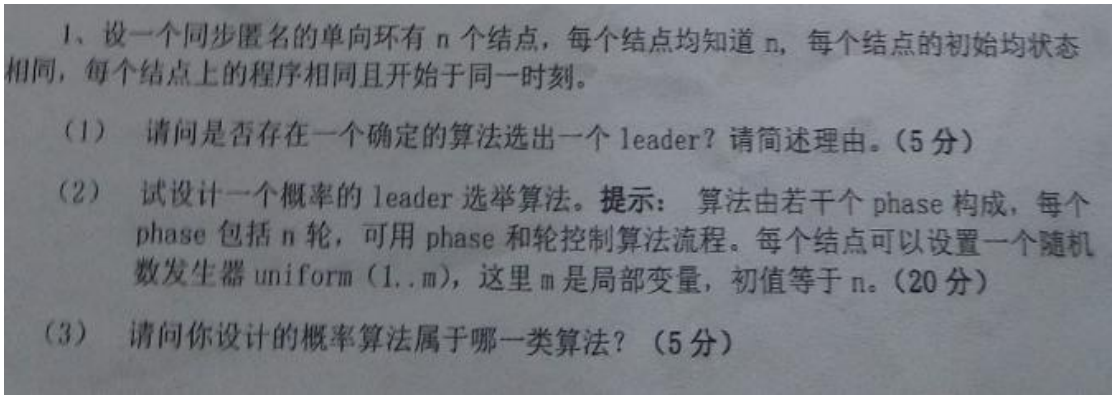
1.



1、设一个同步匿名的单向环有 n 个结点，每个结点均知道 n，每个结点的初始均状态相同，每个结点上的程序相同且开始于同一时刻。

（1） 请问是否存在一个确定的算法选出一个 leader？请简述理由。（5 分）

（2） 试设计一个概率的 leader 选举算法。**提示：** 算法由若干个 phase 构成，每个 phase 包括 n 轮，可用 phase 和轮控制算法流程。每个结点可以设置一个随机数发生器 uniform（1..m），这里 m 是局部变量，初值等于 n。（20 分）

（3） 请问你设计的概率算法属于哪一类算法？（5 分）

(1)由 Lemma3.1 可得。(同步匿名非均匀)

假设 R 是大小为 n>1 的环（非均匀），A 是其上的一个匿名算法，它选中某处理器为 leader。因为环是同步的且只有一种初始配置，故在 R 上 A 只有唯一的合法执行。

Lemma3.1： 在环 R 上算法 A 的容许执行里，对于每轮 k，所有处理器的状态在第 k 轮结束时是相同的。Note：每个处理器同时宣布自己是 Leader。

(2)

n 个节点每个节点随机产生一个(1-n)的随机数作为自己的 id，然后将它发送，

若绕场一周后回到了该节点，该节点就用那个 id;

否则重新产生随机数，直到绕场一周回到该点；

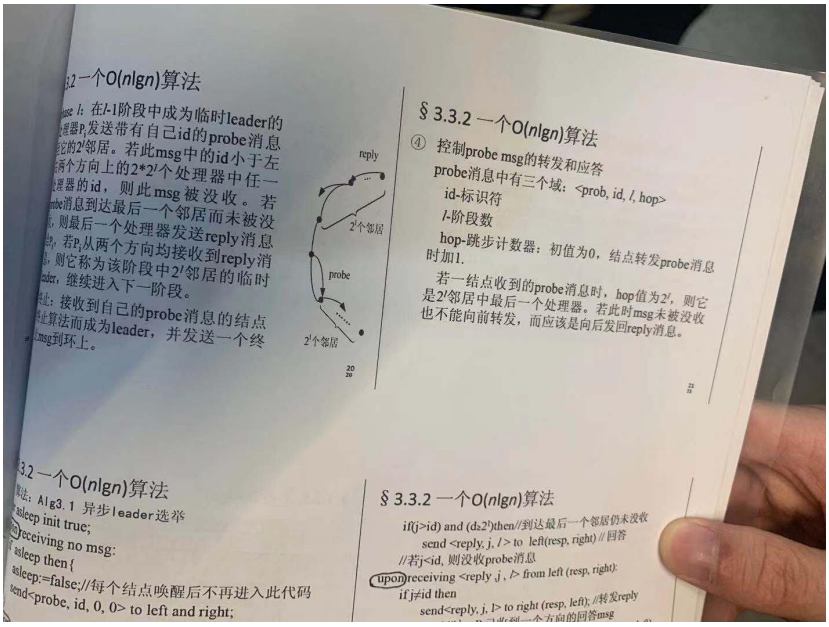
对于每个节点若收到的 msg 中的 id 和自己产生的 id 一样就没收它，反之转发该 msg;

每个节点重复上述过程直至 n 节点的 id 均不相同；

于是得到一个同步非匿名环，可选举一个 id 最大的 leader.

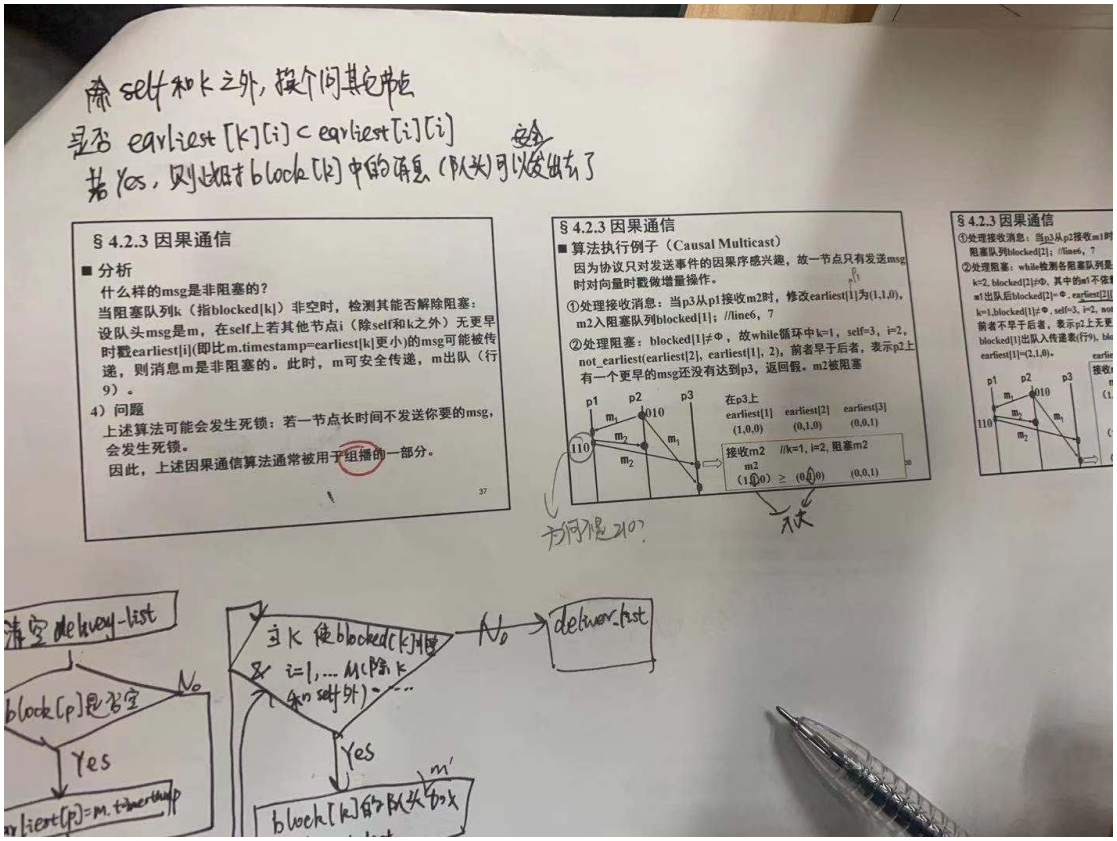
(3)Las Vegas 算法

2.为啥 leader 选举算法选取两边的 2^l 个处理机而不是其他底数

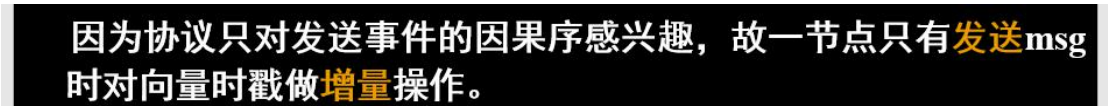


答：如果用其他数作为底数，如 3，复杂的仍为 nlogn，不过 log 的底数是 3。另一方面，如果底数选得过大，有可能造成算法时间接近 n^2。例如，如果有环有十个处理机，底数也选为 10，那么，prob 消息将有 200 个。

3. 下图的数字为何不是 210 而是 110

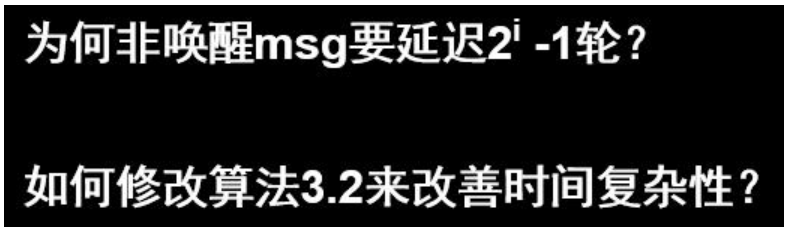


答:



因此在发送 m2 之前 p1 时间戳为 010，发送时变为 110.

4. 同步环问题。算法 3.2.下图



答:

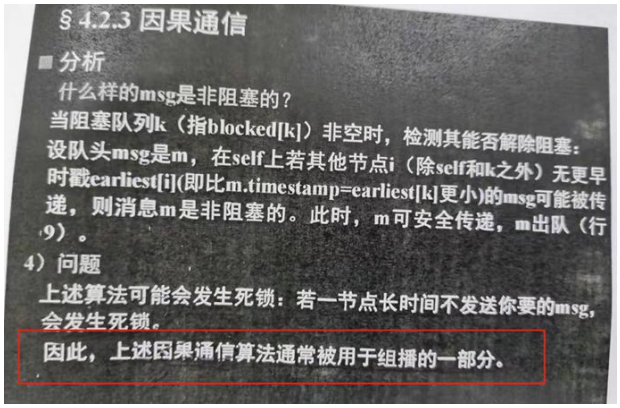
(1) 降低消息复杂度（Id 最小的节点被选举为 Leader，Leader 节点消息的转发速度最快，则可以使得其他非 leader 的转发 msg 被淹没）。

(2) 方案 1：添加 Relay 变量，保证消息在转发节点不延迟，时间复杂度由 O(n*2^i)降为 O(N*2^i+n-N)，N 为自发唤醒的节点数。

方案 2：原算法延迟函数为 f(id)=2^id，时间复杂度为 O（n*2^i）。通过重新定义延迟函数来降低时间复杂度，如 f(id)=c*id 等。

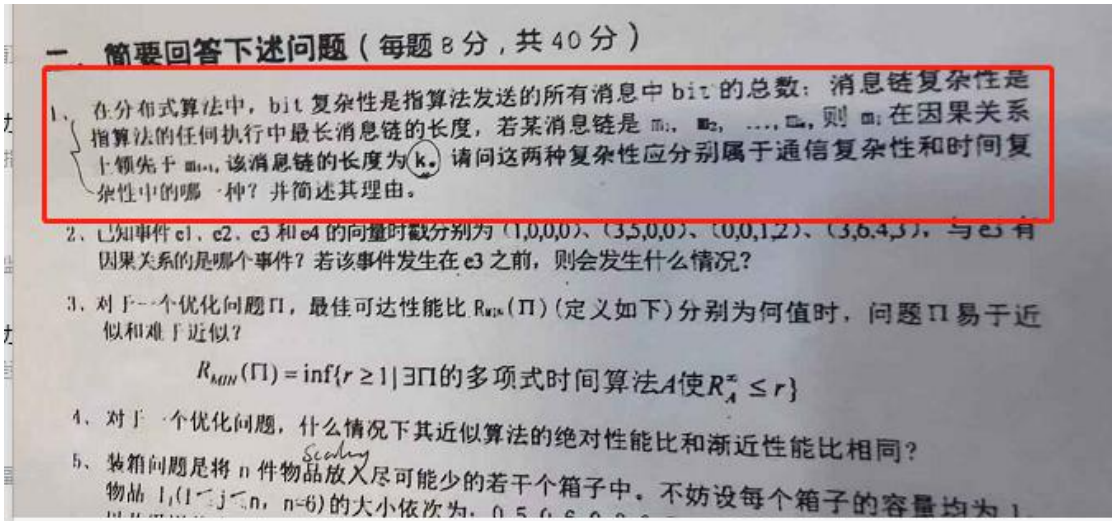
Note：思考方案 2 中消息复杂度和时间复杂度的关系!

5. 为啥被用于组播?



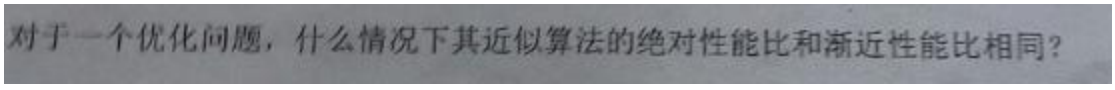
答：因为本地节点的初始化在本地节点完成，有处理阻塞的机制。

6.



bit 复杂性属于通信复杂性，消息链复杂性属于时间复杂性；若在一个分布式算法中每个 msg 信息的 bit 数目相同，则 msg 的个数就等于 bit 的总数除以一个 msg 的 bit 数目，则 bit 复杂性可以等价为 msg 复杂性；消息链复杂性是最长消息链的长度，在同步系统中它就是最大轮数，异步系统中假定任何执行的 msg 延迟至多是一个单位时间，它就是计算直到终止时间的最大运行时间，在同，异步系统中皆为时间复杂性。

7



答：

具有 Scaling 性质的问题，近似算法的绝对性能比和近似性能比是相同的。