Petrozavodsk Summer 2021 Day 2 — Problem A: AND Permutation

1 题目大意

原题链接：QOJ 题面

给定 n 个两两不同的非负整数 $a_1, a_2, \ldots, a_n$。

保证子掩码闭包：对所有 i 以及所有非负整数 x，若 $a_i \mathbin{\&} x = x$（& 为按位与），则 x 在输入序列中。

需把 a 重新排列成 $b_1, \ldots, b_n$，使得

$$b_i \mathbin{\&} a_i = 0 \qquad (1 \le i \le n).$$

可以证明解一定存在，输出任意一组解。

输入 输入第一行 n，随后 n 行给出 a_i。

输出 输出 n 行，第 i 行为 b_i。

2 数据范围

- $1 \le n < 2^{18}$；
- $0 \le a_i < 2^{60}$，互不相同；
- 时间限制 5 秒，内存限制 1024 MB；

3 记号与基本性质

记所有输入数的集合为 A。二进制位下标从 0 起，最低位为第 0 位。对整数 $x \in \mathbb{N}$ 与整数 $k \ge 0$，记

$$h_k(x) = \left\lfloor \frac{x}{2^k} \right\rfloor.$$

定义层为二元组 (S, k)（$k \ge -1$），其中存在整数 $H \ge 0$ 使

$$S = \{x \in A : \ h_{k+1}(x) = H\}.$$

称该 H 为该层的递归路径；若对应集合为空，则该层为空。称 S 为活跃集合，当且仅当 S 取上述形式且在递归过程中被处理。

由于 $0 \le x < 2^{60}$，有 $h_{60}(x)=0$，故**根层**为 $(A,59)$。

当 $k=-1$ 时，所有位模式已固定；因 A 中元素互不相同，若 $S \neq \varnothing$ 则 $|S|=1$。

当 $k \ge 0$ 时，记

$$S_0^{(k)} = \{x \in S : \text{第 } k \text{ 位为 } 0\}, \qquad S_1^{(k)} = \{x \in S : \text{第 } k \text{ 位为 } 1\},$$

并对 x 定义 $\mathrm{low}_k(x) = x \mathbin{\&} (2^k-1)$ 为低 k 位。

引理 1 (子掩码闭包的局部形式)**.** *在任意一层 (S,k) 且 $k \ge 0$，若 $x \in S_1^{(k)}$，把第 k 位清零得 $y = x - 2^k$，则 $y \in S_0^{(k)}$。*

证明. 由 $x \in S_1^{(k)}$ 知第 k 位为 1。令 $y = x - 2^k$，则 $> k$ 的各位与 x 相同，k 位由 1 变为 0，$< k$ 的各位保持不变，故 $x \mathbin{\&} y = y$。由子掩码闭包 $y \in A$，且 $h_{k+1}(y) = h_{k+1}(x)$，在第 k 位为 0，故 $y \in S_0^{(k)}$。 □

引理 2 (同层配对映射的可验证性)**.** *在任意一层 (S,k) 且 $k \ge 0$，映射 $f : S_1^{(k)} \to S_0^{(k)}$，$f(x) = x - 2^k$ 良定且单射，因而 $|S_1^{(k)}| \le |S_0^{(k)}|$。*

证明. 良定性由引理 1。若 $f(x_1) = f(x_2)$，则 $x_1 - 2^k = x_2 - 2^k$，两边同加 2^k 得 $x_1 = x_2$，故为单射。 □

4 详细做法

由于 $0 \le a_i < 2^{60}$，二进制最高可能位索引为 59。自根层 $(A,59)$ 开始递归到 $k=-1$。

在层 (S,k)：若 $k=-1$，则此时所有位均已固定，故 $|S| \le 1$；若 $|S|=1$，返回由该唯一元素构成的赋值集合；若 $|S|=0$ 则结束该分支。若 $k \ge 0$，执行：

1. **分组。** 按第 k 位把 S 划分为 $S_0^{(k)}$ 与 $S_1^{(k)}$。
2. **确定配对。** 对每个 $x \in S_1^{(k)}$，由引理 1 取 $y = f(x) = x - 2^k \in S_0^{(k)}$；由引理 2，$f$ 为单射，因而配对唯一且满足 $|S_1^{(k)}| \le |S_0^{(k)}|$。未被配到的 $S_0^{(k)}$ 元素保持独立。
3. **递归。** 分别在 $S_0^{(k)}$ 与 $S_1^{(k)}$ 内部递归到位 $k-1$，得到两组该集合的局部重排，且仅在位段 $[0,k-1]$ 上满足按位与为 0。在递归子阶段中，赋值集合在每个子集合内分别是该子集合元素的一个置换。
4. **合并（交换赋值）。** 对每一对 (x,y)（x 的第 k 位为 1，y 的第 k 位为 0，且 $\mathrm{low}_k(x) = \mathrm{low}_k(y)$），交换两组递归得到的赋值：$x$ 使用分配给 y 的数，y 使用分配给 x 的数；未被配到的 $S_0^{(k)}$ 元素沿用本组递归赋值。

完成所有二进制位后，对 A 的重排即确定。设赋值函数 $g : A \to A$ 将每个 $x \in A$ 映射到赋给 x 的数，最终输出按输入顺序取 $b_i = g(a_i)$。

5 正确性证明

归纳陈述

在层 (S,k) 的处理完成后（当 $k \ge 0$ 时即合并后）满足：

(i) 对所有 $x \in S$，赋值 v_x 满足 $v_x \,\&\, \mathrm{low}_{k+1}(x) = 0$；

(ii) 赋值集合 $\{v_x :\ x \in S\}$ 与 S 一一对应，为 S 的一个置换。

当 $k=-1$ 时，若 $S=\{z\}$，返回 $v_z = z$，满足 (i) 与 (ii)；若 $S=\varnothing$，结论同样成立。

从 $k-1$ 推到 k（$k \ge 0$）

取任意一对 (x,y)，其中 x 的第 k 位为 1、y 的第 k 位为 0，且 $\mathrm{low}_k(x) = \mathrm{low}_k(y)$。

递归在位段 $[0,k-1]$ 分别给出赋值 v_x 与 v_y，并且（由归纳陈述 (ii)）$v_x \in S_1^{(k)}$、$v_y \in S_0^{(k)}$，满足

$$v_x \,\&\, \mathrm{low}_k(x) = 0, \qquad v_y \,\&\, \mathrm{low}_k(y) = 0.$$

由于低 k 位相同，v_x 与 $\mathrm{low}_k(y)$ 相与为 0，v_y 与 $\mathrm{low}_k(x)$ 相与为 0。合并时交换赋值：令 x 取 v_y、y 取 v_x。则：

- 在第 k 位：v_y 来自 $S_0^{(k)}$，其第 k 位为 0，于是 $v_y \,\&\, x$ 在该位为 0；v_x 来自 $S_1^{(k)}$，其第 k 位为 1，而 y 的第 k 位为 0，于是 $v_x \,\&\, y$ 在该位为 0；
- 在低于 k 的位：由归纳陈述 (i) 与 $\mathrm{low}_k(x) = \mathrm{low}_k(y)$，相与仍为 0。

未被配到的 $z \in S_0^{(k)}$ 保持 v_z，其第 k 位为 0，低位由归纳陈述 (i) 保证。故 (i) 在 $[0,k]$ 成立。

对 (ii)：递归在 $S_0^{(k)}$ 与 $S_1^{(k)}$ 内部分别给出了到各自集合的置换。合并阶段仅在每一对 (x,y) 上交换两侧赋值 v_x 与 v_y，交换是两元素置换的组合，未配到元素保持不变，因而总体仍为 $S = S_0^{(k)} \cup S_1^{(k)}$ 的一个置换。由引理 2，每个 $x \in S_1^{(k)}$ 有唯一配对 $y \in S_0^{(k)}$，交换恰好覆盖这些成对元素，不重不漏，故 (ii) 保持成立。

定理 1 (正确性). *最终得到的是 A 的一个重排，且对每个输入位置 i，输出的 b_i 满足 $b_i \,\&\, a_i = 0$。*

6 复杂度分析

设最高考虑位数 $L = 60$。每层执行分组、配对、交换赋值。

存在如下三种做法：

- **哈希表：** 为每个活跃集合 S 维护一张独立哈希表，以 low_k 为键记录配对。记全局元素数为 n。在整个按位递归过程中，每一层所有活跃集合的元素总数之和为 n，故每层代价为 $\mathcal{O}(n)$，总时间 $\mathcal{O}(nL)$，空间 $\mathcal{O}(n)$。

- **提前排序 + 双指针：** 初始按数值升序排序。对任一层 k，将 S 按第 k 位分裂为 $S_0^{(k)}$ 与 $S_1^{(k)}$；在固定 $>k$ 位与第 k 位后，元素仅在低 k 位上可能不同。由于 A 中元素互不相同，低 k 位在各自子序列中两两不同。高位固定时，数值大小比较等价于按 low_k 比较，故两子序列按 low_k 严格升序。于是可用双指针在线性时间同时扫描两序列，并按相等的 low_k 完成配对。总时间 $\mathcal{O}(n \log n + nL)$，空间 $\mathcal{O}(n)$。
- **平衡树映射（`map`）：** 查找/插入为 $\mathcal{O}(\log n)$，每层 $\mathcal{O}(n \log n)$，总时间 $\mathcal{O}(nL \log n)$，空间 $\mathcal{O}(n)$。

7 参考资料

1. OI-Wiki, 位运算, https://oi-wiki.org/math/bit.
2. OI-Wiki, 递归 & 分治, https://oi-wiki.org/basic/divide-and-conquer.
3. Wikipedia, *Bitwise operation*, https://en.wikipedia.org/wiki/Bitwise_operation.