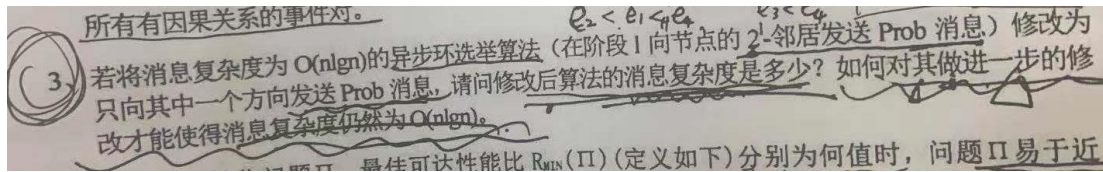


1. 题目如下图所示：



答：修改后算法的消息复杂度是  $O(n^2)$ （最坏情况下）。分析如下：

(1) 在 **Phase  $l$**  里：

- 一个处理器启动的 msg 数目至多为： $2 \times 2^l$
- 启动者数量：  
 当  $l = 0$ ，有  $n$  个启动者（最多），则会产生  $2n$  个 msg；  
 当  $l \geq 1$ ，在  $(l - 1)$  阶段结束时成为临时 leader 的节点均是启动者。在 **Phase  $l$**  结束时，临时 leader 数至多为  $n - 2^l$ 。（解释见附录）

(2) 计算所需轮数：

由于在 **Phase  $l$**  结束时，临时 leader 数至多为  $n - 2^l$ ，所以所需轮数为  $\lg(n - 1)$ 。选举结束时，产生的最后 leader 会转发通知，产生  $n$  个 msg。

(3) 消息总数计算：

$$2n + \sum_{l=1}^{\lg(n-1)} 2 \times 2^l \times (n - 2^{l-1}) + n \leq \frac{8}{3} n^2 = O(n^2)$$

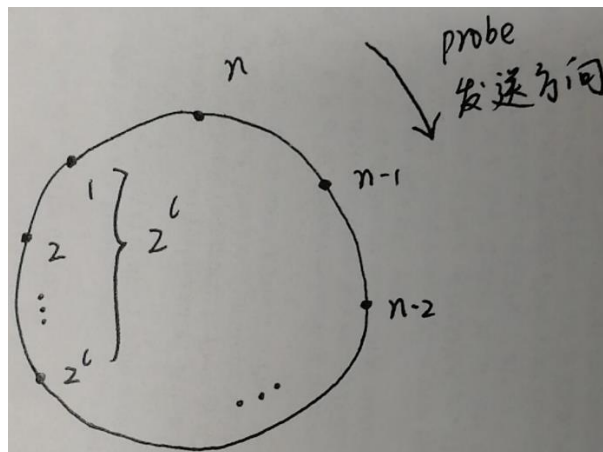
**修改方案：**修改处理器当选临时 leader 的条件。原条件为：若处理器从发送方向收到 reply 消息，则成为局部临时 leader。修改为，若处理器从发送方向收到 reply 消息，且未收到过 id 比自己的 id 大的 probe 消息，则成为局部临时 leader。即，对于一个临时 leader 来说，一旦遇到收到的 probe 消息中 id 比自己 id 大的情况，则该处理器放弃临时 leader 身份。

## 附录

性质：在 **Phase  $l$**  结束时，临时 leader 数至多为  $n - 2^l$ 。

证明：首先明确处理器失去临时 leader 身份的条件：没有从发送方向收到 reply 消息。也就是说，对于一个临时 leader 处理器，即使从另外一个方向收到一个 id 比自己 id 大的 probe 消息，但根据算法规则，只要它能收到发送方向传来的 reply 消息，就能继续作为局部临时 leader。因为题中只修改了 probe 消息发送规则，其他规则并未改变。

所以考虑一个最坏情况：环上  $n$  个处理器根据 id 有序排列（假设 id 为  $\{1, 2, 3, \dots, n\}$ ，排列方式如下图所示）。



在 **Phase  $l$**  结束时，只有最后  $2^l$  个处理器会因为自己的 probe 消息被 id 等于  $n$  的处理器没收到而收不到 reply 消息。所以其他的  $n - 2^l$  个处理器仍然是临时 leader。