \{2, 3\}, n \le 50$

考虑将图按$dis(1, i)$分层，那么每一层中的点都是一段连续的区间。

CF418 Div2 E

考虑将图按$dis(1, i)$分层，那么每一层中的点都是一段连续的区间。

现在将一个点加入，那么我们只需要考虑上一层和这一层已经加入的点还有多少能连一条边和两条边。

CF418 Div2 E

考虑将图按$dis(1,i)$分层，那么每一层中的点都是一段连续的区间。

现在将一个点加入，那么我们只需要考虑上一层和这一层已经加入的点还有多少能连一条边和两条边。

于是记$f[u][i][j][k][l]$为加入了前u个点的方案数，i,j,k,l分别是这一层和上一层点的能连的边的情况。

转移只需要讨论一下这个点的连边情况，注意只有在$i=j=0$时才能转移到下一层，$O(n^5)$。

考虑将图按$dis(1,i)$分层，那么每一层中的点都是一段连续的区间。

现在将一个点加入，那么我们只需要考虑上一层和这一层已经加入的点还有多少能连一条边和两条边。

于是记$f[u][i][j][k][l]$为加入了前u个点的方案数，i,j,k,l分别是这一层和上一层点的能连的边的情况。

转移只需要讨论一下这个点的连边情况，注意只有在$i=j=0$时才能转移到下一层，$O(n^5)$。

$n \le 300$？

CF418 Div2 E

一次考虑一层的贡献。

CF418 Div2 E

一次考虑一层的贡献。

先做预处理，设$f[i][j][k]$表示这一层的个数为i，上一层有j个$d=2$的节点，有k个$d=3$的节点时的方案数。注意这里没有考虑这一层的点与下面的点以及这一层的点之间的连边情况。

CF418 Div2 E

一次考虑一层的贡献。

先做预处理，设$f[i][j][k]$表示这一层的个数为i，上一层有j个$d=2$的节点，有k个$d=3$的节点时的方案数。注意这里没有考虑这一层的点与下面的点以及这一层的点之间的连边情况。

f的转移只需每次考虑一个上层的点的连边情况，可以在$O(n^3)$内解决。

一次考虑一层的贡献。

先做预处理，设$f[i][j][k]$表示这一层的个数为i，上一层有j个$d=2$的节点，有k个$d=3$的节点时的方案数。注意这里没有考虑这一层的点与下面的点以及这一层的点之间的连边情况。f的转移只需每次考虑一个上层的点的连边情况，可以在$O(n^3)$内解决。

接下来再设$g[i][j]$表示现在已经考虑了$i-n$的点的连边情况，其中$i-j$是当前最上一层时的方案数，转移只需要枚举低一层的点的个数。同样是$O(n^3)$的复杂度。

SDOI2011消防

给出一棵带边权的树，现在可以在树上选择两个距离不超过s的点(可以相同)，最小化所有点到以这两个点为端点的路径的距离的最大值。

SDOI2011消防

给出一棵带边权的树，现在可以在树上选择两个距离不超过s的点(可以相同)，最小化所有点到以这两个点为端点的路径的距离的最大值。

$n \leq 3 \times 10^5$

首先一定是在树的直径上选一段，这样答案不会超过直径长的一半，否则一定会超过。

首先一定是在树的直径上选一段，这样答案不会超过直径长的一半，否则一定会超过。
找出直径后算出每个点到直径的距离，接下来就只需要二分答案，判断一下在满足二分值的情况下选择的点的距离是否超出限制。$O(n\log n)$

SDOI2011消防

首先一定是在树的直径上选一段，这样答案不会超过直径长的一半，否则一定会超过。
找出直径后算出每个点到直径的距离，接下来就只需要二分答案，判断一下在满足二分值的情况下选择的点的距离是否超出限制。$O(n \log n)$

CF Edu26 F

给出一个数列A_0，定义数列A_i是A_{i-1}的前缀和。求最小的i，使得存在j，$A_{i,j} > k$。

CF Edu26 F

给出一个数列A_0，定义数列A_i是A_{i-1}的前缀和。求最小的i，使得存在j，$A_{i,j} > k$。

$0 \le A_{0,i} \le 10^9, k \le 10^{18}$

CF Edu26 F

注意到数列中的0不会影响答案，直接把它们删掉。

CF Edu26 F

注意到数列中的0不会影响答案，直接把它们删掉。

如果剩下的数列长度较长，答案一定很小，直接暴力即可。

数列长度较短时，我们可以通过矩阵乘法快速推出A_i，接下来只需要二分答案判断是否有大于k的值就可以了。$O(\log^4 k)$

CF371 Div1 D

给出一个$n \times m$的0/1矩阵，q次询问，每次询问一个矩形内最大的全为0的正方形边长。

CF371 Div1 D

给出一个$n \times m$的0/1矩阵，q次询问，每次询问一个矩形内最大的全为0的正方形边长。

$n, m \leq 10^3, q \leq 10^6$

CF371 Div1 D

首先求出以每个点为右下角的最大全0正方形的边长：

$$f[i][j] = min\{f[i-1][j], f[i][j-1], f[i-1][j-1]\} + 1$$

CF371 Div1 D

首先求出以每个点为右下角的最大全0正方形的边长：

$$f[i][j] = min\{f[i-1][j], f[i][j-1], f[i-1][j-1]\} + 1$$

二分答案k，那么我们每次只需要判断一个矩形内是否存在$f[i][j] \geq k$。

CF371 Div1 D

首先求出以每个点为右下角的最大全0正方形的边长：

$$f[i][j] = min\{f[i-1][j], f[i][j-1], f[i-1][j-1]\} + 1$$

二分答案k，那么我们每次只需要判断一个矩形内是否存在$f[i][j] \geq k$。

用二维RMQ快速处理最大值，$O(nm \log n \log m + q \log n)$

CF413 Div1+Div2 G

给出一个n个点的凸包，q次询问，每次给出凸包内的一个点，求过该点的一条直线，将凸包分成面积相等的两半。

CF413 Div1+Div2 G

给出一个n个点的凸包，q次询问，每次给出凸包内的一个点，求过该点的一条直线，将凸包分成面积相等的两半。

$n \le 10^4, q \le 10^5$

CF413 Div1+Div2 G

设$f(\alpha)$为直线与x轴正方向的夹角为α时，直线左边的面积与右边面积的差。

那么$f(0) = -f(\pi)$，由于f是连续函数，我们只需要在$[0, \pi)$内二分求f的零点。

设$f(\alpha)$为直线与x轴正方向的夹角为α时，直线左边的面积与右边面积的差。

那么$f(0)=-f(\pi)$，由于f是连续函数，我们只需要在$[0,\pi)$内二分求f的零点。

接下来需要解决的问题是快速求$f(\alpha)$，首先需要求出直线与凸包的两个交点，这个问题可以在凸壳上二分哪条边与直线相交来解决；然后需要求一边的面积，我们可以用三角形面积的前缀和快速计算。

设$f(\alpha)$为直线与x轴正方向的夹角为α时，直线左边的面积与右边面积的差。

那么$f(0) = -f(\pi)$，由于f是连续函数，我们只需要在$[0, \pi)$内二分求f的零点。

接下来需要解决的问题是快速求$f(\alpha)$，首先需要求出直线与凸包的两个交点，这个问题可以在凸壳上二分哪条边与直线相交来解决；然后需要求一边的面积，我们可以用三角形面积的前缀和快速计算。

$O(n \log n \log \omega)$，其中ω是精度要求。

CF82 Div2 E

给出n个点，求最小球覆盖。

$n \leq 100$

CF82 Div2 E

本题有直接枚举的做法，但复杂度不优。当然还可以用模拟退火/爬山算法优化，但在本题容易被卡精度，这里略过。

CF82 Div2 E

本题有直接枚举的做法，但复杂度不优。当然还可以用模拟退火/爬山算法优化，但在本题容易被卡精度，这里略过。
不难猜到做法：分别三分球的球心的三维的坐标，找到最远点即是球的半径。
$O(n\log^3\omega)$，其中ω是精度要求。

CF82 Div2 E

本题有直接枚举的做法，但复杂度不优。当然还可以用模拟退火/爬山算法优化，但在本题容易被卡精度，这里略过。
不难猜到做法：分别三分球的球心的三维的坐标，找到最远点即是球的半径。
$O(n\log^3\omega)$，其中ω是精度要求。
能够使用三分的条件是三分的函数是凸函数，证明三分的正确性？

CF82 Div2 E

首先凸函数有一些简单的性质：若干凸函数的和仍然是凸函数，若干下凸函数取max仍然是下凸函数，若干上凸函数取min仍然是上凸函数。于是我们就只需要考虑一个点的情况了。

CF82 Div2 E

首先凸函数有一些简单的性质：若干凸函数的和仍然是凸函数，若干下凸函数取max仍然是下凸函数，若干上凸函数取min仍然是上凸函数。于是我们就只需要考虑一个点的情况了。

定义$d(x, y, z)$为(x, y, z)到一个确定的点的距离。

对于确定的x_0, y_0，定义$f_1(z) = d(x_0, y_0, z)$，显然$f_1(z)$是凸函数。

CF82 Div2 E

对于确定的x_0，定义$f_2(y)$为$y_0 = y$时，$f_1(z)$的最小值，那么$f_2(y)$也是凸函数。

CF82 Div2 E

对于确定的x_0，定义$f_2(y)$为$y_0 = y$时，$f_1(z)$的最小值，那么$f_2(y)$也是凸函数。

假设它不是凸函数，我们一定可以找到两个局部最小值，考虑两个取到最小值的点的连线，那么在这条连线上d不是关于点在直线上位置的凸函数，这与一维情形下f_1是凸函数的结论矛盾。

对于确定的x_0，定义$f_2(y)$为$y_0 = y$时，$f_1(z)$的最小值，那么$f_2(y)$也是凸函数。
假设它不是凸函数，我们一定可以找到两个局部最小值，考虑两个取到最小值的点的连线，那么在这条连线上d不是关于点在直线上位置的凸函数，这与一维情形下f_1是凸函数的结论矛盾。
这样的证明很容易推广到三维。

对于确定的x_0，定义$f_2(y)$为$y_0 = y$时，$f_1(z)$的最小值，那么$f_2(y)$也是凸函数。
假设它不是凸函数，我们一定可以找到两个局部最小值，考虑两个取到最小值的点的连线，那么在这条连线上d不是关于点在直线上位置的凸函数，这与一维情形下f_1是凸函数的结论矛盾。
这样的证明很容易推广到三维。
感觉很复杂？所以还是猜结论吧。

ZJOI2016旅行者

一张$n \times m$的带权网格图，q次询问两点之间的最短路。

ZJOI2016旅行者

一张$n \times m$的带权网格图，q次询问两点之间的最短路。

$n \times m \le 2 \times 10^4, q \le 10^5$

考虑分治，每次将网格沿一条分界线分成两半，则跨过分界线的询问一定会经过分界线上一点，未跨过的询问也有可能经过分界线。

考虑分治，每次将网格沿一条分界线分成两半，则跨过分界线的询问一定会经过分界线上一点，未跨过的询问也有可能经过分界线。

于是每次从分界线上每一个点开始做最短路，对于每个询问枚举一下经过了分界线上的哪个点更新答案。然后再将未跨过分界线的询问分治到两个子矩形处理。

考虑分治，每次将网格沿一条分界线分成两半，则跨过分界线的询问一定会经过分界线上一点，未跨过的询问也有可能经过分界线。

于是每次从分界线上每一个点开始做最短路，对于每个询问枚举一下经过了分界线上的哪个点更新答案。然后再将未跨过分界线的询问分治到两个子矩形处理。

$O(nm\sqrt{nm}\log nm + q\sqrt{nm})$?卡常数好题。

HNOI2010城市建设

n个点m条边的带权无向图，q次修改边权，每次修改后询问最小生成树。

HNOI2010城市建设

n个点m条边的带权无向图，q次修改边权，每次修改后询问最小生成树。

$n \leq 2 \times 10^4, m, q \leq 5 \times 10^4$

可以想到按时间分治，然而问题在于如何缩小问题的规模。

可以想到按时间分治，然而问题在于如何缩小问题的规模。
考虑如何减少点和边，我们可以找到一定会在MST中的边和一定不在MST中的边。

可以想到按时间分治，然而问题在于如何缩小问题的规模。
考虑如何减少点和边，我们可以找到一定会在MST中的边和一定不在MST中的边。
将需要修改的边的边权赋为$-\infty$做$Kruscal$，此时如果一条不需要修改的边在MST中，那么它一定在MST中，将这样的边的边权加起来，然后缩点；

可以想到按时间分治，然而问题在于如何缩小问题的规模。
考虑如何减少点和边，我们可以找到一定会在MST中的边和一定不在MST中的边。
将需要修改的边的边权赋为$-\infty$做$Kruscal$，此时如果一条不需要修改的边在MST中，那么它一定在MST中，将这样的边的边权加起来，然后缩点；

类似的，将需要修改的边权赋成∞，如果一条不需要修改的边不在MST中，那么它一定不在MST中，可以直接删掉。

类似的，将需要修改的边权赋成∞，如果一条不需要修改的边不在MST中，那么它一定不在MST中，可以直接删掉。
注意到这样做之后，图的规模只与要修改的边条数有关。分治到最后由于点数很少就可以暴力算MST了。

类似的，将需要修改的边权赋成∞，如果一条不需要修改的边不在MST中，那么它一定不在MST中，可以直接删掉。
注意到这样做之后，图的规模只与要修改的边条数有关。分治到最后由于点数很少就可以暴力算MST了。
$O(m\log m\log q)$，代码实现颇具HNOI风格。

CF426 Div1 D

一棵n个点的带边权的树，每条边要么是红色，要么是黑色。
定义一条路径的得分是路径上所有边权的乘积；
定义一条路径上红色边的数量为R，黑色边的数量为B，那么路径是合法的当且仅当$max(B, R) \leq 2 \times min(B, R)$。
求所有合法路径的得分的乘积，对$10^9 + 7$取模。

CF426 Div1 D

一棵n个点的带边权的树，每条边要么是红色，要么是黑色。
定义一条路径的得分是路径上所有边权的乘积；
定义一条路径上红色边的数量为R，黑色边的数量为B，那么路径是合法的当且仅当$max(B, R) \leq 2 \times min(B, R)$。
求所有合法路径的得分的乘积，对$10^9 + 7$取模。
$n \leq 10^5$

CF426 Div1 D

对于树上路径问题点分治是常见解法，也就是每次找到树的重心，考虑经过重心的路径对答案造成的贡献，然后再分治解决重心的各个子树内部的路径贡献即可。

对于树上路径问题点分治是常见解法，也就是每次找到树的重心，考虑经过重心的路径对答案造成的贡献，然后再分治解决重心的各个子树内部的路径贡献即可。

对于子树中一条(B_1, R_1)(表示有B_1条黑色边，R_1条红色边)的路径和一条(B_2, R_2)的路径，那么这两条路径组合在一起是合法的，当且仅当

$$\begin{cases} B_1 + B_2 \le 2R_1 + 2R_2 \\ R_1 + R_2 \le 2B_1 + 2B_2 \end{cases} \tag{1}$$

移项有：

$$\begin{cases} B_1 - 2R_1 \le 2R_2 - B_2 \\ R_1 - 2B_1 \le 2B_2 - R_2 \end{cases} \tag{2}$$

移项有：

$$\begin{cases} B_1 - 2R_1 \le 2R_2 - B_2 \\ R_1 - 2B_1 \le 2B_2 - R_2 \end{cases} \tag{2}$$

注意到两个不等式取反不存在交集，那我们对于每条路径只需要求出之前的子树中有多少满足第一个条件的，再去掉不满足第二个条件的即可。

移项有：

$$\begin{cases} B_1 - 2R_1 \le 2R_2 - B_2 \\ R_1 - 2B_1 \le 2B_2 - R_2 \end{cases} \tag{2}$$

注意到两个不等式取反不存在交集，那我们对于每条路径只需要求出之前的子树中有多少满足第一个条件的，再去掉不满足第二个条件的即可。

可以用线段树维护，$O(n\log^2 n)$。

Thanks