好题选讲

SDSZ

2018 年 6 月 21 日

1 第K大区间

1.1 问题描述

定义一个长度为奇数的区间的值为其所包含的的元素的中位数。

现给出n个数，求将所有长度为奇数的区间的值排序后，第k大的值为多少。

$n \leq 10^5$ $k \leq$奇数区间数量

1.2 问题求解

定义一个长度为奇数的区间的值为其所包含的的元素的中位数。

现给出n个数，求将所有长度为奇数的区间的值排序后，第k大的值为多少。

$n \leq 10^5$ $k \leq$奇数区间数量

考虑二分答案，设此时二分的答案为x，可以根据每个数和x的大小关系将序列变为-1，1序列（-1为比x小，1为大于等于x），一个区间内的中位数大于等于x当且仅当$\sum_{i=l}^{r} a_i \geq 1$（其中$a_i$为转换后的序列），这就可以直接用树状数组统计答案了，时间复杂度$O(nlog^2n)$

当然，我们还有一种$O(nlogn)$的做法，我们可以发现，每个转换后的数值仅为-1和1，那对于前缀和的改变也是-1和1，对于值域数组上一个区间左端点的改变也就是-1和1，改变的时间复杂度为$O(1)$，对于查询后的修改操作，我们只需要判断当且左端点和修改位置的大小关系就可以$O(1)$进行修改，总的时间复杂度为$O(nlogn)$

2 Candy Piles

2.1 题目来源

AtCoder Grand Contest 002 E

2.2 问题描述

现在桌子上有N堆糖果，这N堆糖果用1至N编号。初始时，第i堆糖果中有a_i颗糖果。

Snuke和Ciel正在玩一个游戏。他们轮流操作，Snuke先手。在每次操作中，当前玩家必须完成下面两个操作之一：

选择剩余糖果数最多的一堆，并把这堆糖果全部吃掉。

在仍有糖果的每一堆中各吃掉一颗糖果。

吃掉桌上最后一颗糖果的玩家，为游戏的负方。请求出在两个人都使用最优策略的时候，谁会赢得游戏。

$N \leq 10^5$ $a_i \leq 10^9$

2.3 问题求解

现在桌子上有N堆糖果，这N堆糖果用1至N编号。初始时，第i堆糖果中有a_i颗糖果。

Snuke和Ciel正在玩一个游戏。他们轮流操作，Snuke先手。在每次操作中，当前玩家必须完成下面两个操作之一：

选择剩余糖果数最多的一堆，并把这堆糖果全部吃掉。

在仍有糖果的每一堆中各吃掉一颗糖果。

吃掉桌上最后一颗糖果的玩家，为游戏的负方。请求出在两个人都使用最优策略的时候，谁会赢得游戏。

$N \leq 10^5$ $a_i \leq 10^9$

直接考虑问题不太容易，我们考虑将问题转化一下。

将糖果按数量从大往小排序，建立一个网格，在第i列的网格上填上a_i糖果，可以发现两种操作分别对应将一行的糖果去除与一列的糖果去除，最后去除糖果的一方就输了。进一步，我们可以在进行一下转化，另一个枚棋子放在$(0,0)$的位置（即第1 列第1行网格的左下角），每次只能向上走与向右走，最后谁到达糖果在右上方所构成的边界的人就输了。我们发现可以直接进行dp，但时间复杂度为$O(\sum a_i)$，不足以通过此题。

再进行观察，我们发现，每个斜线$(y = x + b)$上每个点的胜负状态是一样的，证明类似数学归纳法，对于坐标为$(x + 1, y + 1)$的状态，若其为负状态，则$(x + 1, y)$与$(x, y + 1)$的状态必然都是胜状态，因此(x, y)为负状态，反过来也是一样，因此我们就可直接将$(0, 0)$的状态定位到边界线上的状态，在枚举一下往上和右情况下的输赢，就能知道当前局面的输赢了，时间复杂度$O(NlogN)$

3 Go Home

3.1 题目来源

AtCoder Grand Contest 023

3.2 问题描述

在一条数轴线上有N座公寓，其中第i座公寓的坐标X_i，$Atcoder$公司在S的位置上，每个员工都住在其中一个公寓里，其中第i个公寓里面有P_i个人。

所有的员工都在同一时刻做公司的大巴回家，大巴会离开S这个位置并按如下规则移动：

每个人在大巴车上会投票选出大巴应该往哪个方向行驶，沿x轴正方向或者负方向。每人都有一票，并且不允许不投票。大巴将会向多的那个方向行驶1的距离，如果出现平局，就往负方向行驶。如果行驶到了一个位置为公寓，所有住在这个公寓里的员工都会下车。重复以上步骤，直到所有人都下车了。

每个人都会投能让自己更早回家的方向，更准确的说，在假设所有人都按照这个策略的投票的前提下，每个人会看他往哪个方向投票能使自己尽早回家，如果两个方向是一样的，他会投负方向。

请你求出大巴车最终多长时间停下。

$N \leq 10^5$ ，X_i，P_i和$S \leq 10^9$

3.3 问题求解

在一条数轴线上有N座公寓，其中第i座公寓的坐标X_i，$Atcoder$公司在S的位置上，每个员工都住在其中一个公寓里，其中第i个公寓里面有P_i个人。

所有的员工都在同一时刻做公司的大巴回家，大巴会离开S这个位置并按如下规则移动：

每个人在大巴车上会投票选出大巴应该往哪个方向行驶，沿x轴正方向或者负方向。每人都有一票，并且不允许不投票。大巴将会向多的那个方向行驶1的距离，如果出现平局，就往负方向行驶。如果行驶到了一个位置为公寓，所有住在这个公寓里的员工都会下车。重复以上步骤，直到所有人都下车了。

每个人都会投能让自己更早回家的方向，更准确的说，在假设所有人都按照这个策略的投票的前提下，每个人会看他往哪个方向投票能使自己尽早回家，如果两个方向是一样的，他会投负方向。

请你求出大巴车最终多长时间停下。

$N \leq 10^5$ ， X_i和$S \leq 10^9$

直接按照每个人下车的顺序考虑这道题比较困难，我们反着来想。

当倒数第三个公寓里的人下车后，数轴上就只剩下坐标为1和坐标为N的两个公寓，此时若$P_1 \geq P_n$ 则，不管怎么样，N号公寓一定是最后一个到达的，为了使自己的到达时间最少，他们就会希望1号公寓里的到达时间最少，因此他们就一定会投和1号公寓一样的票直到倒数第三个公寓里的人下车（虽然之后投和1号公寓不一样的票没有什么意义）。因此，我们可以认为N号里的人搬到1号公寓中，即将P_1加上P_n，而搬过去所走的路程我们可以当大巴方向改变时再加上。同理当$P_1 < P_n$的做法是类似的，这样我们就可以将一个规模为N的问题转变为了规模为$N-1$的问题，我们就可以在$O(N)$的时间复杂度下内完成了这道题目。

4 Countari

4.1 题目来源

CodeChef

4.2 问题描述

给出A_1，A_2,……，A_n，统计满足$i \leq j \leq k$，A_i+A_k=2A_j，(i,j,k)的数量。

$N \leq 10^5$，$A_i \leq 30000$

4.3 问题求解

给出A_1，A_2,……，A_n，统计满足$i \le j \le k$，A_i+A_k=2A_j，(i,j,k)的数量。

$N \le 10^5$，$A_i \le 30000$

题目要求比较奇怪，容易想到转化成生成函数来思考。但i,j,k的位置关系比较难处理。

考虑将A序列分块，则i,j,k的位置关系分以下几种情况。

(1)i,j,k分别属于三个不同的块：枚举j所在的块，发现对于某个j，满足条件的i,k对数是卷积形式，可以通过对两边的整块做FFT计算。

(2)i,j,k属于同一个块：枚举j所在的块，枚举j,k，枚举过程中，随时更新小于j的位置的数的出现次数。

这样对于不同的k ，满足条件的i的个数可以$O(1)$得到。

(3)i,j属于同一个块：枚举j所在的块，枚举i,j，更新j所在块右侧的数的出现次数，这样对于不同的i,满足条件的k的个数可以$O(1)$得到。

(4)j,k属于同一个块：枚举j所在的块，枚举j,k，更新j所在块左侧的数的出现次数，这样对于不同的k，满足条件的i的个数可以$O(1)$得到。

设块大小为S，A_i值域为W，(1)复杂度为$\frac{NWlogW}{S}$，(2)(3)(4)复杂度为nS,令$S = \sqrt{WlogW}$，则总复杂度为$N\sqrt{WlogW}$

5 Rotatable Number

5.1 题目来源

Codeforces 303 D

5.2 问题描述

定义一个x位数A为“可旋转数”当且仅当A，$2A$，$3A$.....$(x-1)A$均可以由A旋转出来（比如说142857）。现在将“可旋转数”推广到b进制，（如142857为10进制的旋转数，0011为2进制的旋转数），现在让你找到一个最大的b，满足$(1 < b < x)$且在b进制下存在一个长度为n的正“可旋转数”（允许有前导零），若不存在请输出-1。

$1 \leq n \leq 5 * 10^6$，$2 \leq x \leq 10^9$

5.3 问题求解

定义一个x位数A为“可旋转数”当且仅当A，$2A$，$3A$.....$(x-1)A$均可以由A旋转出来（比如说142857）。现在将“可旋转数”推广到b进制，（如142857为10进制的旋转数，0011为2进制的旋转数），现在让你找到一个最大的b，满足$(1 < b < x)$且在b进制下存在一个长度为n的正“可旋转数”（允许有前导零），若不存在请输出-1。

$1 \le n \le 5 * 10^6$，$2 \le x \le 10^9$

不妨假设我们要考虑的数为A，（它的位数为b），构造这样一个数x=$0.AAAAA...$（在b进制下），由于它是纯循环小数，x可以表示为q/p 的形式并且$gcd(p,q)=1$。考虑这样两列数，其中B=$(x, 2x, 3x....(p-1)x)$，C=$(x, xb, xb^2....xb^{p-2})$，由循环数的定义可知，这两个集合中每个数的小数部分（在b 进制下）组成的新的集合是一样的，而小数部分只由和模p的值有关，也就是说，集合D=$(q, 2q, 3q...(p-1)q)$与集合E=$(q, qb, qb^2....qb^{p-2})$在模$p$意义下是相同的。又因为$gcd(p,q)=1$，所以$q$对$p$有乘法逆元$q^{-1}$，也就是说集合$D$=$(1, 2, 3...(p-2))$，与集合$E$=$(1, b, b^2....b^{p-2})$在模$p$意义下是一样的，由于$b^{\phi(p)}=1$，所以$\phi(p) \ge p-1$，所以$p$必须是质数，并且$b$是$p$的一个原根。有了这个结论，再加上原根的数量是相当多的，我们就可以直接从大往小枚举进制b，判它是不是原根就好了，令k为原根分布密度，则时间复杂度$O(n+k\sqrt{n})$

6 Numbers

6.1 题目来源

Codeforces 241 D

6.2 问题描述

你有1,2.....n的排列a_1,a_2.....a_n。你想删除一些整数,使得结果序列满足以下三个条件:

1.由此产生的序列不是空的;

2.序列中所有数异或和等于0;

3.如果你把所有数按十进制从前往后依次无间隔地写在一行形成一个大的十进制数，这个数将会被p整除。

给你序列和p，找到一种方法来满足上述条件。

$1 \leq n, p \leq 50000$,且p为质数。

6.3 问题求解

你有1,2.....n的排列a_1,a_2.....a_n。你想删除一些整数,使得结果序列满足以下三个条件:

1.由此产生的序列不是空的;

2.序列中所有数异或和等于0;

3.如果你把所有数按十进制从前往后依次无间隔地写在一行形成一个大的十进制数，这个数将会被p整除。

给你序列和p，找到一种方法来满足上述条件。

$1 \leq n, p \leq 50000$,且p为质数。

直接来看好像并没有什么做法，那我们将n变为63的话，是不是很简单？

定义f_{ijk}表示前i个数，目前异或值为j，除以p的余数为k是否可行，转移就是枚举下一位选不选即可。

那这个和原题又有什么关系呢？

考虑这样一件事，我们把原序列中所有大于等于64的数全部删掉，在剩余的数里进行dp，用dp结果构造解出来。

这样可以保证正确性吗？

我们可以发现由63个数组成的排列最多有$2^{63}-1$种非空序列，而对于（异或值，余数）这样的二元组，实际上只有$64p$个，远小于非空序列个数，所以也就可以保证正确性了。

7 Maze

7.1 问题描述

给你一棵由N个节点组成的树，初始时间为1，你一开始在根节点。每个时间单位你可以到达一个与你目前所有已经到达过的节点所组成的联通快中相邻的一个未到达节点，花费的代价为$T \cdot a_i$，其中T为当前时间。问把整棵树遍历完的最小代价。

$N \leq 5000$

7.2 问题求解

给你一棵由N个节点组成的树，初始时间为1，你一开始在根节点。每个时间单位你可以到达一个与你目前所有已经到达过的节点所组成的联通快中相邻的一个未到达节点，花费的代价为$T \cdot a_i$，其中T为当前时间。问把整棵树遍历完的最小代价。

$N \leq 5000$

考虑这样一个问题：对于一个节点x，它的其中两个儿子p_1，p_2，我们先走p_1所在联通块，再走p_2 所在联通块与先走p_2所在联通块再走p_1所在联通块哪一个更优呢？

你可能对于所在联通块有些疑问，这很正常，因为它和我们后面要解决这个问题的算法是有关系的。

设p_1所在联通块的大小为s_1，p_2所在联通块的大小为s_2，则有如下式子：

先p_1后p_2的代价$A = \sum_{i=1}^{s_1} a_i \cdot i + \sum_{i=1}^{s_2} b_i \cdot (i + s_1)$

先p_2后p_1的代价$B = \sum_{i=1}^{s_2} b_i \cdot i + \sum_{i=1}^{s_1} a_i \cdot (i + s_2)$

令$C = A - B$，可以得到$C = \sum_{i=1}^{s_2} b_i \cdot s_1 - \sum_{i=1}^{s_1} a_i \cdot s_2$

若$C > 0$则$\frac{\sum_{i=1}^{s_2} b_i}{s_2} > \frac{\sum_{i=1}^{s_1} b_i}{s_1}$，即$p_1$这个联通快的平均权值小于$p_2$这个联通快的平均权值。

有了这个结论，我们又能做什么呢？

考虑这样一个算法，一开始我们认为树上所有节点所在联通块就只有它自己，即联通块大小都为1，每个联通块值就为它上原本的值。接下来重复以上步骤$n-1$次直到只剩一个联通块：

找到联通块值最大的联通块，将答案加上走联通块的值加上其父亲联通块的个数，并将其的值合并到它的父亲上，即将这其父亲所在联通块的值改为新的联通块的平均权值。这个算法实质上就是在用我们的那一个结论，当我们第一次走到这个节点的父亲节点时，由这个结论我们必然会先走完平均权值最大的那个联通块，所以我们就可以将这个联通块合并到其父亲，转化为了一个更小的问题，这样就证明了这个算法的正确性，时间复杂度$O(N^2)$

8 Letter Stamper

8.1 题目来源

GCJ 2010 Final A

8.2 问题描述

有一个栈式打字机，支持以下三种操作：

1.把一个字母加到栈顶

2.打印栈顶的字母

3.删除栈顶字母

给出你需要打印的字母等级序列，你需要算出打印这个序列的最小操作数。

$T \leq 20$，$\sum |S| \leq 2000$，字母只由A，B，C组成

8.3 问题求解

有一个栈式打字机，支持以下三种操作：

1.把一个字母加到栈顶

2.打印栈顶的字母

3.删除栈顶字母

给出你需要打印的字母等级序列，你需要算出打印这个序列的最小操作数。

$T \leq 20$，$\sum |S| \leq 2000$，字母只由A，B，C组成

首先栈顶上的前两个元素一定不是由两种同样的字母组成，不妨假设栈顶为A，弹出栈顶后为B，考虑以下两种情况：

若下一个字符为A，则我们不需要任何的操作，就能直接打印这个字符。

若下一个字符为B，那我们就有两种选择：弹出栈顶的元素，或是新加一个字符B，我们发现这样两种操作的花费都是1，并且前一种的栈的大小比后一种还要小，所以我们一定会去选择第一种方案。

若下一个字符为C，则我们即可以弹出顶上两个元素，又可以新加一个元素。

由以上性质我们就可以发现对顶的元素一定是A，B，C，A，B，C，A.....，所以我们只要记录栈内最下面三个元素的顺序即可。

定义f_{ijt}表示考虑过前i个字母，目前栈的高度为j，最下面三个元素的情况为t的最小操作数，转移就简单看一下我上面说的三种情况即可。时间复杂度$O(|S|^2T)$