

[比赛链接](#)

## A Assignment Algorithm

solved by qxforever

题目描述

解题思路

## B Buffalo Barricades

upsolved by Potassium

题目描述

平面上有一些牛，依次在某些点加入一些栅栏，问此次加入的栅栏和之前加入的且在左下角的栅栏组成的密闭空间中有多少头牛。保证没有栅栏同 $x$ 或同 $y$

解题思路

不妨只考虑添加完所有栅栏后，某一个 $y$ 下的牛、栅栏分布情况。容易想到这是一个牛和栅栏交替的结构，而栅栏可以跨 $y$ 继续向下延伸，故考虑用 $set$ 存一下栅栏的情况（加入时间 $\square x$ 位置），然后按 $y$ 从大到小进行枚举处理。

将所有牛和栅栏的坐标信息排序，按 $y$ 从大到小 $\square x$ 从大到小、从栅栏到牛的顺序枚举并添加/删除信息，这就能求出最终每个栅栏圈住了几头牛。

在上面的过程中，求出来每个时间 $i$ 放下的栅栏圈住了外面哪个时间更早放下的的栅栏 $outer[i]$ 的牛，用并查集倒序处理即可获得答案。

## C Cumulative Code

unsolved

题目描述

解题思路

## D Donut Drone

unsolved by Potassium

### 题目描述

有一个围成一个甜甜圈的  $r \times c$  矩阵  $a$  (上下可达、左右可达)，初始位置在  $(1,1)$ ，有  $q$  次、两种操作：走  $k \leq 10^9$  步并询问位置；将  $a_{xy}$  赋为  $z$

$r, c \leq 2000, q \leq 5000$

### 解题思路

观察到  $q$  比较小，可以考虑记录循环节，但由于循环节是  $O(rc)$  级别的，考虑降级，即记录  $(x,1)$  走  $c$  步后位于  $(nxt[x],1)$  询问比较容易，问题在于修改。

分别考虑  $(x-1,y-1), (x-1,y), (x-1,y+1)$  三个点，只有到达这三个节点的区间才有可能被更新。维护一下对应的初始区间，很容易考虑到每次  $y$  坐标  $-1$  的时候，区间只会修改端点附近的  $\pm 1$  故通过这个特性更新即可。

## E Embedding Enumeration

unsolved

### 题目描述

### 解题思路

## F Faulty Factorial

solved by Potassium

### 题目描述

给定  $n, p, r (n \leq 10^{18}, p \leq 10^7)$  求一个序列  $a$  满足  $a_i = i, i \neq t; a_i < i, i = t$  ( $t$  可任选)，且  $\prod_{i=1}^n a_i \equiv c \pmod{p}$

### 解题思路

根据 $r$ 是否为 $0$ 讨论。

若 $r=0$ 即可以整除 $p$ 当 $n>p$ 的时候把 $n$ 变为 $p$ 即可当 $n=p$ 的时候把 $2$ 变为 $1$ 即可，这里特判一下 $p=2$ 的情况；否则显然无解。

若 $r \neq 0$ 即不可以整除 $p$ 当 $n \geq 2p$ 的时候，阶乘必然至少有两个 $p$ 的因数，只能改变其中一个当 $r=0$ 必然成立当 $p \leq n < 2p$ 的时候，则考虑修改 $a_p$ 的值当 $n < p$ 的时候枚举每个 $i$ 计算（满足 $\frac{n!}{i!} \times x \equiv r \pmod{p}$ 的 $x$ 即可。

## G Gambling Guide

solved by Potassium

### 题目描述

给一个连通无向图，初始位置在 $1$ 号节点，目标为 $n$ 号节点。每一天，所处位置的机器会等概率随机抽取与当前节点相邻的节点，你可以选择留在原地或者去往这个节点，问采取最优策略的情况下，到达目标的期望时间为多少。

### 解题思路

设 $f[i]$ 表示点 $i$ 到点 $n$ 的期望时间，那么我们必然是从 $f$ 大的向 $f$ 小的节点走。设与 $i$ 直接相连的集合为 $P$ 有

$$f[i] = 1 + \frac{\sum_{p \in P} \min\{f[p], f[i]\}}{\deg[i]}$$

我们将所有点分为两个部分，一部分是已经确定 $f$ 最终值的，这部分设为 $S$ 另一部分是未确定 $f$ 最终值的。

初始时 $f[n]=0, f[i]=\infty, S=\{n\}$ 按照 $f$ 从小到大的顺序从优先队列取出 $S$ 中的元素，依次更新与其相邻且不在 $S$ 中的节点的 $f$

容易观察得到，对于在从点 $i$ 往外更新其他节点之前的所有 $S$ 中的点 $j$ 有 $f[j] < f[i]$ 故当更新到点 $i$ 时，有

$$f[i] = 1 + \frac{\sum_{p \in S} f[p]}{\deg[i]} + \frac{\sum_{p \notin S} f[p]}{\deg[i]}$$

设与 $i$ 相邻的 $S$ 中元素个数为 $cnt[i]$ 其 $f$ 之和为 $s[i]$ 则

$$f[i] = 1 + \frac{s[i]}{\deg[i]} + \frac{\deg[i] - cnt[i]}{\deg[i]} f[i]$$

化简得到

$$f[i] = \frac{s[i] + \deg[i]}{cnt[i] + 1}$$

于是通过类似dijkstra的方法一路取出并向外更新节点即可。

## H Hidden Hierarchy

solved by nikkukun

### 题目描述

### 解题思路

## I Intrinsic Interval

upsolved by Potassium

### 题目描述

给一个  $1..n$  的排列  $a_1..a_n$  次询问区间  $[l,r]$  问最小的覆盖  $[l,r]$  区间的、满足排完序后连续的区间。

### 解题思路

首先考虑扩展区间  $[i,i+1]$  不妨设  $a_i < a_{i+1}$  则要使得  $[i,i+1]$  区间位于连续区间中，区间  $[l,r] = [\min_{a_i \leq j \leq a_{i+1}} \text{pos}_j, \max_{a_i \leq j \leq a_{i+1}} \text{pos}_j]$  必然也位于同一个区间中，这里的  $l,r$  容易通过  $\text{rmq}$  求得。可以说，区间  $[i,i+1]$  依赖区间  $[l,l+1], [l+1,l+2], \dots, [r-1,r]$

把区间  $[i,i+1]$  抽象为一个节点，对于每一个  $i$  求出其依赖区间，连边建图缩点后转化成 DAG 上的动态规划问题，从叶节点向上不断更新实际依赖区间即可。

由于建图是  $n^2$  级别的，这里建图又是区间连边，需要用线段树优化建图。

## J Justified Jungle

solved by nikkukun

### 题目描述

给一棵树，求所有的正整数  $k$  满足：可以把树切成若干个大小为  $k$  的连通块  $n < 1000000$

### 解题思路

$O(\sqrt{n})$  枚举每个因数，对于一个因数  $k$  判断是否合法。

判断合法的根据：将所有子树大小为  $k$  的倍数的节点，与父亲的连边割断，如果恰好割断了  $\frac{n}{k}$

-1\$条边，那么合法，否则不合法。

## K Kitchen Knobs

upsolved by qxforever

### 题目描述

给 $n$ 个长度为 $l$ 的数字，一次操作表示将某个连续区间内所有数字轮换任意次数。要让所有数字最终处于最大值，问最少操作数。

### 解题思路

首先排掉 $l$ 位均相同的数字，其他数字的轮换必然有且只有一个最大值，问题转化为“有 $n$ 个 $[0,6]$ 之间的数，一次操作是一个区间加某个值模 $7$ ，问多少步能够使得所有数字变为 $0$ ”。

将数组差分，问题变为：对于差分数组，一次操作为某个数 $+x$ 另一个数 $-x$ 问多少步能使所有数字变为 $7$ 的倍数。

那么 $1,6;2,5;3,4$ 分别配对，最后只剩下三种未配对的情况 $dp$ 一下求出最多可以配成多少对即可。

## L Lunar Landscape

solved by Potassium

### 题目描述

给定平面上一堆平行于坐标轴或者成 $45^\circ$ 度的正方形，求总覆盖面积。

### 解题思路

对两种情况分别做二维前缀和，对每个格子的四部分用二进制表示状态即可。

From:

<https://wiki.cvbbacm.com/> - CVBB ACM Team

Permanent link:

[https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:i\\_dont\\_know\\_png:cerc2017&rev=1588743820](https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:i_dont_know_png:cerc2017&rev=1588743820)

Last update: 2020/05/06 13:43