题解

?

November 20, 2020

1 T1 出了个大阴间题

1.1 算法一

我会全排列！dfs 枚举全排列，按题意模拟。
时间复杂度 $O(n!)$，期望得分 40。

1.2 算法二

我会状压！$f[i][j]$ 表示已经合并了 i 的集合的前缀，其 a 为 j。
枚举集合，枚举 a，枚举下一个插入哪个元素，直接按题意模拟合并。
时间复杂度 $O(2^n n(\max_i a_i + n))$，对于一些部分分有 $a_i \le n$，期望得分 80。

1.3 算法三

对于 a_i 大怎么办？由于合并的时候，a_i 只会加 1，所以对于一个集合 S，其 a 满足 $\max_{i \in S} a_i \le a \le \max_{i \in S} a_i + |S| - 1$。
因此我们只需要记录一个 a 相对于 $\max_{i \in S} a_i$ 增加了多少即可。
时间复杂度 $O(2^n n^2)$，期望得分 100。

1.4 算法四

实际上对于一个集合最大是 $\max_{i \in S} a_i + 1$，因此只要记 $f[i][2]$。
时间复杂度 $O(2^n n)$，期望得分 100。

2 T2 最简单辣快来做

2.1 算法一

我会暴力！按题意模拟。

时间复杂度 $O(nQ)$，期望得分 40。

2.2 算法二

把绝对值拆开，分成四个矩形和。只需要支持单点加矩形求和即可。

使用四次二维前缀和，分别求出四个方向矩形的和，在转移的时候按需乘上系数 a 或 b。

时间复杂度 $O(hw+n+Q)$，期望得分 60。

2.3 算法三

对于每个卫星相对查询点所在的四个矩形哪一个的序列，只有 $O(n^2)$ 种可能的情况。

也就是对于卫星 (x_i, y_i)，用直线 $x = x_i$ 和 $y = y_i$ 将平面分割成 $O(n^2)$ 个格子。

如果查询点落在直线上，根据绝对值性质直接选择任意一个相邻的矩形。

如果落在格子里，就直接选择对应的矩形。

类似之前的二维前缀和，在直线的每一个交点处求出四个方向矩形的和。

在查询的时候，通过先前的离散化找到对应的格子，然后利用 $O(\log v)$ 快速幂，或者是分块预处理的 $O(1)$ 快速幂，计算出查询点到格子四个角的额外贡献。

时间复杂度 $O(n^2 + n\log n + Q\log n + \sqrt{\max(h,w)})$。期望得分 100。

2.4 算法四

对于 a, b 和 M 互质，在模 M 意义下 a, b 是有逆的。

这一档部分分是给前缀和只是简单的加法，最后再做求逆来算出负数幂次贡献的选手的，期望得分 $0 \sim 50$。

3 T3 是我的你不要抢

记字符串总长为 L。

3.1 算法一

暴力枚举答案，每次用类似 strcmp 的方法判定。
时间复杂度 $O(QL^2)$，期望得分 0。

3.2 算法二

暴力枚举答案，为了加快判定使用 hash。
时间复杂度 $O(QL)$，期望得分 0。

3.3 算法三

一直零分我放弃了，还是输出字符串长度吧。
时间复杂度 $O(Q+L)$，期望得分 10。

3.4 算法四

好像很多询问是重复的，我记忆化一下。
时间复杂度 $O(n^2L)$，期望得分 30。
看起来是这个复杂度，但是枚举答案上界的时候枚举的串长的 min。
其实时间复杂度是 $O(nL)$，期望得分 50。

3.5 算法五

如果把所有串建出 AC 自动机。设 B 满足 $S=S'+B$，$T=B+T'$，则 S 在 fail 树上到祖先链里恰好有 B。

我们要最大化这个 B，其实是要求 trie 树上根到 T 的链，与 fail 树上，根到 S 的链，并的深度最大的节点。

将问题离线，在 trie 上 dfs，每次把当前点的信息加入数据结构，dfs 到 T 处时查询 S 处的答案。

直接使用树链剖分，时间复杂度 $O(n\log^2 n)$，期望得分 $70\sim100$。

考虑我们的操作只是子树 max，单点查询，考虑标记永久化后使用可回退数据结构。时间复杂度 $O(n\log n)$，期望得分 100。

3.6 算法六

对于长度大于等于 $\sqrt{L}$ 的只有 $O(\sqrt{L})$ 个，处理它们两两之间答案。
否则只要存在一个长度小于 $\sqrt{L}$ 的，就可以直接 $O(\sqrt{L})$ 比较。
复杂度 $O((L+Q)\sqrt{L})$，期望得分 $70\sim100$。

调一下块大小能够做到 $O(L\sqrt{Q})$。
由于出题人忘记造到 trie 上了，没有体现更好的区分度在此谢罪。

4 T4 显然也是我整的

4.1 算法一

我是 xpp，我会 `puts("1")!`
由于第二个部分分数据随机，其大概率是直接连通的。所以能直接通过。
时间复杂度 $O(T)$，期望得分 10。

4.2 算法二

我会按题意模拟`!!!!`
直接使用 bfs 等方法求连通块，时间复杂度 $O(nm)$，结合算法一期望得分 20。

4.3 算法三

考虑连边实际上是变量的加减。由裴蜀定理，猜想在某些情况下，答案就是集合里元素的 gcd。

由于序列长度的限制，有 n 个元素时，我们把元素映射到 $0 \dots n-1$。

当 S_i 比较小时，即假设 $S_i \le m$，考虑到我们的方程 $\sum a_i S_i = g$，考虑构造方案。

对于我们的方程，如果 $a_i < 0$，得到 $-a_i$ 个 $-S_i$，否则得到 a_i 个 S_i，然后丢进一个可重集合 M 里。

设当前和为 v。初始 v 为 0。不断从 S 里取出元素更新 v，直到 S 空。

当 $v \ge m$ 时，如果存在 $-S_i$ 则 $v - S_i \ge 0$，一定合法。不存在则 $v = m$，因为 $\sum a_i S_i = \gcd(S_i) \le m$。

当 $v < m - 1$ 时，不一定有一个 $-S_i$ 比 v 小。但是一定存在一个 $S_i \le m$，使得 $v + S_i < 2m - 1$。

这使我们得到了一个构造，也就是当 $2m - 1 \le n - 1$，即 $S_i \le m = \frac{n}{2}$ 时的情况。

也就是说，我们能对所有值 $\le \frac{n}{2}$ 的元素求 gcd 得到 d，并把元素按照模 d 分组。

那么如果不存在 $S_i \le \frac{n}{2}$ 呢？这时候，会发现序列中间出现了一些孤立点，即 $i - S_i < 1$ 且 $i + S_i > n$。

记最小的 S_i 为 V，那么 $n - V + 1 \le i \le V$，一共有 $B = 2V - n$ 个，把这些点累加到答案里，再删除后，n 和所有 S_i 都要减去 B。

这样子新的最小值就是 $n-V$，而由之前的 $V>\frac{n}{2}$，有 $n-V<\frac{n}{2}$。也就是回归到了上面存在元素 $\le \frac{n}{2}$ 的问题。

再考虑 $S_i>\frac{n}{2}$，当 $d+S_i\le n$ 时，实际上所有分组都被它覆盖到了，我们可以令新的 $d'=\gcd(d,S_i)$。

因为 d 在这个过程中会不断变小，这样子符合 $d+S_i\le n$ 的元素是不断变多的，最大的 S_i 是会变大的。所以我们只需要把 S_i 从小到大排序，再扫一遍。如果一个元素符合条件，则 d 变小以后还是符合。如果不符合条件，则大于等于它的都不符合。所以直接扫是可以正确地更新 d 的。

剩下没处理过的 S_i 满足 $d+S_i>n$ 且 $S_i>\frac{n}{2}$。那么使用的最大的左端点一定是 $n-S_i$，最大的右端点一定是 n。由于 $n-S_i<d$，n 可以缩小为 $n'=d+(n \bmod d)$，所有剩下的 S_i 变为 $S_i'=n'-n+S_i$。由于 $g+(n \bmod d)-n+S_i\equiv S_i \pmod d$，且 $S_i-n<d\le d+(n \bmod d)$，此时问题仍然合法，且因为模意义分组，答案不变。

接下来看起来不太好做，因为有模意义分组不太好解决。但是我们可以直接往集合加入一个新的元素 d 来限制模意义分组。

如果我们把最后处理得到的 S_i' 和 d 作为一个新的集合，n' 作为一个新的点数，那么就得到一个子问题。

至于复杂度，发现我们对集合的操作，要么是用集合里的最小元素去模，要么是全体减去一个值。直到边界状态。这就相当于给一个集合求 gcd，而减去值直接将这个值加入求 gcd 的集合中。每次调用函数至多 $O(1)$ 个元素不会变小，而每个元素最多变小 $O(\log n)$ 次。因此最多递归 $O(\log n)$ 层。

加上排序等过程，得到一个时间复杂度上界 $O(m\log^2 n)$，期望得分 100。

至于某些部分分，可以看做标做法的子集。标比较简洁，细节也不多，在~~随机数据~~ 下表现非常良好，~~一点都不像俩 log 做法~~。

~~实属一道拥有良好区分度的，不可多得的联赛好题！~~