Quark Round 1

A

题意

给定 n,m 求满足 $i+j=n$ 且 $\lfloor i/j\rfloor+\lceil j/i\rceil=m$ 的正整数对 (i,j) 的对数。

有 10^5 组数据，$n,m\leq 10^7$。

题解

将 $j=n-i$ 带入第二个式子后发现是先减后增的。在极值点两侧分别二分即可。

或者分别讨论 $i<j$ 以及 $i\geq j$ 的情况，最后推出式子 $\lfloor \frac{n-1}{m} \rfloor-\lfloor \frac{n-1}{m+1}\rfloor+\lfloor \frac{n}{m} \rfloor-\lfloor \frac{n}{m+1} \rfloor$。

B

题意

给定一个 n 个点 m 条边的边带权无向图，在 0 时刻你在点 1 上。

假设当前是 t 时刻，你在点 v 上，你可以选择两种操作：

- 仍停留在点 v 上，操作后到 $t+1$ 时刻。
- 选择一条边 (a,b,w) 满足 $a=v$ 或 $b=v$，则你到这条边连接的另一个点上，操作后到 $t+w$ 时刻。

有 k 条信息，(t_i,v_i) 表示在 t_i 时刻，点 v_i 上会出现一只猪。如果这时你在这个点上，则你抓到了这只猪。

求最多能抓多少只猪，$n\leq 200$，$k\leq 5000$，$t\leq 10^9$。

解题思路

感觉和这个题好像。

先跑一遍 floyd。

对时间排序。设 $f[i]$ 表示考虑前 i 条信息最多抓到的数量。枚举之后的信息判断转移即可。时间复杂度 $O(n^3+k^2)$。

可以做一个小的优化。枚举之后的信息可以改为枚举点对应的信息，复杂度是 $O(n^3+n\times k)$。

C

题意

给一颗 n 个点有点权（可能为负）的树，加一条不存在的边（不能为自环），使得得到的基环树上所有点的深度与该点点权乘积之和最大。即最大化 $\sum w_i\times dep_i$ ，dep 为到基环树的最短距离， $n\leq10^6$

解题思路

换根dp，待补

D

题意

有 n 个空字符串。对其进行 q 次以下操作，设当前为第 i 次操作：

- `1 l r`，将编号在 $[l,r]$ 范围内的字符串末尾添加字符 i ；
- `2 L R`，求 $[L,R]$ 范围内所有字符串的最长公共子序列长度。

$n,q\leq 10^5$。

解题思路

对于每次操作一，考虑对之后的询问的贡献。由于每次添加的字符是不会重复的，发现当两种操作满足 $l\leq L\leq R \leq r$ 时，会对 $[L,R]$ 的 LCS 产生 1 的贡献。于是问题变成了一个二维数点问题，统计左端点在 $[1,L] $ 中，右端点在 $[R,n]$ 的操作一的个数。树状数组套线段树解决。

E

题意

有 n 个二元组 (a_i,b_i)，当 $b_i\leq0$ 时不可操作。可以执行以下两种操作

- 对所有可操作的二元组，$b_i=b_i-a_i$ ，花费 p ；
- 对所有可操作的二元组，交换 a_i,b_i ，花费 q ；

求使所有二元组均不可操作的最小花费， $n\leq 10^5$ ，$a,b\leq 10^7$ 。

解题思路

将 n 个二元组看作二维平面的向量。

假设经过一些操作后，二元组变为 (x,y) 若该二元组可以操作，则有 $x>0$ $y>0$ 这里 x,y 为关于 a,b 的一次多项式 $x>0$ $y>0$ 对应平面上的两条直线所夹的区域。故在经过一些操作之后仍可操作的向量，被夹在两条过原点的直线之间。若所有向量都不可操作，那么没有任何向量夹在这两条直线之间。将 n 个二元组极角排序后，枚举相邻的二元组，表示最终的两条直线从他们之间穿过。问题便转化为两个向量的问题。

假设当前枚举的向量为 $\vec{v_1},\vec{v_2}$ 前者极角序更小。

1. 若 $\vec{v_1},\vec{v_2}$ 都位于直线 $y=x$ 上方，则执行操作一。
2. 若 $\vec{v_1},\vec{v_2}$ 两个向量都位于直线 $y=x$ 下方，则执行操作二。
3. 若 $\vec{v_1}$ 为于 $y=x$ 上方 $\vec{v_2}$ 位于 $y=x$ 下方，则取以下两者花费最小的：
    - 执行一次操作一，使 $\vec{v_2}$ 下方的向量变为不可操作。之后执行操作二，再执行若干次操作一，使 $\vec{v_1}$ 上方的向量不可操作。
    - 先执行一次操作二，再执行一次操作一，使 $\vec{v_1}$ 上方的向量不可操作；之后执行一次操作二，再执行若干次操作一，使 $\vec{v_2}$ 下方的向量不可操作。

其中 1 和 2 为类似辗转相除的过程，3 只会执行一次。时间复杂度为 $O(n\log\min(a,b))$

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

Permanent link:
https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:i_dont_know_png:qxforever:qkoi_r1

Last update: **2020/05/10 18:26**