hw3

1c

存在问题：当前PoA封印只依赖单节点签名，存在单点失效风险。一旦该节点的私钥泄露或被盗，攻击者即可伪造任意合法区块，不需要任何算力成本。

可能解决设计：采用混合系统，规定PoA节点连续封印的最大区块数，超过就必须等待新的PoW块或轮换其他节点，降低单节点长期掌控风险，从而最大程度的减小单点私钥泄露的危害。

2.1

假设：公司基于请求提交顺序依次处理提款

可能的攻击方式：恶意且资金充足的用户通过提交大量提款请求来挤占系统的可用流动性，这些用户可以通过大额提款或将资金拆分为大量小额提款请求，垄断提款队列。

处理方式：除了对单个用户每日提取限额以外，还要按比例分配。当剩余金额不足时，按每个人请求金额占比，把有限的钱公平分掉，防止恶意用户抢光资源。无论谁先请求，都根据每个人请求的金额占总请求的比例，平均分配当前能拿到的钱。

这样无论如何诚实用户总会提取到一部分的钱，永远不会提取失败，

2.2

提出：找零聚合机制

把多笔找零累积到一起再统一生成UTXO，减少Moonbase需要维护的UTXO数量。在每次提款后，不为每一笔找零单独生成新的UTXO，而是将所有小额找零暂时累积到一个统一的找零钱池中，当零钱池累积到一定金额，Moonbase再统一生成一个UTXO来记录，这样可以减少UTXO碎片化。

3.1

答案：$\frac{1}{4}n$

记腐化人数为 f，诚实人数为 $n-f$

诚实玩家收到的正确 (VOTE, m) 的数量是至少 $n-f$

因此要满足： $n-f \geq \frac{3}{4}n$

也就是说，$f \leq \frac{n}{4}$

- 举例：设 n=12，f=3。
- 发送者诚实，发送消息 m。
- 9 个诚实玩家都收到 m 并投 (VOTE, m)。
- 3 个坏玩家投了其他的假消息或不投。
- 诚实玩家刚好收到$\frac{3}{4}n$的正确消息，因此输出m，若$f > \frac{n}{4}$，则收不到，就会输出错误。

3.2

一致性要求所有所有诚实玩家输出相同的值。

方案：发送者变坏，在第1轮中，给不同的玩家发送不同的消息，比如

- 给n-2的玩家发送 m1
- 给1名玩家发送 m2 (m1 ≠ m2)

第2轮，各玩家将自己收到的消息 (VOTE, mi) 广播出去

在第3轮，由于只有发送者时坏的：

- 对于n-2名玩家，收到的投票数量满足需求，因此会输出m1。
- 对于那一名玩家，由于没有任何其他的投票，因此出输出错误。
- 因此破坏了一致性。

3.3

- 设两位诚实玩家 i 和 j。
- 假设 i 输出 m1，j 输出 m2。
- 根据协议，i 必须收到至少 3/4n 个 (VOTE, m1)。
- 同样，j 必须收到至少 3/4n 个 (VOTE, m2)。
- 由于每位玩家仅投票一次，因此这两个集合的大小加起来是至少：

$$\frac{3}{4}n + \frac{3}{4}n = \frac{3}{2}n$$

- 但整个系统只有 n 个玩家，其中腐化玩家最多 f < n/2。
- 因此，两个集合必须有至少有n/2的交集。
- 又由于坏人数小于 n/2，所以交集里面一定有至少一个诚实玩家，那么 m1 和 m2 必须是相同的。
- 因此满足弱一致性。