Question 1 [easy formula questions related to gas etc]

Suppose block # 100,000 has a base fee per gas of 12.3 Gwei per gas. Alice makes a transaction that consumes 90,000 gas. She sets the max priority fee per gas as 4.0 Gwei and max fee per gas as 30 Gwei.

(a) Assuming that this transaction is valid and transfers no Ethers, how much Ethers does Alice lose for making this transaction?

(b) Assume that block number #100,000 has a total gas of 17M. What is the % gas used and % gas target?

(c) Assume that block # 100,001 consumed 20M gas, #100,002 consumed 22M gas, # 100,003 consumed 17M gas, # 100,004 consumed 30M gas, # 100,005 consumed 27M gas, # 100,006 consumed 25M gas, # 100,007 consumed 24M gas, # 100,008 consumed 30M gas, # 100,009 consumed 29M gas, # 100,010 consumed 28M gas, # 100,011 consumed 29M gas.

Alice tries to publish the same transaction (with a valid transaction nonce). Which block will the latest block that can accomodate this transaction?

Question 2 [Something on MPTs, also easy marks in my opinion]

Simplified World State								
Address								Ethers
1	2	a	c	4	d	1	8	0.2
6	4	2	3	4	d	1	8	0.4
3	f	2	d	1	9	e	c	0.6
3	f	2	d	1	0	d	a	1.1
4	1	2	7	5	c	e	a	0.7

Question 3 [PBFT] Let's suppose there are 10 servers of which 1 is malicious and remains silent. A client makes a request to the primary server.

(a) How many messages are transmitted in the pre-prepare stage?

(b) How many messages are transmitted in the prepare stage?

(c) How many messages are transmitted in the commit stage?

(d) What is the maximum number of faulty nodes that this system can tolerate?

Question 4 [Epochs and PoS] Suppose that epoch 0 has blocks 0-31, epoch 1 has blocks 35-60, epoch 2 has blocks 64-80, epoch 3 has blocks 98-110. What are the source and target blocks for block number 35, 64 and 110?

Question 5 [Understanding based stuff] Suppose Alice and Bob want to play a weird lottery game. They both write a smart contract and deposit 5 Ethers each into this smart

contract. After both Alice and Bob have deposited 5 Ethers into this smart contract, either Alice or Bob must make a transaction that **randomly** assigns the total 10 Ethers to either Alice or Bob.

(a) Everything on the blockchain is deterministic. One technique to get "random" data is to use the previous block hash as a random number. Why is this approach a bad approach if we want to be fair to Alice and Bob?

(b) How could Alice and Bob (who do not trust each other) modify the smart contract to be fair? Precisely define your scheme.

Question 6 [Understanding based stuff] Let's suppose you want to start mining Bitcoin immediately and independently (without joining a pool). Nowadays, mining bitcoin takes time – not just because of the difficulty of finding a hash. It can take a few months for your miner to synchronize to the blockchain and construct a UTXO database.

You decide to skip creating the UTXO database entirely. You come up with a new scheme.

a. You download the entire Bitcoin blockchain (which will take a lot, lot, less than a few months).

b. You look in your mempool for a transaction.

c. You verify the signature of the transaction etc.

d. Now, you must find the parent UTXOs for this transaction and confirm they are unspent. In-order to do this, you do:

   - You start back-tracking in the blockchain until you find the parent UTXO. Along the way you make sure that no transaction is referring to the same UTXO.
   - If you find the parent UTXO and you are the only transaction referring to the parent UTXO, you can do further checks on the transaction (input$\geq$output) etc.

e. You add the output of any valid transaction you find (and put in a block on the blockchain) to a new UTXO database that is initially empty.

Will this approach work? What are the downsides of this approach compared to the conventional approach in which you create a UTXO database first? What assumptions will be necessary about the types of transactions you are getting that ensure that this approach is better than the conventional approach?