

「多控制反转」解题报告

出题人：张致(ioi2023_016)

题目大意

给定三个正整数 n, m, Q ，其中 $n + 1 < m, Q \geq 8n$ 。有 m 个 01 变量 $a_0, a_1, \dots, a_{m-1}$ ，初值未知。对于一个 $U = \{0, 1, \dots, m-1\}$ 的子集 S 和一个 $b \in U$ ，假如 $b \notin S$ ，那么可以进行一次操作，令 $a_b := a_b \oplus (\bigwedge_{i \in S} a_i)$ ，其中 $\oplus$ 是异或运算， $\wedge$ 是逻辑与运算，这样的操作记为 (S, b) ，假如 $|S| = n$ ，那么也称该操作为 n 控制反转操作。现在你只能使用 0 控制反转操作、1 控制反转操作和 2 控制反转操作，要求你构造一个长度不超过 Q 的操作序列，使得依次执行完序列里的操作后与执行操作 $(\{0, 1, \dots, n-1\}, n)$ 等价。

注意：显然我们不可能用所有 2^m 种初始情况都对你构造的操作序列进行检查，在真实测试中，对于每个测试点，我们会选取一部分可能的初始情况进行检查，设这些情况构成的集合为 T ，你只需要保证你构造的操作序列在 T 内的所有情况满足条件即可。

数据范围

子任务编号	$n \leq$	$Q =$	$m =$	特殊限制	子任务分数	子任务依赖
1	20	$8n + 1$	$2n + 2$	A	15	
2	50	$8n + 1$	$2n + 2$	A	10	1
3	50	$8n + 1$	$2n + 2$		10	2
4	20	$n^2 + 1$	$n + 2$	A	10	
5	20	$n^2 + 1$	$n + 2$		20	4
6	50	$28n + 1$	$n + 2$		10	
7	100	$8n + 1$	$n + 2$	A	10	2, 4
8	100	$8n + 1$	$n + 2$		15	3, 5, 6, 7

特殊限制A： $\forall (a_0 a_1 \dots a_{m-1}) \in T$ ，保证 $a_{n+1} = a_{n+2} = \dots = a_{m-1} = 0$ 。

对于所有数据，保证 $0 \leq n \leq 100, m \geq n + 2, Q \geq 8n + 1$ 。

题解

注意到在操作 $(\{0, 1, \dots, n-1\}, n)$ 中， $a_{n+1}, a_{n+2}, \dots, a_{m-1}$ 这些元素并没有参与操作，而且多控制反转操作是可逆的，那么我们可以用它们来辅助运算，称它们为辅助 bit。特别的，如果辅助 bit a_i 初始一定是 0，那么称其为 clean bit，否则称其为 dirty bit。（注意，下文中的 clean bit 和 dirty bit 一定是辅助 bit，辅助 bit 是相对于操作定义的， a_c 是 (S, b) 的辅助 bit 当且仅当 $c \notin S \wedge c \neq b$ 。）

情况一：保证有 $m = 2n + 2$ ，特殊限制 A

考虑在这种情况下， $a_{n+1}, a_{n+2}, \dots, a_{m-1}$ 都是 clean bit，那么我们可以执行操作 $(\{a_0, a_1\}, a_{n+1})$ ，那么此时 $a_{n+1} = a_0 \wedge a_1$ ，再执行操作 $(\{a_{n+1}, a_2\}, a_{n+2})$ ，那么 $a_{n+2} = a_0 \wedge a_1 \wedge a_2$ ，类似地进行 $n - 2$ 次后， $a_{2n-2} = \bigwedge_{i=0}^{n-2} a_i$ ，再进行操作 $(\{a_{2n-2}, a_{n-1}\}, a_n)$ ，那么此时已经成功使得 $a_n = a_n \oplus \bigwedge_{i=0}^{n-1} a_i$ ，知道多控制反转是可逆的，那么把 $a_{n+1}, a_{n+2}, \dots, a_{2n-2}$ 变回 0 就行了。

情况二：保证 $m = 2n + 2$

此时， $a_{n+1}, a_{n+2}, \dots, a_{m-1}$ 是 dirty bit，但我们仍然可以用它们辅助运算。

更一般地，我们想要实现 $(X = \{x_0, x_1, \dots, x_{n-1}\}, z)$ ， $Y = \{y_0, y_1, \dots, y_{n-3}\}$ 是辅助 bit。

假如 Y 里全是 clean bit，由上一种情况，可以等价于：

$$\begin{aligned} &(\{x_0, x_1\}, y_0), (\{x_2, y_0\}, y_1), (\{x_3, y_1\}, y_2), \dots, (\{x_{n-2}, y_{n-4}\}, y_{n-3}), \\ &(\{x_{n-1}, y_{n-3}\}, z), \\ &(\{x_{n-2}, y_{n-4}\}, y_{n-3}), (\{x_{n-3}, y_{n-5}\}, y_{n-4}), \dots, (\{x_2, y_0\}, y_1), (\{x_0, x_1\}, y_0). \end{aligned}$$

当 Y 中为 dirty bit 时：

注意到无论 a_2 初始为什么值，都有 $(\{1, 2\}, 3), (\{0\}, 2), (\{1, 2\}, 3), (\{0\}, 2)$ 等价于 $(\{0, 1\}, 3)$ 。且 $(\{0\}, 2), (\{1, 2\}, 3), (\{0\}, 2), (\{1, 2\}, 3)$ 也与 $(\{0, 1\}, 3)$ 等价。（可以枚举所有初始情况验证）^[1]

稍微扩展以下，有 $(\{x, c\}, z), (S, c), (\{x, c\}, z), (S, c)$ 等价于 $(S \cup \{x\}, z)$ 。

注意到两个相邻的操作如果涉及的元素不交，那么这两个操作执行的先后顺序可以调换，且如果两个相邻的操作相同，那么可以删去。

那么：

$$\begin{aligned} (\{x_0, x_1, \dots, x_{n-1}\}, z) &= (\{x_{n-1}, y_{n-3}\}, z), (\{x_0, x_1, \dots, x_{n-2}\}, y_{n-3}), (\{x_{n-1}, y_{n-3}\}, z), (\{x_0, x_1, \dots, x_{n-2}\}, y_{n-3}) \\ &= (\{x_{n-1}, y_{n-3}\}, z), \\ &\quad (\{x_{n-2}, y_{n-4}\}, y_{n-3}), (\{x_0, x_1, \dots, x_{n-3}\}, y_{n-4}), (\{x_{n-2}, y_{n-4}\}, y_{n-3}), (\{x_0, x_1, \dots, x_{n-3}\}, y_{n-4}), \\ &\quad (\{x_{n-1}, y_{n-3}\}, z), \\ &\quad (\{x_0, x_1, \dots, x_{n-3}\}, y_{n-4}), (\{x_{n-2}, y_{n-4}\}, y_{n-3}), (\{x_0, x_1, \dots, x_{n-3}\}, y_{n-4}), (\{x_{n-2}, y_{n-4}\}, y_{n-3}) \\ &= (\{x_{n-1}, y_{n-3}\}, z), \\ &\quad (\{x_{n-2}, y_{n-4}\}, y_{n-3}), (\{x_0, x_1, \dots, x_{n-3}\}, y_{n-4}), (\{x_{n-2}, y_{n-4}\}, y_{n-3}), \\ &\quad (\{x_{n-1}, y_{n-3}\}, z), \\ &\quad (\{x_{n-2}, y_{n-4}\}, y_{n-3}), (\{x_0, x_1, \dots, x_{n-3}\}, y_{n-4}), (\{x_{n-2}, y_{n-4}\}, y_{n-3}) \end{aligned}$$

继续进行类似的展开和化简后，可以知道 (X, z) 等价于：

$$\begin{aligned} &(\{x_{n-1}, y_{n-3}\}, z), (\{x_{n-2}, y_{n-4}\}, y_{n-3}), (\{x_{n-3}, y_{n-5}\}, y_{n-4}), \dots, (\{x_2, y_0\}, y_1), \\ &(\{x_0, x_1\}, y_0), (\{x_2, y_0\}, y_1), (\{x_3, y_1\}, y_2), \dots, (\{x_{n-2}, y_{n-4}\}, y_{n-3}), \\ &(\{x_{n-1}, y_{n-3}\}, z), \\ &(\{x_{n-2}, y_{n-4}\}, y_{n-3}), (\{x_{n-3}, y_{n-5}\}, y_{n-4}), \dots, (\{x_2, y_0\}, y_1), (\{x_0, x_1\}, y_0), \\ &(\{x_2, y_0\}, y_1), (\{x_3, y_1\}, y_2), \dots, (\{x_{n-2}, y_{n-4}\}, y_{n-3}). \end{aligned}$$

至此，我们就可以在 $4n$ 次操作内，利用 $n - 2$ 个 dirty bit，实现任意 n 控制反转操作。利用 dirty bit 是很好的，因为一个变量能作为辅助 bit 辅助运算，只需要它在我们要实现的多控制反转操作中不出现，而且一个 dirty bit 辅助运算后它本身的值并没有改变。

情况三： $m = n + 2, Q = n^2 + 1$

现在辅助 bit 只剩一个了，利用上一个情况里我们有的一个结论：

$(\{0\}, 3), (\{1, 3\}, 2), (\{0\}, 3), (\{1, 3\}, 2)$ 等价于 $(\{0, 1\}, 2)$ ，其中 a_3 是 dirty bit。

扩展一下，对于 $A, B \subset U, A \cap B = \emptyset, z \in U, z \notin A \cup B$ ，以及 dirty bit a_c ，那么：

$(A, c), (B \cup \{c\}, z), (A, c), (B \cup \{c\}, z)$ 等价于 $(A \cup B, z)$ 。

那么我们发现，利用一个 dirty bit，我们把一个 n 控制反转操作，拆成了 2 个 $|A|$ 控制反转操作，2 个 $|B| + 1$ 控制反转操作。在进行 $|A|$ 控制反转操作时， $B \cup \{z\}$ 里的元素都是 dirty bit；进行 $|B| + 1$ 控制反转操作时， A 里的元素都是 dirty bit，所以进行这些操作时都有辅助 bit。那么令 $T(n)$ 为利用一个辅助 bit 时， n 控制反转操作需要多少个 0, 1, 2 控制反转操作实现，那么有

$T(0) = T(1) = T(2) = 1, T(n) = 2 \min_{i=0}^n \{T(i) + T(n-i+1)\}$ ，那么有 $T(n) \leq n^2$ 。

情况四： $m = n + 2, Q = 8n + 1$

考虑把上两种情况结合起来，上一种情况拆解 n 控制反转操作时，拆成了 2 个 $|A|$ 控制反转操作，2 个 $|B| + 1$ 控制反转操作。在进行 $|A|$ 控制反转操作时， $B \cup \{z\}$ 里的元素都是 dirty bit；进行 $|B| + 1$ 控制反转操作时， A 里的元素都是 dirty bit。不难使得 $|B| + 1 \geq |A| - 2 \wedge |A| \geq |B| - 1$ ，那么进行 $|A|$ 控制反转操作时，至少有 $|A| - 2$ 个 dirty bit，进行 $|B| + 1$ 控制反转操作时，至少有 $|B| - 1$ 个 dirty bit，都可以应用情况二的解法解决。

总操作数为 $2(4|A| - 8 + 4(|B| + 1) - 8) = 8n - 24$ ，实现精细可能可以获得更小的常数。

参考资料

[1] 刘洋, 龙桂鲁. 任意量子比特门的分解方法, CN101118608A[P]. 2008.