

《子集匹配》 解题报告

广州大学附属中学 彭博

September 2021

1 Description

给定 n, K , 保证 $2K > n$ 。设 $[n] = \{0, 1, \dots, n-1\}$ 。设 $\mathcal{L}$ 为所有 $[n]$ 的大小为 K 的子集组成的集合, $\mathcal{R}$ 为所有 $[n]$ 的大小为 $K-1$ 的子集组成的集合, 并且对于 $S \in \mathcal{L}, T \in \mathcal{R}$, S, T 之间有边当且仅当 $S \subset T$ 。请你求出一组大小为 $|\mathcal{L}|$ 的匹配。

为了防止 IO 量过大, 使用交互的方式询问匹配。

保证 $1 \leq n \leq 27$ 、 $2K > n$ 。

对于 20% 的数据, 保证 $n \leq 15$ 。

对于 40% 的数据, 保证 $n \leq 19$ 。

对于 60% 的数据, 保证 $n \leq 23$ 。

对于 80% 的数据, 保证 $n \leq 25$ 。

空间限制 22MB , 其中交互库最多可能使用 18MB 。

时间限制 3s , 其中交互库最多可能使用 0.3s 。

2 Solution

为了方便, 我们用一个长度为 n 的 01 串 s 来表示一个 $[n]$ 的子集 S , $x \in S$ 当且仅当 $s_x = 1$ 。比如 ‘11100’ 就代表 $\{0, 1, 2\}$ 。

事实上, 如果空间限制是 512MB , 那么一些优化过的网络流或二分图匹配算法可以直接得到很高的分数, 甚至拿到满分。这也是本题卡空间的原因。

因为空间限制是 4MB , 所以各类网络流和二分图算法都无法发挥作用。我们只能尝试直接构造解。也就是说, 我们希望构造一个显式定义的函数 f , 使得对于每个 $S \in \mathcal{L}$, 有 $f(S) \subset S, f(S) \in \mathcal{R}$ 。

这也有很多种构造方式, 这里列举三种。

2.1 Approach 1

因为 $f(S)$ 就是 S 删去一个元素，所以我们希望有一个函数 g ，使得 $g(S) \in [0, K)$ ，表示我们要删掉 S 从小到大的第几个元素。设 $rnk_S(i) \in [0, K)$ 表示 i 在 S 中排到第几位；设 $id_S(i)$ 为 rnk_S 的反函数，即 S 中从小到大第 i 个元素是什么。

因为现在没有什么头绪，所以不妨先构造一个最简单的例子：

$$g(S) = \sum_{x=0}^{n-1} [x \notin S] x \pmod{K}$$

容易发现这并不是一个合法的方案：当 $n = 5, K = 3$ 时，‘10110’ 和 ‘11100’ 都被映射到了 ‘10100’。

考虑这种冲突怎样才能发生。必须是两个集合 S_1, S_2 ，满足 $S_1 \Delta S_2 = \{x, y\}$ (Δ 表示集合的对称差)，且 $id_{S_1}(g(S_1)) = x, id_{S_2}(g(S_2)) = y$ 。不妨假设 $x > y$ 。

如果是上面那种求和的方法，那么当 $S_1 \Delta S_2 = \{x, y\}$ 时，恰好有 $g(S_1) - g(S_2) = y - x \pmod{K}$ 。相当于是从 $g(S_1) = rnk_{S_1}(x)$ 出发，“绕了一圈”走到了 $rnk_{S_2}(y)$ 。

但是注意到 $x - y > rnk_{S_1}(x) - rnk_{S_2}(y)$ ，这是否说明，如果不绕圈，就无法精准打击到 y 呢？

受这个启发，我们令 $g'(S) = -g(S)$ ，此时 $g'(S_1) - g'(S_2) = x - y \pmod{K}$ ，也就只能在 $x - y = rnk_{S_1}(x) - rnk_{S_2}(y) \pmod{K}$ 时发生冲突。因此只有 $x - y = rnk_{S_1}(x) - rnk_{S_2}(y) + K$ 才会冲突。

可以发现，这是不可能出现的情况。否则我们有

$$\begin{aligned} |S_1| &= \sum_{i=0}^{y-1} [i \in S_1] + \sum_{i=x}^{n-1} [i \in S_1] + \sum_{i=y}^{x-1} [i \in S_1] \\ &= \sum_{i=0}^{y-1} [i \in S_1] + \sum_{i=x}^{n-1} [i \in S_1] + (rnk_{S_1}(x) - rnk_{S_2}(y)) \\ &\leq y + (n - x) + (rnk_{S_1}(x) - rnk_{S_2}(y)) \\ &= n - K \\ &< K \end{aligned}$$

矛盾！

于是我们获得了一个 $O(n)$ 回答一次询问的方法。

2.2 Approach 2¹

对于一个 01 串 s ，把 ‘1’ 看做 -1 ，把 ‘0’ 看做 1 ，找出此时的最大前缀和位置 p 。（如有多个取最大的位置。 $p = -1$ 是合法的，表示最大前缀和是空。）

显然 $p \neq n - 1$ ，且 s_{p+1} 为 ‘1’。把 s_{p+1} 改为 ‘0’ 即为答案。

正确性留给读者自证。

2.3 Approach 3²

把给出的 01 串 s 取反得到 s' ，然后把 s' 首尾相接得到一个环。记录 01 串 t 表示答案，初始为空。

从 s' 的第一个 ‘1’ 开始，在环上找到下一个位置 x ，使得 $s'_x = t_x = 0$ ，并令 $t_x := 1$ ，然后跳到 s' 的下一个 ‘1’。重复 $K - 1$ 次，此时的 t 即为答案。

正确性留给读者自证。

3 Acknowledgments

感谢周雨扬、陈孙立同学帮我验题。

¹ 此为周雨扬同学在验题时给出的做法

² 此为陈孙立同学在验题时给出的做法