GXUACM2022GreenHand02 题解

A Odd Divisor(数论, 模拟)

> 给定一个正整数**n**(2 $\le$ **n** $\le 10^{14}$), 问**n**是否存在大于1的奇数因子(即问**n**是否存在除1以外的奇数能够整除**n**), 时间复杂度为O(log_2n)

```
# 当n能整除2时不断整除
# 只有当n==奇数 时退出
# 最后判断n是否=1即可
while n % 2 == 0:
    n /= 2
if n == 1:
    print("NO")
else:
    print("YES")
```

B Array Reodering(贪心)

> 给定一个长度为**n**的数组a[](2 $\le$ **n** $\le$ 2000, 1 $\le a_i \le 10^5$), 问对数组进行任意排序能找出的i, j(*1* $\le i \le j$ $\le n$)且 **gcd(2a_i, a_j) > *1*** 的组数有多少组

思路

可以观察出当a_i为偶数时, 只需将其放到数组最后就可以使得任意 k < i, *gcd(2a_k, a_i)* $\ge$ *2*

又当a_i, a_j为奇数时, *gcd(2a_i, a_j)* == *gcd(a_i, a_j)*, 因此只需将剩下的所有奇数暴力查询即可, 时间复杂度为O(n^2)

C Sum of Cubes(打表)

> 给定一个数**n**(2 $\le$ **n** $\le 10^{12}$), 问是否存在正整数a, b, 使得a^3 + b^3 == n成立

思路

打表, 因为n的范围是10^{12}, 因此a, b的值不会超过10^4, 只需要打表记忆i^3 (1 $\le$ i $\le 10^4$), 随便对i进行枚举即可, 时间复杂度为O($n^{1/3}$)

```
for i in range(1, pow(n, 1/3) + 1):
    if map[n - pow(i, 3)] == 1:
        print("YES")
        return
print("NO")
```

D Mocha and Hiking(DFS模板题)

> 给定一个数n($1 \le$ **n** $\le 10^4$), 表示有n + 1个点和2n - 1条边, 其中有n - 1条边为 **i** 到 **i - 1**($1 \le$ **i** $\le$ n - 1), 以及一个长度为n的数组, 代表n条边, 其中当a_i = 0 时, 代表有一条从 i 到 n + 1 的边, 当a_i = 1时, 代表有一条从 n + 1 到 i 的边($0 \le a_i \le 1$, $1 \le$ **i** $\le$ n) 时间复杂度(不会算)

```
res = []
seen = []
find = False
def dfs(now: int) {
    if len(res) == n + 1:
        print(res)
        return
    for i in G[now]:
        if i not in seen:
            res.append(i)
            seen.append(i)
            dfs(i)
            res.pop(i)
            seen.pop(i)
}
for i in range(1, n + 2):
    res.append(i)
    seen.append(i)
    dfs(i)
    res.clear()
    seen.clear()
    if find:
        return
print("-1")
```

F Pride(贪心 模拟)

> 给定一个整数n($1 \le$ n $\le$ 2000), 以及一个长度为n的数组a($1 \le a_i \le 10^9$) 可以执行以下操作任意次

- 从a中选定2个相邻的元素a_i、 a_j, 并将其中任意一个替换成gcd(a_i, a_j)

> 问使a中所有元素都变成1所需的最小操作数 由于n范围很小因此O(n^2)完全可以实现

思路

分类讨论特判

- 当整个数组的gcd != 1时, 无法操作数组全部变成1
- 当数组中存在1时, 由于1与任何数的gcd都等于1, 因此最小操作数为n - cnt1
- 当数组中不存在1时, 可以贪心求出整个数组最短gcd = 1的子数组长度, 而后将其变为第二种情况来讨论即可

G Multiples of Length(思维 构造)

> 给定一个长度为n的数组a(1 $\leq$ n $\leq$ 10^5, -10^9 $\leq a_i \leq 10^9$), 可以进行如下操作

- 选择一段连续子数组, 将其中元素加上子数组长度len的任意整数 倍(相加的倍数可以不同)

> 问在固定三次操作下如何将数组中所有元素变为0, 时间复杂度O(n)

思路

由于是固定要三次操作, 所以对任意数组的操作都应该是一样的

设数组中元素为*x*, 则若选取长度为n - 1的数组, 将其中所有元素扩大n - 1倍, 则操作之后的元素应该为n * x, 考虑到n - 1可能等于0, 因此可以对长度为1的数组进行特判

- 当n = 1时, 任何整数可以是1的倍数, 因此当a_0 = x时, 只需要在第一步就加上*-x*即可将整个数组变为0, 剩下的两个操作只需要加上0即可
- 当n != 1时, 第一步操作将前n - 1项全部加上n - 1倍, 这样前n - 1个元素都变成了原来的n倍, 第二步将最后一个元素加上n - 1倍, 由于最后一项的长度为1, 任何整数都可以是1的倍数, 因此操作合法; 进行完前两步之后整个数组中的元素都是n的倍数, 因此第三次操作只需要选择整个数组并减去其本身即可

H Engineer Artem(构造 奇偶性 贪心 模拟?)

> 给定一个**n x m**(1 $\leq$ n $\leq$ 100, 1 $\leq$ m $\leq$ 100, $1 \leq a_{ij} \leq 10^9$)的矩阵, 可以对矩阵中的任意元素进行0次或者1次操作(使其加1), 问如何将这个矩阵变成任意相邻元素都不相等的矩阵(即一个元素不能与其上下左右若存在的元素相等) 时间复杂度O(n * m)

思路1

要使矩阵中相邻元素都不同, 最简单的方法只需要其奇偶性不同即可, 我们可以规定奇行奇列的元素为奇数, 奇行偶列和偶行奇列的元素为偶数, 偶行偶列的元素为奇数, 这样就能保证元素的上下左右都为奇数

```
for i in range(1, n + 1):
    for j in range(1, m + 1):
        if i % 2 == j % 2:        # 奇行奇列 偶行偶列
            if a[i][j] % 2 == 0: # 如果本来为偶数
                a[i][j] += 1      # 则变成奇数
            else:                 # 否则不变
                None
        else:                     # 奇行偶列 偶行奇列
            if a[i][j] % 2 != 0: # 如果本来为奇数
                a[i][j] += 1      # 则变成偶数
            else:                 # 否则不变
                None
```