

比赛记录

D

n 位长的十进制数字，在其中可以任意插入分割线，分割后，要使每一段不为空，并且可以整除 m 合法分割的方案数。

若 A, B 都能被 m 整除，则 $AB = A * 100... + B$ 一定能被 m 整除。

求有多少个前缀能恰好被 m 整除。若有 m 个 (不包括末尾)，结果就是 2^m 。(相当于枚举每个位置分割 or 不分割)。

A

签到题。就是只需要将点阵换为数，数换为点阵即可。完完全全是模拟。没什么好说的。

K

有两个队的骑士 1 到 n 和 $n+1$ 到 $2n$ 每个骑士只能互相攻击对手队的一个骑士 $\square$ kernel 的意思是在这个 kernel 里的骑士不会互相攻击，在 kernel 外的骑士被 kernel 里的骑士攻击。

现在告诉你所有骑士攻击的骑士，求一个 kernel $\square$

没人攻击的骑士一定在 kernel 里，把没人攻击的加入队列，然后被他攻击的骑士一定在 kernel 外。

kernel 外的骑士的攻击无效，因为如果一个骑士如果只被外面的骑士攻击，他就是 kernel 里的。

于是被外面的骑士攻击的骑士的被攻击次数 -1 ，如果被攻击次数为 0 了就加入队列。

WA 是由于一些愚蠢的手误呜呜 $\sim$ RE 是因为数组大小没有乘 2

B

一个非常简单的暴力签到题 $\square$ T 掉的原因是没有使用输出优化 QAQ

F

又是奇妙的数学题。

题意是一个二元函数。递推式是 $F[i, j] = a * F[i, j-1] + b * F[i-1, j] + c$ $\square$ 递推边界是 $F[k, 1] = l_k$ 和 $F[1, k] = t_k$ $\square$ 给定 $\{l_k\}, \{t_k\}, a, b, c$ 以及一个正整数 $n (2 \leq n \leq 200000)$ $\square$ 求 $F[n, n]$ $\square$

这个题经过简单的递推迭代之后，可以轻松的得出 l_k, t_k 前的系数，但是常数项却很难得

出 l_k, t_k 前的系数分别是 $\sum_{\forall k(2 \leq k \leq n-1)} C_{2n-k-2}^{n-k} a^{n-1} b^{n-k}$, $\sum_{k=1} C_{2n-k-2}^{n-k} b^{n-1} a^{n-k}$ $k=1$ 时系数为0, $k=n$ 时, 上面公式中 $2n-k-2$ 改成 $n-1$

而c的系数比较复杂, 前半段是 $(a+b)$ 的某个次幂, 后半段则是可以整体递推的, 具体结果有点复杂就不写了, 贴一个[代码](#)吧

J

你被雇佣升级一个旧果汁加工厂的橙汁运输系统。系统有管道和节点构成。每条管道都是双向的, 且每条管道的流量都是1升每秒。管道可能连接节点, 每个节点最多可以连接3条管道。节点的流量是无限的。节点用整数1到n来表示。在升级系统之前, 你需要对现有系统进行分析。对于两个不同节点s和t, $s \rightarrow t$ 的流量被定义为: 当s为源点, t为汇点, 从s能流向t的最大流量。以下面的第一组样例数据为例, 1-6的流量为3, 1-2的流量为2。计算每一对满足 $a < b$ 的节点a-b的流量的和。

答案显然只有0 1 2 3

0: 分别处理联通块

1: 同个联通块的不同边双

2和3考虑依次删掉每一条边, 再求边双, 如果两个点不论删除哪一条边, 都一直在同一个边双联通分量里, 那么流量就为3, 否则为2

每次把边双联通分量的id hash起来就可以了。

E

题意是说给了一个n个点m条边的有权无向简单图。有q次询问, 每次询问给定一个权值下限p, 所有权值小于p的边删除, 孤点删除, 然后进行缩点: 如果一个点的度数恰好为2且没有自环, 就把这个点删除, 然后把连接这个点的两条边连起来。(所有询问独立) 对于每个询问求剩下几个点几条边。
 $n, m, q, p \leq 3 \cdot 10^5$ 边权可能为0

自然而然的, 我们考虑建立线段树(树状数组也够了)。线段树范围从0到 $3e5$ 注意一定要有0! 我WA的那一次就是因为没有注意到0!

一棵树road_tree 对于每条边, 若边权为w 令 $[0, w] + 1$ 表示这条边在这些范围内被保留

一棵树point_tree 对于每个点, 若与其相连的边最大的边边权为w1 令 $[0, w1] + 1$ 表示这个点再这些范围内不会因为是孤点被删除

一棵树delete_tree 对于每个点, 若与其相连的边的次大边权为w2 第三大边权为w3 令 $[w3+1, w2]$ 表示这个时候这个点会因缩点操作被删除(同时会少一条边)

但是, 这个做法无法处理自环的情况。虽然原图是简单图, 但是如果图中存在简单环, 就会缩点形成一个孤点和这个孤点的自环, 根据delete_tree 这个点应该会被删除, 但是这不符合题意。

在经过与队友的讨论以后, 我发现, 将边从大到小排序, 逐步加边, 然后启发式合并, 可以在 $O(n \log n)$ 的复杂度内判断出何时产生简单环。因此再建一棵树loop_tree即可。

代码贴一个[链接](#)吧

From:

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

Permanent link:

https://wiki.cvbbacm.com/doku.php?id=2022-2023:teams:loaf_on_contest:front_page:st2



Last update: **2022/08/31 22:08**