

GXUACM2022GreenHand01 题解

A K-divisible Sum (模拟)

给定两个正整数 $n, k (1 \leq n, k \leq 10^9)$, 问由正整数组成 的长度为 n 的数组中最大的元素最小为多少时数组的和可以被 k 整除

可以分为 3 种情况

- 第一种情况, 当 $k \geq n$ 时数组中元素至少为 k/n , 如果 $k \% n \neq 0$ 则最大元素还要 $+1 (a_{ij} \leq k/n + (k \% n \neq 0))$
- 第二种情况就是 $n > k$ 且 $n \% k == 0$, 此时数组中每个元素都为 1 即可 ($a_{ij} == 1$)
- 第三种情况为 $n > k$ 且 $n \% k \neq 0$, 此时数组中元素至少存在 2 ($a_{ij} \leq 2$)

B Morning Jogging (贪心)

给定一个 n 行 m 列的数组 ($1 \leq n, m \leq 100, 1 \leq x_i \leq 10^9$), 问对每行元素如何排列可以使每列中最小元素之和最小

只需对数组中所有元素进行按从小到大排列, 然后将排好序后最小的 m 个元素分配在其原来所在行的对应 m 列就可以实现每列最小元素之和最小, 最后再将剩下的元素任意次序放回对应行中输出即可

C Di-visible Confusion (LCM 最小公倍数)

给定一个长度为 n 的数组 ($1 \leq n \leq 1e5, 1 \leq a_i \leq 10^9, 1 \leq i \leq n$), 当 a_i 不能被 $i+1$ 整除时, 可以删去 a_i , 每次删除会改变剩余元素的下标, 问能否删除所有元素

最暴力的做法可以直接从 $[2, i+1]$ 检验每个数能否不被范围内某个下标整除, 但是考虑到 $1 \leq a_i \leq 10^9$, 因为 $13! > 10^9$, 所以实际上遍历的次数会很少;

或者根据 lcm(能同时整除 $[1, 26]$ 的数字大于 10^9), 因此我们也可以直接遍历 $[2, \min(26, i+1)]$ 来得到答案

D String Modification (字符串)

给定一个长度为 n 的字符串 $S (1 \leq n \leq 5 * 10^4)$, 选定一个数字 $k (1 \leq k \leq n)$ 来对字符串进行处理, 问处理后能得到的最小字典序的字符串和 k

例如对 $S = \text{"qwer"} , k = 2$ 进行处理

- $S = \text{"wqer"}$ (长度为 k 的第一个子串)
- $S = \text{"weqr"}$ (长度为 k 的第二个子串)
- $S = \text{"werq"}$ (长度为 k 的最后一个子串) 处理结束之后得到的 S' 为 "werq"

结论是观察出当 $n - k$ 即从 $k-1$ 到结尾的字符串长度为奇数时前面的字符串要进行反转, 偶数时不用反转

最终需要判断

$$n - k$$

奇数得到的字符串为 `s.substr(k-1, n - k) + reverse(s.substr(0, k))`

偶数得到的字符串为 `s.substr(k-1, n - k) + s.substr(0, k)`

E Mocha and Diana (Easy Version) (并查集)

给定共有 n 个节点和边数分别为 m_1, m_2 的两个森林(没有环的无向图)($1 \leq n \leq 1000, 0 \leq m_1, m_2 \leq n$), 并给出对应的边 $u, v (1 \leq u, v \leq n, u \neq v)$ 此时可以对两个森林进行加边, 但是加边操作要满足

- 加完边之后两个无向图仍然是森林, 即加完边之后不能出现环
- 加边操作是同步的即一个森林加了边, 另一个森林就要加上同样的边

问最多能加多少条边

依据题意可以得知是并查集, 对两个森林都创建并查集, 因为($1 \leq n \leq 1000$), 可以直接暴力 $O(n^2)$ 遍历每一条边能否同时加到两个森林中且满足条件即可

F Weights Assignment For Tree Edges (贪心)

给定两个长度为 n 的数组 $b, p (1 \leq n \leq 2 * 10^5)$, b_i 代表 i 的父亲节点为 b_i , p_i 代表到 i 到根节点的距离排行, 问 p 所代表的距离排行能否成立, 如果可以成立, 则输出 i 到 i 的父亲节点的距离 dis ; 否则输出 -1

由 b 数组可以知道若 $i == b_i$ 则 i 为根节点

由 p 数组可以知道 i 到根节点的距离排行

因此我们只要判断每个点到父亲的节点是否小于到根节点的距离排行就可以判断出 p 数组是否成立, 即如果我离我的祖宗比我的父亲更近不成立 此时输出 -1

当 p 数组成立时我们只需要输出每个点到父亲的距离即可, 我到父亲的距离 = 我到祖宗的距离 - 我父亲到祖宗的距离

```
dis = [0] * 200001          # 距离数组
for i in range(n):
    dis[p[i]] = i
ok = 1
for i in range(n):
    if dis[i] < dis[b[i]]: # dis[i]: i到根节点的距离
                           # dis[b[i]]: i的父亲节点到根节点的距离
        ok = 0
        break
if not ok:
    print(-1)
else:
    for i in range(n):
        print(dis[i] - dis[b[i]])
```

G Vasya and Chess (博弈)

给定一个数字 $n \times n$ 的国际象棋棋盘 ($2 \leq n \leq 109$), 其中白皇后位于 $(1, 1)$, 黑皇后位于 $(1, n)$, 其他全部点上有不属于两边的绿色棋子。皇后每次走要么吃绿棋要么吃对面皇后, 只能斜, 横, 竖走。白皇后先手, 黑皇后后手, 问双方都发挥最好谁能获胜, 若是先手获胜要输出 白皇后先手

讨论:

- 当 $n = 2$ 时, 白胜(白皇后只需要一步就能吃掉黑皇后)
- 当 $n = 3$ 时, 黑胜(白皇后一往右走就会一步就被黑皇后吃掉, 要是不往右走斜着走也会一步就被黑皇后吃掉, 要是竖着走, 黑皇后就跟着镜像走最终白皇后要么斜着走, 往右走, 黑皇后只要镜像着白皇后走就一定能吃掉白皇后)
- 当 $n = 4$ 时, 白胜(白皇后只需要一开始就往右走就会变成 $n = 3$ 黑皇后先手的情况, 此时胜方就会互换)
-

总结

- 当 $n = 2$ 时, 白皇后走 $(1, 2)$ 就可以直接吃掉黑皇后
- 当 $n > 2 \ \&\& \ n \% 2 == 0$ 时, 白皇后走 $(1, 2)$ 就能将情况变为之前 $n - 1$ 的情况, 互换胜方
- 其他情况, 黑皇后镜像走必胜

```
if n % 2 == 0:
    print("white")
    print("1 2")
else:
    print("black")
```