

# 吉林省赛题解

哈尔滨工业大学

2020 年 9 月 27 号

# Chord

- 建立起音和整数的映射即可。
- 先搞一个初始映射：C 映射为 0，C# 映射为 1，...，B 映射为 11。
- 然后依次读入三个音。 $N_1$  的初始映射值就作为其真实映射值。如果  $N_2$  的初始映射值比  $N_1$  小，就加上 12，作为其真实映射值；如果  $N_3$  的初始映射值比  $N_2$  的真实映射值小，就不断地加上 12（其实至多加 2 次），直到超过  $N_2$  的真实映射值
- 最后根据相邻两个音所对应的真实映射值的差进行判断即可。

## Problem Select

---

- 把字符串末尾的数提取出来，从小到大排序，取前  $k$  个输出即可

# String Game

- 抱歉数据范围出锅
- 这个题是让求 B 字符串在 A 中出现多少次，可以不连续。
- 考虑计数 dp 做，思路和最长公共子串类似。
- 设状态  $f[i][j]$  为 B 的前 j 个字符在 A 的前 i 个字符中出现多少次。
- 可以得到状态转移方程

$$f[i][j] = f[i-1][j] + [A[i] == B[j]] \times f[i-1][j-1]$$

- 可以用滚动数组优化

# Trie

---

- 建立 AC 自动机，一个结点的后缀对应的结点为 fail 树上的祖先结点。
- 所以这其实是一个子树修改，单点查询的问题。
- 每次修改时修改的子树是所有结点对应子树的并。
- 使用 DFS 序，修改就变成的求区间并，之后区间修改。

# Shorten The Array

序列可分为下列三种情况：

- 情况 1: 最小值  $\min$  只有一个。利用此最小值向左右任意  $\%$  其他数，就可以把序列缩减为 1
- 情况 2: 最小值  $\min$  有  $N$  个，但是存在一个数  $X$  不是最小值的倍数。可以利用  $X$  最近的一个  $\min$ ，让该  $\min \%$  别的数靠近  $X$ ，利用  $X \% \min$  一定小于  $\min$ ，可以构造出一个单一最小值，转化为情况 1，答案为 1。
- 情况 3: 最小值  $\min$  有  $N$  个，且其余数均为  $\min$  的倍数。只能让  $\min$  把其余大数都删去，然后  $\min$  之间两两抵消归零。答案为  $\lceil \frac{N}{2} \rceil$ 。

# Queue

题目大意：给一个数列，和一个交换序列，每次交换两个位置上的数，求按给定序列交换过程中数列逆序对最少是多少。

- 先跑一遍逆序对：树状数组或者归并排序均可。
- 注意到每次交换只有  $l$  到  $r$  之间的数会对逆序对产生影响。
- 注意  $r - l \leq 100$ , 对  $10^3$  次询问，每次直接  $O(r - l)$  暴力统计逆序对变化情况即可。

# Matrix

- 设  $x$  的约数个数为  $d_x$ ，则对于元素  $a_{ij}$ ，它会被翻转  $d_i \times d_j$  次。
- 这意味着，当且仅当  $i$  和  $j$  的约数个数均为奇数时， $a_{ij}$  最终才取 1。由数论知识易知，一个正整数的约数个数为奇数，当且仅当它是完全平方数。
- 因此，有  $n$  行  $m$  列矩阵的答案就是行列下标均为完全平方数的元素个数，即  $\lfloor \sqrt{n} \rfloor \times \lfloor \sqrt{m} \rfloor$ 。



# Curious

- 很裸的莫反。。
- 用  $g(d)$  表示问题，那么

$$g(d) = \sum_{i=1}^n \sum_{j=1}^n [(a_i, a_j) = d]$$

- 非常套路地，

$$f(d) = \sum_{d|x} g(x) = \sum_{d|x} \sum_{i=1}^n \sum_{j=1}^n [(a_i, a_j) = x] = \sum_{i=1}^n \sum_{j=1}^n \sum_{d|x} [(a_i, a_j) = x]$$

- 会注意到， $\sum_{d|x} [(a_i, a_j) = x] = [(d|a_i) \wedge (d|a_j)]$ （显然  $a, b$  的 gcd 是  $d$  的倍数当且仅当  $a, b$  都是  $d$  的倍数）。

# Curious

- 那么  $f(d) = \sum_{i=1}^n \sum_{j=1}^n [(d|a_i) \wedge (d|a_j)]$ 。
- 显然对于每一个  $x$ ，需要处理出满足  $x|a_i$  的数量，设为  $h(x)$ 。
- $f(d) = h^2(x)$ ，那么  $g(d) = \sum_{d|x} f(x) \mu(\frac{x}{d})$ 。
- 直接枚举倍数计算  $h(x)$  与  $g(x)$  即可。
- $n(1 + \frac{1}{2} + \frac{1}{3} + \dots \frac{1}{n}) = O(n \log n)$ ，总复杂度  $O(n \log n)$ 。

# World Tree

- 题面翻译为人话：对有根树钦定一个 **dfs** 序，排列后使  $\sum_{i=1}^n a_i \sum_{j=i+1}^n b_j$  最大。
- 考虑如果是一个菊花图该怎么做：除了根可以任意钦定顺序。
- 考虑如果只有 2 个点，那么当 12 的顺序大于 21 的顺序，当且仅当  $a_1 b_2 > a_2 b_1$ ，即  $\frac{a_1}{b_1} > \frac{a_2}{b_2}$ 。
- 考虑一个序列中相邻的两个点，当  $\dots, i, i+1, \dots$  大于  $\dots, i+1, i, \dots$  时，会发现条件依然为  $a_i b_{i+1} > a_{i+1} b_i$ ，即  $\frac{a_i}{b_i} > \frac{a_{i+1}}{b_{i+1}}$ 。

# World Tree

- 也就是说，一个序列，只要任意两个相邻值不满足就可以更大，所以按照  $\frac{a_i}{b_i}$  排序后即是极大值也是最大值。如果不是菊花图，很容易发现对于一个点的几棵子树，子树内部的节点的排序不会影响这个子树与其他几棵子树排顺序时的答案。
- 所以对每一个子树按照  $\frac{\sum a_i}{\sum b_i}$  排序即可。
- 如果让  $b = 0$ ，应该可以卡一下有些直接写成  $\frac{a}{b}$  的人。但是我没有。



## Situation

- 这样直接搜索显然会超时, 我们可以加入一个记忆化。
- 将一个状态搜索的结果保存在数组中, 后续搜索到这个状态就直接调用。
- 这需要一个哈希函数来将状态映射到一个整数。
- 一个很简单的思路是: 将 Alice 的棋子看作 1, Bob 的棋子看作 2, 空位看作 0, 这样一个 3\*3 的棋盘可以视作一个 9 位三进制整数, 然后将 Alice 先手的状态和 Bob 先手的状态分别保存就可以了。

# Forager

- 对于每个发射点，从源点连接一条容量为 1，花费为 0 的边（当所有转向石方向固定后，每个发射点的射线最多到一个宝石）。
- 对于每个宝石，连接一条容量 1，花费  $-z_i$  的边到汇点。
- 对于每个转向石，将创建一个额外的结点用来，将它们直接连接一条容量为 1，花费为 0 的边来限制只有一个方向。之后从额外的点来建立转向后的对应的边。
- 建图时，显然空地和墙是无额外收益的，可以跳过它们直接和宝石和转向石连边。
- 之后跑一个最小费用最大流即可获得答案。

# Swimmer

---

- 每个人是来回游泳，所以一次来回是一个周期。
- 速度乘时间取模后根据剩余时间计算位置即可。
- 注意开 long long。



# Upanishad

- 因为发现原题所以放在热身赛。
- 设  $A$  为“区间内所有元素的异或和”， $B$  为“区间内所有**不同**元素的异或和”。
- 询问的答案转化为  $A \text{ XOR } B$ 。
- $A$  通过前缀异或和即可求解。
- $B$  通过莫队或树状数组求解即可。
- 出题人卡掉了莫队。

# Upanishad

树状数组的求解思路：

- 把所有询问按照右端点从小到大排序，然后依次考虑序列中的每个元素。
- 维护  $last[val]$  表示值  $val$  上一次在序列中出现的位置。
- 假设当前在位置  $r$ ，则询问  $[l, r]$  的答案就是  $\sum_{i=l}^r cnt[i]$ ，其中  $cnt[i] = 1$  当且仅当  $last[a_i] = i$ 。
- 走到下一个位置  $r' = r + 1$  时，需要如下赋值操作：  
 $cnt[last[a_{r'}]] = 0, cnt[r'] = 1, last[a_{r'}] = r'$ 。
- $cnt$  和  $last$  都是容易维护的，再用树状数组维护  $cnt$ ，即可在  $O(\log n)$  时间内得到其区间和。

# Expressway

---

- 因为发现原题所以放在热身赛。
- 显然，只有删除最短路上的边才会对最短路产生影响。
- 考虑最短路径树，将图中的点根据其所连通的最近最短路上的点分层。
- 如果一条边跨层了，那么它就会对一些最短路上的边作出贡献。所以只要做多次区间修改即可。
- 注意：最短路使用 Dijkstra 算法，出题人卡了自己会写的 SPFA。SPFA 已死