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

```cpp

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

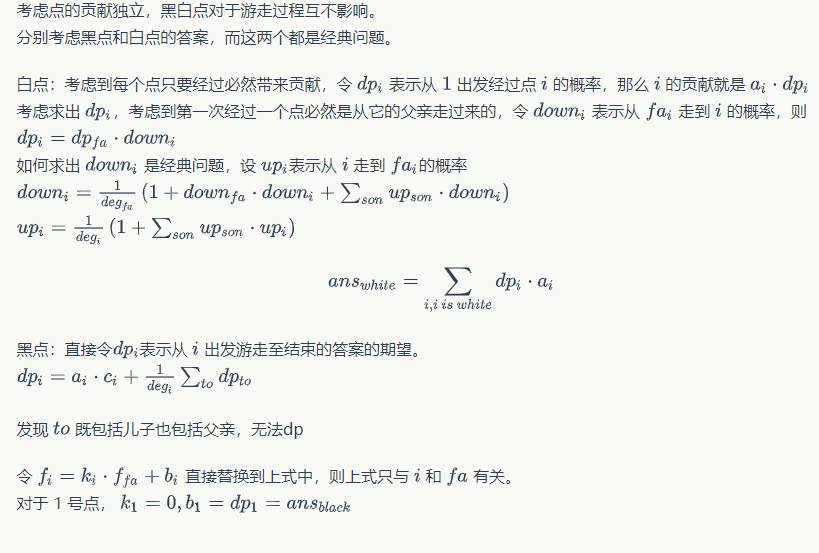
writer:l1ll5 联系方式：QQ2817629709



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



# 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。

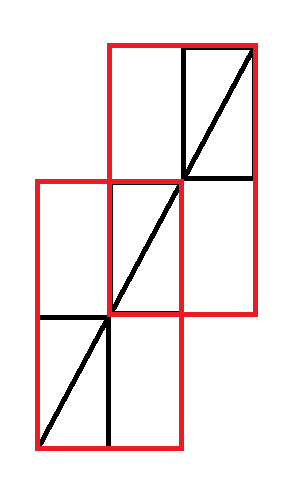
## 题解

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的矩形有个，若gcd(i,j)=1则加上去，若gcd(i,j)=2则减掉。

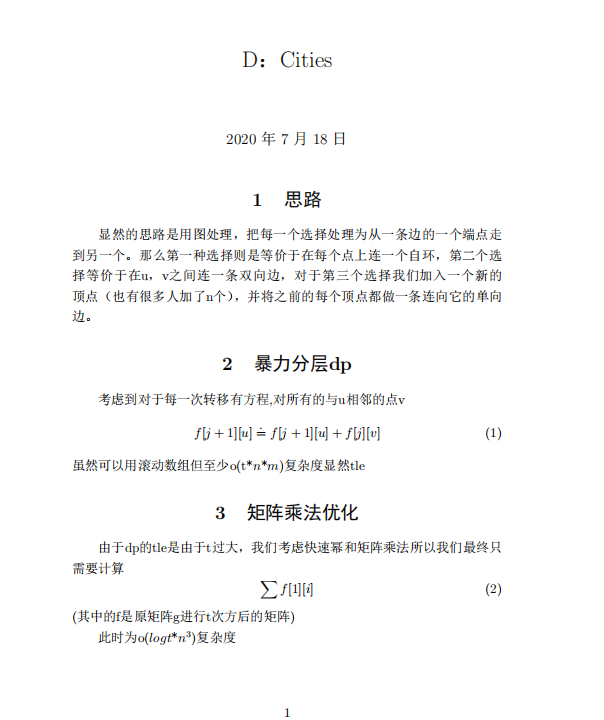


把小括号里面拆出来，维护三个前缀和，多次莫反就做完了。

## 要点

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。



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。