《子集和》解题报告

赵英智

1 题目

1.1 题目大意

对于一个可重集 S，定义 $f(S)$ 是从 S 中选出一个子集（可以为空集），使得这个子集的元素之和是 m 的倍数的方案数。

给定一个大小为 n 的可重集 $S = \{a_1, a_2, \ldots, a_n\}$。有 q 次询问，每次给定四个正整数 $1 \le l_1 \le r_1 < l_2 \le r_2 \le n$，求：

$$\sum_{l_1 \le i \le r_1} \sum_{l_2 \le j \le r_2} f(S \setminus \{a_i, a_j\})$$

答案对 $10^9 + 7$ 取模。

$1 \le n \le 10^4$，$2 \le m \le 200$，$1 \le q \le 10^6$，$0 \le a_i < m$ $(1 \le i \le n)$。

时间限制 3s，空间限制 512MB。

1.2 子任务设置

子任务	$n \le$	$m \le$	特殊性质	分值
1	100	—	AB	5
2	500	—	AB	5
3	—	20	AB	20
4	—	150	A	20
5	—	—	B	20
6	—	—	$l_1 = r_1, l_2 = r_2$	5
7	—	—	—	25

特殊性质 A：每次询问都在所有满足条件的 (l_1, r_1, l_2, r_2) 中随机选择。

特殊性质 B：$q \le 10^5$。

1.3 题目来源

本题是原创题。

2 题解

对于可重集 S，设 $g(S)$ 是一个长为 m 的数组 $b_0, b_1, \dots, b_{m-1}$，其中 b_i 表示选出一个 S 的子集使得元素之和模 m 余 i 的方案数。

2.1 子问题 1

先考虑如何对于每个 $1 \le i \le n$ 求出 $g(S \setminus \{a_i\})$。

由于模 m 意义下的背包问题不可撤销，使用经典的分治算法，设当前分治到 $[l,r]$，令 $mid = \lfloor \frac{l+r}{2} \rfloor$。对于子问题 $[l,mid]$，将所有 $[mid+1,r]$ 中的物品加入背包；对于子问题 $[mid+1,r]$，将 $[l,mid]$ 中的物品加入背包。分治到长为 1 的区间 $[i,i]$ 时，求出的就是 $g(S \setminus \{a_i\})$。

这个算法的时间复杂度是 $O(nm \log n)$。

2.2 子问题 2

从前向后 DP，设 $f_{i,0}$ 表示 $g(\{a_1, a_2, \dots, a_i\})$，$f_{i,1}$ 表示 $\sum_{j=1}^{i} g(\{a_1, a_2, \dots, a_i\} \setminus \{a_j\})$，有转移 $f_{i,0} = f_{i-1,0} \cdot \{a_i\}$，$f_{i,1} = (f_{i-1,1} \cdot \{a_i\}) + f_{i-1,0}$。

此处用 $g(S) \cdot \{x\}$ 表示将元素 x 加入背包 $g(S)$ 得到的背包数组。

这样可以对于每个前缀 i，求出 $\sum_{j=1}^{i} g(\{a_1, a_2, \dots, a_i\} \setminus \{a_j\})$。时间复杂度 $O(nm)$。

2.3 $l_1 = r_1, l_2 = r_2$ 的做法

使用分治算法处理所有询问，假如当前分治到 $[l,r]$，令 $mid = \lfloor \frac{l+r}{2} \rfloor$。

设 $S_1 = \{a_1, a_2, \dots, a_{mid}\}, S_2 = S \setminus S_1$。用子问题 1 的分治算法求出 $g(S_1 \setminus \{a_j\})$ $(l \le j \le mid)$ 和 $g(S_2 \setminus \{a_j\})$ $(mid < j \le r)$。对于一次询问 (L, L, R, R)，如果 $l \le L \le mid < R \le r$，可以用 $g(S_1 \setminus \{a_L\})$ 和 $g(S_2 \setminus \{a_R\})$ 两个数组合并出答案。

时间复杂度 $O(nm \log^2 n + qm)$。

2.4 算法 1

沿用 $l_1 = r_1, l_2 = r_2$ 的分治做法。对于一次询问 (l_1, r_1, l_2, r_2)，找到唯一的分治区间 $[l, r]$ 使得 $l \le l_1 \le mid < r_2 \le r$。如果 $r_1 \le mid$ 且 $l_2 > mid$，那么答案就是：

$$\begin{aligned}
&\sum_{l_1 \le i \le r_1} \sum_{l_2 \le j \le r_2} \sum_{0 \le x < m} g(S_1 \setminus \{a_i\})_x \cdot g(S_2 \setminus \{a_j\})_{(m-x) \bmod m} \\
&= \sum_{0 \le x < m} \left(\sum_{l_1 \le i \le r_1} g(S_1 \setminus \{a_i\})_x \right) \left(\sum_{l_2 \le j \le r_2} g(S_2 \setminus \{a_j\})_{(m-x) \bmod m} \right)
\end{aligned}$$

该式子可以 $O(m)$ 计算。

如果 $r_1 > mid$，可以将询问拆成 (l_1, mid, l_2, r_2) 和 $(mid + 1, r_1, l_2, r_2)$ 两个询问。第一个询问仍然能用上面的式子计算，第二个询问可以递归到分治区间 $[mid + 1, r]$ 计算。$l_2 \le mid$ 的情况同理。

于是，我们会将原先的一次询问拆成 $O(\log n)$ 个询问，每个询问需要额外花费 $O(m)$ 时间计算答案。

所以，时间复杂度是 $O(nm \log^2 n + qm \log n)$，期望得分 70 分。

2.5 算法 2

将每次询问 (l_1, r_1, l_2, r_2) 拆成四个形如 $(1, L, R, n)$ 的询问。

仍然使用分治，找到唯一的分治区间 $[l, r]$ 使得 $l \le L \le mid < R \le r$。如果预先使用子问题 2 的算法，对于每个前缀、后缀，求出去掉每个元素得到的背包数组之和，则对于 $j \in [1, l-1]$，只需要在预处理的背包数组中加入 $a_l, a_{l+1}, \ldots, a_{mid}$ 即可。对于 $j \in [l, L]$，可以用子问题 1 的算法预处理。在右侧的情况同理。

所以，对于每次询问，只需要进行一次背包合并。

时间复杂度 $O(nm \log^2 n + qm)$。

3 致谢

感谢郭权皞、曹轩铭同学与我交流本题。