华中科技大学 2023 年新生赛题解

1 A 简单的加法乘法计算题

设 f_i 表示从 1 到 i 的最少操作次数。由于乘法操作的集合很小，可以直接暴力枚举转移；而加法操作只能从 $f_j, j \in [i-n, i-1]$ 转移，显然维护这段区间的最小值即可，但由于 n 较大，不能使用一般的数据结构维护，不难发现使用单调队列优化 DP 即可。

时间复杂度 $O(ym)$。

2 B 逆 KMP

可以发现，在 $a_i \neq 0$ 时，这实际上提供了一个区间相等的条件，这启发我们使用并查集算法。

但普通的并查集只支持维护单点相等，而要把每个区间相等都拆开做完，复杂度肯定是爆炸的，因此，我们要对此做出优化。

我们观察到上述做法会超时的原因是重复的合并太多，考虑减少合并次数。我们又发现区间相等这一条件是可重复贡献的（这也是普通并查集复杂度爆炸的原因），所以，我们考虑用st表维护并查集。

我们记录 $\lceil \log n \rceil$个并查集，第 k 个并查集维护哪些长度为 2^k 的区间应该相等。第 k 层并查集的结果可以下传到第 $k-1$ 层，传到第 0 层时就可以统计答案了。

时空复杂度均为 $O(n \log n)$。

3 C 排列排序问题

考虑先把所有差值的绝对值不为 1 的位置断开（显然这些位置是必须断开的），之后每一段也一定能够组成按顺序递增的排列，因此记录一下断开的次数即可。

时间复杂度 $O(n)$。

4 D 网格染色

后者存在使自己必胜的策略，其中一种简单易懂的策略是：将某个对角线作为对称轴，无论先手如何涂色，后手根据对称轴左右涂色即可。

时间复杂度 $O(1)$。

5 E 序列配对

根据题目性质不难发现将配对点连边后会有若干个环，如果将 -1 的节点向 $+1$ 的节点连有向边，则这些操作等价于给每条边定向，那么一个节点对最终元素平方和的贡献等价于出度和入度的差值的平方。显然这个差值（的绝对值）只能是 0 或 2，且只有为 2 时才是有意义的，所以考虑有多少种方案可以使得恰好 $\frac{k}{4}$ 个节点的差值为 2。

首先单独考虑一个环，容易发现差值为 2 的节点只能是偶数个，如果已经确定了这些节点的位置，能且仅能构造出 2 种方法（随便选一个差值为 2 的节点，考虑要么出度为 2 要么入度为 2，此时其他边的方向都是唯一确定的）。因此可以很容易求出一个环上的生成函数为 $\sum_{0\le i\le \lfloor \frac{L}{2} \rfloor} 2\binom{L}{2i}x^{2i}$，其中 L 为环的长度。最后用启发式合并 NTT 将每个环的生成函数相乘即可。

时间复杂度 $O(n\log^2 n)$。

6 F 新取模运算

将每个数字根据其因子内 p 的最高次幂进行分组，即 $1,2,3\ldots,p-1,p+1,p+2,\ldots$ 为一组，$p,2p,3p,\ldots,(p-1)p,(p+1)p,(p+2)p$ 为一组，以此类推。不难发现每一组内对 p 的新取模运算的结果呈现 $1,2,\ldots,p-1$ 循环出现，因为预处理 1 到 $p-1$ 的阶乘即可。

使用快速幂的时间复杂度是 $O(p+T\log^2 n)$，用威尔逊定理可以优化至 $O(p+T\log n)$。

7 G 广义线段树

可以注意到，修改 A_i 的值只会影响对应叶节点到根节点路径上所有节点的权值。直接模拟是 $O(\sum_{i=1}^{n} h_i)=O(n^2)$ 的（h_i 为点 i 深度），因而无法通过本题。

重要结论：由于线段树的编号性质，**修改 A_i 的顺序恰好和线段树的先序遍历访问叶节点的顺序相同**。这对本题的广义线段树同样适用。

结合上述两点，我们可以用先序遍历解决本问题。

首先，建立**静态**广义线段树 `aseg` 和 `bseg`，分别维护由原序列 $\{A_n\},\{B_n\}$ 生成的广义线段树的各节点权值。我们在之后不修改 `aseg` 和 `bseg`。

在先序遍历时，我们维护当前节点到根节点路径上（包括自身）所有节点权值之和 `sum` 和答案 `ans`。

- 进入节点 `p` 时，有 `sum += aseg[p]`。
- 到达叶节点进行修改时，有 `new_sum = sum * b[i]` 和 `new_ans = ans - sum + newsum`。计算后，输出 `new_ans` 作为答案。
- 退出节点 `p` 时，有 `sum -= aseg[p] * bseg[p]`。
  - 这是因为退出节点 `p` 时，`p` 控制区间的所有叶节点的 `a[i]` 都已经访问过了（都乘上了 `b[i]`），因此当前点 `p` 的权值就是控制区间内所有 `a[i]` 和 `b[i]` 的乘积。

时间复杂度为 $O(n)$。

8 H 狭义线段树

题目给出的伪代码其实就是线段树的建成方式，~~大家应该都看出来了吧。~~

所以单个节点加上一个值就等价于区间加，询问就等价于区间查询。

对于2操作，容易得到一次2操作实际上是对一个连续区间进行一次区间加，所以求出范围内节点的最左左端点和最右右端点即可。单次复杂度为 $O(\log n)$。

对于1操作，解法很多，以下讲解两种。

解法1：我们把线段树的节点按层分类，对于每层节点（最深层除外），由于节点维护区间之间没有空隙，单次询问就相当于一个区间加，可以直接维护。而对于最深层，由于节点维护区间之间存在空隙，因此可以单独将其视为一个序列，独立维护其区间和，在查询时将本层贡献与其他贡献相合并，即可得到答案，单次复杂度 $O(\log^2 n)$。

解法2：按编号从小到大依次考虑本次操作的节点。如果当前节点的整个子树都在本次操作中，就在该节点执行一次子树加，然后跳过整个子树，否则执行一次区间加，然后考虑下一个节点。（子树加可以通过给每个叶子赋上等同于其深度的权值完成）。复杂度分析：对于每一层，至多存在一个节点，该节点的子树不全在修改区间中，因此单次操作复杂度为 $O(\log^2 n)$，精细实现可以达到 $O(\log n)$。

9 I Fujisaki 讨厌数学

令 $f(n) = x^n + x^{-n}$

有 $f(n) * f(1) = x^{n+1} + x^{n-1} + x^{-(n-1)} + x^{-(n+1)} = f(n+1) + f(n-1)$

所以有 $f(n+1) = f(1)f(n) - f(n-1)$

又因为 $f(1) = k$ ， $f(n+1) = kf(n) - f(n-1)$，矩阵快速幂即可。

时间复杂度 $O(\log n)$。

10 J 基因编辑

考虑枚举 S_k，则我们只需知道 S_k 的每个前缀/后缀分别可以和多少个串的前缀/后缀相等即可。这部分可以使用trie树完成。

但这样直接算的话会有重复，所以我们可以将前缀的匹配差分一下，这样每个串在取前缀时，就只会在最长匹配处被计算一次，就没有重复的问题了。

时空复杂度均为 $O(\sum |S|)$。

11 K 不定项选择题

不难发现操作次数至少是 $2^n - 1$，通过构造格雷码可以达到这个下界。

（即使不知道格雷码也可以猜样例。）

时间复杂度 $O(1)$。

12 L Azur Lane

显然每个喵箱越晚放入越好。

显然我们每天放的喵箱等级是一个单调不上升连续子序列。

考虑贪心。我们的策略是，开始每天只放一个喵箱，但是这可能会导致喵箱放不完，为了保证合法性，我们得在最后都尽可能地放喵箱，也就是说，最后几天每天放的喵箱等级一定是一个极大单调不上升连续子序列。

考虑选一个中间点 k，前 k 个喵箱在前 k 天中每天放一个，而后面每天都尽可能地放喵箱。

所以我们可以从后往前遍历数组，当遍历到第 k 个位置时，我们已经计算出了最后 $n-k$ 个喵箱组成的极大单调不上升连续子序列的个数 num 和这么放所最后 $n-k$ 个喵箱需要的钱数 $money$，而前 k 个喵箱每天放一个，所以我们就得到了天数为 $k+num$ 时的答案，最小可能的钱数为 $k(k+1)/2+num\cdot k+money$。

时间复杂度 $O(n)$。

13 M 近似递增序列

这个题做法好像很多， 有各种各样的乱搞。

不妨从大到小来依次生成这个序列， 考虑 $f(n,m,0/1)$ 表示接下来填数的乘积最多是 n， 最大可以填 m ， 有没有使用那个不满足严格递增限制的位置。

那么一方面可以填小于等于根号的数， 如果填 i ， 则可以转移到 $f(\lfloor\frac{n}{i}\rfloor,i,0/1)$ 。

还可以填大于根号的数， 如果填 i ， 则可以转移到 $f(\lfloor\frac{n}{i}\rfloor,min\{\lfloor\frac{n}{i}\rfloor,i\},0/1)$ 。由于 $i>\sqrt{n}$ ， 也就是转移到 $f(\lfloor\frac{n}{i}\rfloor,\lfloor\frac{n}{i}\rfloor,0/1)$ 。于是这一部分可以整除分块计算。

那么第一, 二维都是通过 n/i 得到或者由一个小于等于根号的数得到， 故总状态数是 $\sqrt{n}^2=n$ 级别的， 并且不难想象实际用到的位置会显著低于这个数字。

然后考虑第一种转移的总转移量， 考虑第一维度为 n时， 约为 $\sqrt{n}+\sqrt{\sqrt{n}}+\ldots$， 不妨约等于为 $\sqrt{n}$ ， 那么考虑所有 n 第一维转移量之和，约为 $\sqrt{n}+\sqrt{\frac{n}{2}}+\sqrt{\frac{n}{3}}+\ldots$ 显然不超过 n 。

再考虑第二种转移的转移量， 如果仅进行第二种转移， 那么复杂度同杜教筛的分析， 如果先进行第二种转移再进行第一种转移， 则会访问到仅进行第一种转移访问到的位置 ，我们采用记忆化的方法， 则这部分的复杂度不超过 $O(n^{\frac{3}{4}})$ 。如果先进行第一种转移再进行第二种转移， 由于最大数的限制， 这部分复杂度也不会超过 $O(n)$ 。

故本做法上限复杂度不会超过 $O(n)$ ， 在记忆化 $n\le 5000$ 的答案以后不精细实现在不开 O2 条件下可以在 $2s$ 内通过 $n=10^8$ 的数据， 且可以轻松扩展到有 k 个不合法位置的情况。

另一个有意思的事情是， 如果直接搜索有多少严格递增序列的乘积小于等于 n ， 只会搜出来大约 10^7 种情况， 可以基于这个性质暴力/乱搞直接通过这个题~~甚至可能比std快还短~~ 。

~~欢迎大家开发新做法薄纱std~~

14 N A+B problem

签到题，模拟一位 26 进制加法即可。

时间复杂度 $O(1)$。