A

---

marp: true

Dragon Boat

RegMs If

题意

给定n个正整数a_i，求有多少个非负整数x，满足x小于所有给定的数，且所有给定的数减去x之后，Nim游戏后手必胜。

数据范围：$1 \le n \le 2 \times 10^5, 1 \le a_i \le 10^{18}$。

SG定理

后手必胜的条件是所有数的异或和为0。

逐位考虑

枚举x显然不可行。可以尝试从低位到高位依次确定。

考虑低位对高位的影响，易知只有当低位退位时才会改变高位。而对于每一位，若此位上有偶数个0，则此位可以填1，若有偶数个1则可以填0（这样才能使得异或和为0）。

DP

令$f[i][j]$表示从低到高处理到第i位时，有j个数字在第$(i+1)$位退位的方案数。

根据退位的性质，这j个数字一定是所有给定的数按后i位从小到大排序后的前j个。

基数排序

枚举i的同时基数排序。

枚举j的同时维护第i位上0和1的个数以及第$(i+1)$位上退位的个数（会受到第i位退位个数的影响）。

最后需要特判一下x不能等于所有给定的数中最小的数。

核心代码

```
f[0][0] = 1;
for (int i = 0; i < M - 1; ++i) {
 int c0 = n - cnt[i], c1 = cnt[i], c = 0;
 for (int j = 0; j <= n; ++j) {
  if (j)
   if (a[j - 1] >> i & 1)
    ++c0, --c1;
```

```
      else
        --c0, ++c1, ++c;
      if (!(c0 & 1))
        f[i + 1][c + c0] += f[i][j];
      if (!(c1 & 1))
        f[i + 1][c] += f[i][j];
    }
    *s[0] = *s[1] = 0; // radix sort
    for (int j = 0; j < n; ++j)
      s[a[j] >> i & 1][++*s[a[j] >> i & 1]] = a[j];
    for (int j = 1; j <= *s[0]; ++j)
      a[j - 1] = s[0][j];
    for (int j = 1; j <= *s[1]; ++j)
      a[*s[0] + j - 1] = s[1][j];
  }
```

Contour

RegMs If

题意

平面上有n个点，现在要用这n个点构造一个图形，满足：

- 这个图形是闭合的；

- 每条边的端点必须是给定的点，且每个给定的点必须被用到；
- 每个点连接的两条边必须互相垂直；
- 每条边必须平行于坐标轴；
- 任意两条边除了顶点外不能相交；
- 这个图形的周长最小。

如果有解，输出最小周长，否则输出0。

数据范围：$4 \le n \le 10000$。

构造

将所有点以横坐标为第一关键字、纵坐标为第二关键字从小到大排序。

如果有解，那么对于任意整数x，横坐标等于x的点的个数必须是偶数个，只要将第一个点和第二个点、第三个点和第四个点……相连，再以纵坐标为第一关键字、横坐标为第二关键字从小到大排序，进行一样的操作。

易知如果存在满足条件的图形，则它是唯一的。

最后需要判断是否连通（并查集）以及是否相交（暴力）。

另解

对y坐标建树状数组，按x坐标从小到大扫描，同时判断是否连通以及是否相交。

B

07/18 groupB
 problemB solution

MUG2

用音游和车万给大家带来温暖

writer:l1ll5 联系方式：QQ2817629709


简述题意：

一棵树，n个节点，点有黑白颜色(not trivial or trivial)和权值。从1出发随机游走，终点是deg<=1的点。经过黑点和第一次经过白点可以带来对应权值的贡献，求和的期望。

n<=20w

考虑点的贡献独立，黑白点对于游走过程互不影响。
分别考虑黑点和白点的答案，而这两个都是经典问题。

白点：考虑到每个点只要经过必然带来贡献，令 dp_i 表示从 1 出发经过点 i 的概率，那么 i 的贡献就是 $a_i \cdot dp_i$
考虑求出 dp_i，考虑到第一次经过一个点必然是从它的父亲走过来的，令 $down_i$ 表示从 fa_i 走到 i 的概率，则
$dp_i = dp_{fa} \cdot down_i$
如何求出 $down_i$ 是经典问题，设 up_i 表示从 i 走到 fa_i 的概率
$down_i = \frac{1}{deg_{fa}}(1 + down_{fa} \cdot down_i + \sum_{son} up_{son} \cdot down_i)$
$up_i = \frac{1}{deg_i}(1 + \sum_{son} up_{son} \cdot up_i)$

$$ans_{white} = \sum_{i, i\ is\ white} dp_i \cdot a_i$$

黑点：直接令 dp_i 表示从 i 出发游走至结束的答案的期望。
$dp_i = a_i \cdot c_i + \frac{1}{deg_i}\sum_{to} dp_{to}$

发现 to 既包括儿子也包括父亲，无法dp

令 $f_i = k_i \cdot f_{fa} + b_i$ 直接替换到上式中，则上式只与 i 和 fa 有关。
对于 1 号点，$k_1 = 0, b_1 = dp_1 = ans_{black}$

C - 结界「点与线的境界」

题意

给你n×m个格点，两两连直线，求直线数。

题解

主要是思路题，式子对了基本就做完了。

横的线加竖的线有n+m条。

斜的线可以用类似容斥的思路处理，考虑斜率为正的线。

i×j的矩形有 $(n-i)(m-j)$ 个，若gcd(i,j)=1则加上去，若gcd(i,j)=2则减掉。

img

$$\sum_{i=1}^{n-1}\sum_{j=1}^{m-1}(n-i)(m-j)[gcd(i,j)==1]-\sum_{i=1}^{n-1}\sum_{j=1}^{m-1}(n-i)(m-j)[gcd(i,j)==2]$$

把小括号里面拆出来，维护三个前缀和，多次莫反就做完了。

要点

$\sum_{i=1}^{n}\sum_{j=1}^{m}(n-i+m-j-1)[gcd(i,j)==1]$ 为错解，至于为何错，可自行计算n=3 m=4看遗漏了哪些线。

注意n, m中存在1的情形，直接输出答案。

D：Cities

2020 年 7 月 18 日

1 思路

显然的思路是用图处理，把每一个选择处理为从一条边的一个端点走到另一个。那么第一种选择则是等价于在每个点上连一个自环，第二个选择等价于在u，v之间连一条双向边，对于第三个选择我们加入一个新的顶点（也有很多人加了n个），并将之前的每个顶点都做一条连向它的单向边。

2 暴力分层dp

考虑到对于每一次转移有方程,对所有的与u相邻的点v

$$f[j+1][u] \doteq f[j+1][u] + f[j][v] \tag{1}$$

虽然可以用滚动数组但至少o(t*n*m)复杂度显然tle

3 矩阵乘法优化

由于dp的tle是由于t过大，我们考虑快速幂和矩阵乘法所以我们最终只需要计算

$$\sum f[1][i] \tag{2}$$

(其中的f是原矩阵g进行t次方后的矩阵)

此时为o($logt$*n^3)复杂度

E

1. m(logn)^2 做法：利用线段树维护节点父亲，利用LCT维护树形态，对线段树上每一个区间建立一个虚拟节点，考虑[L,R]被分为log个区间，对于每个区间，如果区间内节点父亲一样，那么直接把这个区间的虚拟节点取出连到指定父亲上，如果不一样则在线段树上递归下去，合并所有节点(例如[L,R]父亲不一样，那么递归合并[L,Mid],[Mid,R]),考虑一次换父亲会切割log个区间，产生log个区间，所以区间的总数级别为mlogn，所以总复杂度为m(logn)^2.我稍微卡了一下，应该比较难跑过去。

2. mlogn做法：用Treap或者splay维护dfs序，考虑所有结点的父亲，必然是连续的一段是同一个父亲，考虑每次操作会切开头尾两段，并把中间合成一段，那么，m次操作形成的总段树为n级别，考虑每次操作的时候先找出要修改的若干个连续父亲区间，考虑这些点及其子树的dfs序列应该是连续的，利用Treap或splay提取出这些区间，按从左到右的顺序插回当前新父亲以保证dfs序连续，注意，提取的时候要先提取深度深的区间，以防这段区间里的点在深度浅的区间的点的子树中，所以还要维护一下深度，这些都可以在平衡树上完成。