Homework 2

Problem 1 - Basic Ethereum Knowledge

1. GASLIMIT is the actual amount of gas spent at the completion of the Block creation.
2. The Ethereum Accounts can send transactions for ether transfer or they can send transactions to invoke a smart contract code.
3. Ethereum full node hosts the software needed for transaction initiation, valida- tion, mining, block creation, and smart contract execution.
4. Miner nodes receive, verify, gather and execute transactions.
5. A Smart Contract is a piece of code deployed in the Blockchain node. Execution of a smart contract is initiated by a message embedded in a transaction.

Answer:

1. False
2. True
3. True
4. True
5. True

Problem 2 - Etheurem vs. Bitcoin
Review the Ethereum white paper and answer the following questions.

1. What are the most important limitations for the scripting language as imple- mented in Bitcoin?
2. What is the purpose of gas in Ethereum? Why is gas not needed in Bitcoin?
3. What is the main difference between Ethereum and Bitcoin with regard to the blockchain architecture?
4. Although Ethereum stores the entire states in every block, Ethereum has com- parable efficiency with Bitcoin, and more savings in space. Why?

Answer:

1. The most important limitation of bitcoin is its Turing Incomplete property, in other words, bitcoin script lacks of Turing Completeness.
2. The Gas fee in Ethereum can prevent the operating nodes from malicious programs which can destroy the Ethereum network by using a simple loop operation to cost computational resource of all nodes. In Bitcoin, since it does not need all node to run applications, and the script is Turing Incomplete, the Bitcoin network won't be affected by the malicious programs with loops.
3. The difference between Ethereum and Bitcoin is the fact that Bitcoin is nothing more than a currency, whereas Ethereum is a ledger technology that companies are using to build new programs.

   In architecture part, the block of Ethereum is more complicated, the Ethereum requires each block contains the transaction list and the list of headers of uncle blocks.
4. Because in Ethereum, the state is stored in the tree structure, and after every block only a small part of the tree needs to be changed. Thus, in general, between two adjacent blocks the vast majority of the tree should be the same, and therefore the data can be stored once and referenced twice using pointers (ie. hashes of subtrees).

   A special kind of tree known as a "Patricia tree" is used to accomplish this, including a modification to the Merkle tree concept that allows for nodes to be inserted and deleted, and not just changed, efficiently. Additionally, because all of the state information is part of the last block, there is no need to store the entire blockchain history - a strategy which, if it could be applied to Bitcoin, can be calculated to provide 5-20x savings in space.

Problem 3 - Bitcoin Double-Spending Analysis

In Bitcoin, the standard practice for a merchant is to wait for n confirmations of the paying transaction before providing the product. While the network is finding these confirming blocks, the attacker is building his own branch which contradicts it. When attempting a double-spend, the attacker finds himself in the following situation. The network currently knows a branch crediting the merchant, which has n blocks on top of the one in which the fork started. The attacker has a branch with only m additional blocks, and both are trying to extend their respective branches. Assume the honest network and the attacker has a proportion of p and q of the total network hash power, respectively.

1. [10 pts] Let a_z denote the probability that the attacker will be able to catch up when he is currently z blocks behind. Find out the closed form for a_z with respect to p, q and z. Detailed analysis is needed. (Hint: a_z satisfies the recurrence relation $a_z = pa_{z+1} + qa_{z-1}$.)

2. [10 pts] Compared with the Bitcoin white paper, we model m more accurately as a negative binomial variable. m is the number of successes (blocks found by the attacker) before n failures (blocks found by the honest network), with a probability q of success. Show that the probability for a given value m is $P(m) = \begin{pmatrix} m+n-1 \\ m \end{pmatrix} p^n q^m$.

3. [10 pts] We assume one block was pre-mined by the attacker before commencing the attack (as in Finney attack). The probability for the double-spend to succeed, when the merchant waits for n confirmations, is equal to $r = \sum_{m=0}^{\infty} P(m) a_{n-m-1}$. Visualize the relation between r and q for various confirmation numbers $n = 2, 4, 6, 10$.

Answer:

1. Indicated by the hint, we can write the recursive form of a_z.

$$a_z - a_{z-1} = (\frac{q}{p})^{z-1}(a_1 - a_0) + a_0 \tag{1}$$

And if we do the same thing for a_{z-1}, we can write a_z as,

$$a_z = \sum_{i=0}^{z-1} (\frac{q}{p})^i (a_1 - a_0) \tag{2}$$

We have one special case: a_0. In this case, it means that the attacker has the same amount of blocks as the current chain. Therefore, the probability is 1.

Thus, we have:

$$a_z = \sum_{i=0}^{z-1} (\frac{q}{p})^i (a_1 - 1) + 1 \tag{3}$$

If $q \neq p$, we can treat equation (3) as geometric sequence, therefore, we can rewrite equation (3) to:

$$a_z = (a_1 - 1) \frac{1 - \left(\frac{q}{p}\right)^z}{1 - \frac{q}{p}} + 1 \tag{4}$$

Also, if $q = p$, we have,

$$a_z = a_1 z - k + 1 \tag{5}$$

In order to solve a_z, we need to find out the value of a_1, in this case, we can assume:

$$\lim_{z \to \infty} a_z = 0 \tag{6}$$

where it means that after infinite blocks that the probability of catching up is equal to 0.

By equation (4), we can calculate a_z since we already know that when z goes to infinite, a_z goes to 0.

$$\lim_{z \to \infty} (a_1 - 1) \frac{1 - \left(\frac{q}{p}\right)^z}{1 - \frac{q}{p}} + 1 = 0 \tag{7}$$

$$a_1 = 1 - \frac{1 - \frac{q}{p}}{1 - \left(\frac{q}{p}\right)^z} \tag{8}$$

When $p > q$, $a_1 = \frac{q}{p}$, when $p < q$, $a_1 = 1$.

By combining equation (4) and (5) and results above, we have,

$$a_z = \begin{cases} \left(\frac{q}{p}\right)^z, & \text{if } p > q \\ 1, & \text{otherwise} \end{cases} \tag{9}$$

2. We can write the probability of m success and n failures as:

$$P = p^n q^m \tag{10}$$

Since the last trial must be a failure, so there must be m success among previous $m + n + 1$ trials. Therefore, we can have:

$$P(m) = \begin{pmatrix} m+n-1 \\ m \end{pmatrix} p^n q^m \tag{11}$$

3. As scholar Meni Rosenfeld's analysis [1], we can have the visualization of the relationship between r and q, as shown in figure 1 below.

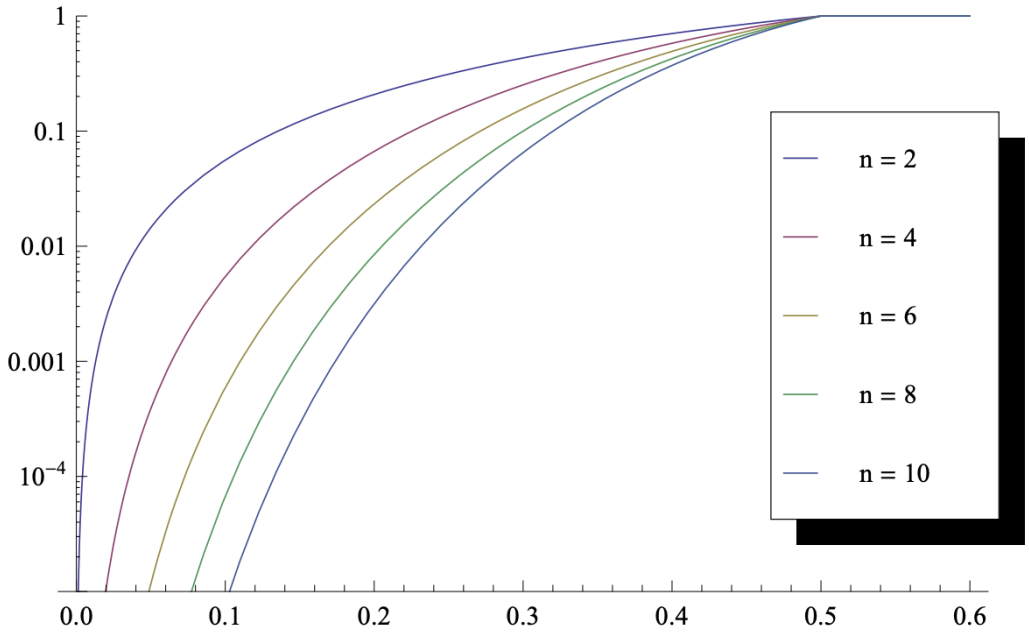

Figure 1: Relation bewteen r and q.

Also, in order to practice and get a better understanding of double spending problem, I use Matlab to simulate the relationship between r and q, the simulation is shown below:

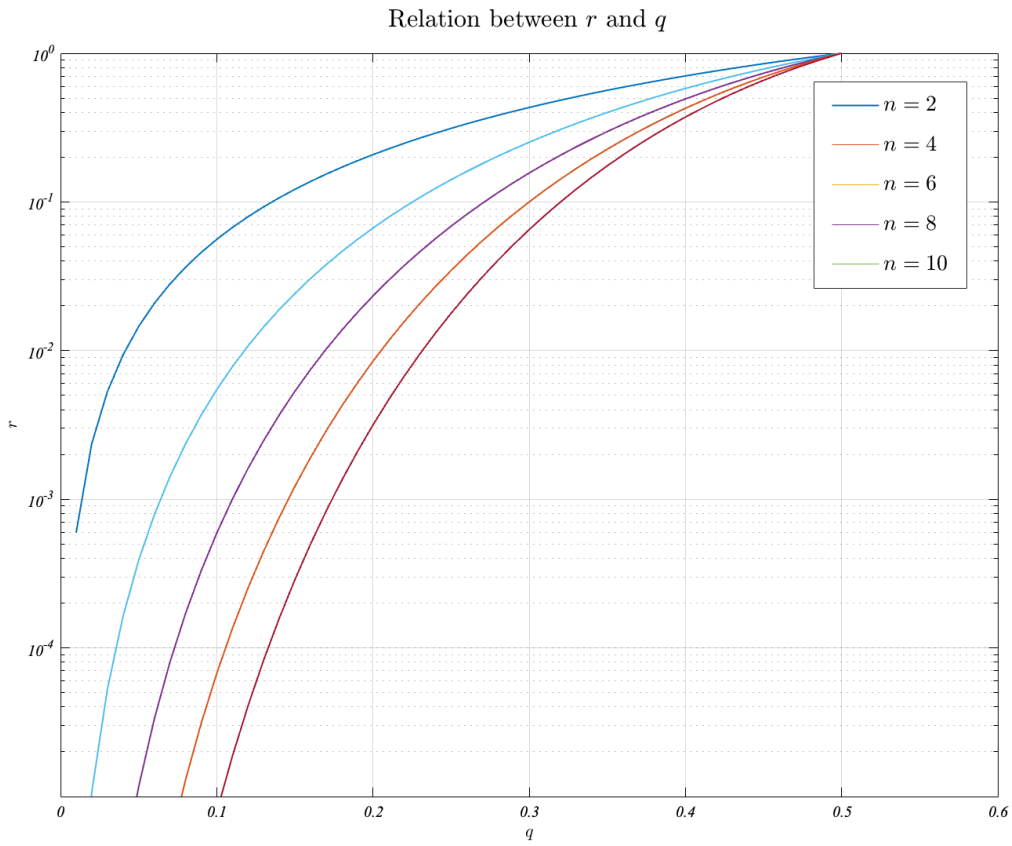

Figure 2: Relation bewteen r and q.

Here is the main function of the r-q relation visualization.

```
function f = prob(n)
    q = [0: 0.01: 0.5]
    p = 1 - q
    r = 1
    for m = 0: n
        r = r - factorial(m+n-1)/(factorial(m)*factorial(n-1))*(p.^n .* q
            .^m - p.^m .* q.^n)
    end
    semilogy(q, r)
    hold on
    q = [0.5: 0.01: 1]
    semilogy(q, 1)
end
```

Problem 4 - Bitcoin Blacklisting Attack
Suppose a mining pool wants to blacklist transactions from address X. In other words, they want to freeze the money held by that address, making it unspendable.

1. (Punitive Forking) The mining pool announces that they will refuse to work on a chain containing a transaction originating from address X. Explain why this strategy can guarantee that the blacklisted transactions will never be published if the mining pool has the majority of the hash power.

2. (Feather Forking) The mining pool announces that they will attempt to fork if they see a block that has a transaction from address X, but they will give up after the transaction from address X has k confirmations. The success of this attack depends entirely on the motivation of other miners to join the attacker. If a miner includes a transaction from address X in his block, he will receive block reward plus transaction fee from address X. Otherwise, the miner only receives block reward. Suppose the attacker controls $q = 20\%$ of the network hash power. Let $k = 2$ and block reward be 12.5BTC $\approx \$48,550$. What is the minimum transaction fee address X has to pay in order to avoid being blacklisted? (Hint: first find out the probability that the attacker successfully prunes the block containing a transaction from address X.)

Answer:

1. If a transition from address X is included, a fork will be created and since the mining pool has the majority of the hash power, the mining pool will soon create a longer chain which will invalidate the chain contains the transition from address X.

   And the miners, who know the fact mentioned above will no longer try to include the transition from address X.

2. Since $k = 2$, we can obtain the probability that the attacker will succeed in build the block without help from other miners is q^2. Therefore, to avoid being blacklisted, the expected payment amount is

$$\mathrm{E} = (1 - q^2)(Block\ Reward + Transaction\ fee) \tag{12}$$

   And the amount should be greater than the reward other miners could have mined from other blocks.

   Therefore,

$$\mathrm{E} = (1 - q^2)(Block\ Reward + Transaction\ fee) \geq Block\ Reward \tag{13}$$

   Therefore, we can calculate the minimum transaction fee the attacker needs to pay, which is:

$$Transaction\ Fee \geq \frac{Block\ Reward}{0.96} \approx 0.521(BTC) \tag{14}$$

   Therefore, the minimum transaction fee is 0.521 BTC.

Problem 5 - Bitcoin Selfish Mining
Consider the selfish mining problem in Bitcoin. Notations are given in Lecture 3, Slide 35.

1. Describe all events that cause each transition in the state transition diagram in Lecture 3 , Slide 36 .
2. At which transitions in the state transition diagram in Lecture 3, Slide 36, the honest miners would earn the block reward, and how many?

Answer:

1. Based on the Markov Chain in the lectures notes shown below,

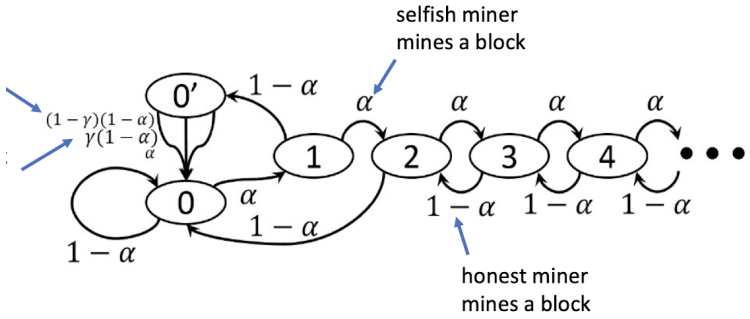

Figure 3: Markov Chain for selfish minning problem in Bitcoin.

We can obtain the detailed transition events.

START	***END***	***Probability***	***Description***
0'	0	$(1-\gamma)(1-\alpha)$	honest miner mines a block on the main branch
0'	0	$\gamma(1-\alpha)$	honest miner mines a block on the selfish miner's block
0'	0	α	selfish miner mines a block
0	0	1-α	honest miner mines a block
1	0'	$1-\alpha$	honest miner mines a block on the main branch, selfish miner publishes his private branch containing one block
2	0	$1-\alpha$	honest miner mines a block on the main branch, selfish miner publishes his private branch containing one block
n	n+1	α	selfish miner mines a block
n	n-1	$1-\alpha$	honest miner mines a block

2. The table for detailed reward of honest miners is shown below.

START	***END***	***Probability***	***Description***	***Reward Num***
0'	**0**	$(1-\gamma)(1-\alpha)$	**honest miner mines a block on the main branch**	**2**
0'	**0**	$\gamma(1-\alpha)$	**honest miner mines a block on the selfish miner's block**	**1**
0	**0**	$1-\alpha$	**honest miner mines a block**	**1**

References

[1] Meni Rosenfeld. Analysis of hashrate-based double spending, 2014.