In=25 March 11:45pm; Out=11 March

*You may partner with up to four others (**a group of max 5**) to submit a single write up in a single group. We encourage discussion with other students in class, even if they are not in your group. Each student must build a good understanding of all the questions even if they discuss and collaborate with others. Merely dividing the problems among group members and putting those together before submission will severely impact your learning, and we **strongly advice against it**. Furthermore, **blatant copying** from online or other resources is **forbidden**. If there are confusions or questions, post those on Piazza or see your TA or Instructor.*

Submission Instructions: *You must submit a PDF on LMS in the appropriate tab. This includes hand-written content (which you may scan and upload). There will be NO late days or extentions. Please write down the names and roll numbers on a front page. Make only one submission per group. We will NOT accept any hard-copy submissions.*

Question 1 [4+2 marks]

a) What design decisions went into making Bitcoin?

b) What are the consequences of these design decisions with regards to scalability?

Question 2 [3+3 marks] Suppose Taleef started a blockchain, Dripcoin which is exactly the same as the Bitcoin block (with the same average block generation time), except that the blocksize in Dripcoin is 4MB. Please use 1M = 10^6 for all calculations.

a) Assuming one transaction is 50B, what is the average transaction per second (TPS)?

b) Suppose Bitcoin started on day 0 and Dripcoin started on day 8000. On what day is the size of the Dripcoin blockchain greater than that of Bitcoin? (Bitcoin block size = 1MB)

Question 3 [3 marks] What class of numbers would you input into ScriptSig to unlock the following ScriptPubKey:

```
3
OP MUL
5
OP ADD
OP DUP
OP MUL
2
OP MOD
```

Question 4 [3 marks] Write a Bitcoin ScriptPubKey that unlocks if you input a number x such that $5x^3 + 3x^2 + 32 = 59$.

Question 5 [5 marks] Suppose there is a UTXO with a value of 1 BTC that is locked to Ahsan. Ahsan sends Hashim 0.4BTC. Ahsan then sends Hasnain 0.3 BTC. Hasnain further sends 0.2BTC to Hashim. Assuming there were no other transactions (or fees), no other UTXOs, and that they did not send any change to the miner but to themselves:

a) What is the total number of UTXOs at the end?

Question 6 [2+3+14 marks] Mustafa is a Bitcoin miner he mined a Bitcoin block with a block number of 500,000 and a bits field of 402,691,653.

a) What is the target hash? How many leading zeros are there?

b) How many tries does he need on average?

c) What is the block difficulty?
For the following parts, assume that the average machine can compute 5TH/s. It also has a power consumption of 4000W. Also assume that the average Bitcoin block has a size of 1 MB and takes 10 minutes to be mined on average.

d) How many total machines are there?

e) What is the yearly energy consumption of these machines? (assume all months = 30 days)

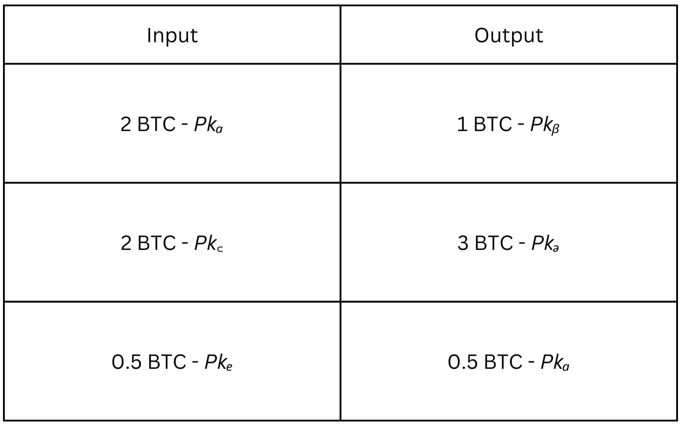
f) Assuming that average size of a transaction in Bitcoin is 200 bytes, what is the average power consumed to mine a transaction in KWh?

g) Assume that the cost of electricity is $0.1 per KWh around the world. Also, the reward for mining a Bitcoin block is 12.5 bitcoin (transaction fees are negligible). What is the minimum price of bitcoin (in $) for it to be worthwhile to get a mining rig?

h) Suppose that it took the network 1173251 seconds to mine the previous 2016 blocks. The difficulty gets adjusted in block 501984 automatically. What is the new difficulty?

Question 7 [5 marks] If Mustafa wants to buy some very expensive khokha chai from khokha-Uncle (for 20 BTC). khokha-Uncle is requesting a Bitcoin advance before sending the chai. But, Mustafa is concerned that if he sends the BTC and khokha-Uncle refuses to give the pricey khoka chai, he will lose the BTC he sent. In the event of a dispute, khokha-Uncle and Mustafa both trust Nameer to break the tie and award the BTC to the rightful party. Create a scriptPubKey that allows them to do this.

Question 8 [5 marks] Public keys are used on the Blockchain to somewhat protect the anonymity of senders and receivers. However, it is often the case that one uses one transaction to extrapolate the identity of multiple users. For example, if you pay Najeeb in Bitcoin to give you extra marks, Najeeb now knows your public key is tied to you in particular and can see what other transactions you have made. He can now start guessing the recipients public keys too, using the information that you were the sender. This is why people try to keep various public keys. Consider the following Bitcoin transaction:



Input	Output
2 BTC - Pk_{a}	1 BTC - Pk_{β}
2 BTC - Pk_{c}	3 BTC - $Pk_{\text{ə}}$
0.5 BTC - Pk_{e}	0.5 BTC - Pk_{a}

Let's say you are a Bitcoin detective who wants to identify how Jafri spends his money, and know that Pk_a (Public Key A) surely belongs to Jafri. What are the possible scenarios this transaction is showing you (think of who could be behind the senders and receivers, and why this is the case)? Give at least 5 scenarios. Secondly, do you think there are actor(s) (Jafri or anyone else) with more than one public key in this transaction? Why/why not?

Question 9 [5 marks] Mining is a process in which we try to find a hash value that meets a certain criteria (less than the current difficulty of the blockchain) by changing the nonce value which changes the hash value of the block. There are on average 2500 transactions in a Bitcoin block. We can just change the order of those transactions and it will also produce a new hash value (avalanche effect). Then why do we still need the nonce value field? Give an example.

Question 10 [15 marks] Your TA Ahsan wants to design a ScriptPubKey. He wants to design a ScriptPubKey in a way that takes three numbers that gets unlocked if three numbers in the scriptSig are *pythogorean triples* and in **no** other circumstance.

Note: The input is pushed to the stack (a b c) and the starting state of the stack is c (top of the stack) b a (bottom of the stack).

Question 11 [5 + 5 marks] Bitcoin uses and stores the merkle tree root hash in the block header. You are Bitcoin miner and while mining a block you want have to make a merkle tree and add it's root hash to the header. Suppose you want to add 4 transactions to the block you want to mine (transactions are shown below).

txn1 = "10 Bitcoins from Class to Ahsan"
txn1 = "1 Bitcoin from Ahsan to Humza"
txn1 = "0.5 Bitcoins from Ahsan to Ayesha"
txn1 = "5 Bitcoins from Ahsan to Sara"

a) You are required to make a merkle tree as discussed in class using the above four transactions. Explain every step while making the tree e.g how you are concatenating the hashes. Also write the hashes of every node inside that node (you can shorten the hash such as 123...345). Note: Use SHA256 for calculating the hashes

b) Suppose another miner Fatima wants to verify a transaction and she has your merkle tree. What is the time complexity of this operation and what hashes will she need to compute to verify that transaction. Give an example form you tree in part a to get full marks.

Question 12 [12 marks] Bitcoin does not allow immediate transaction finality. That means that you can't be sure any transaction your receive is valid until it is in a block with 4-5 confirmations.

However, in certain circumstances, it is possible to have ,*instantaneous* transaction finality *after* one transaction has been confirmed. Moever, all of this happens off-chain (except the initial transaction). Read up on the bitcoin lightning network and explain how it allows this? You should write down the script and explain how Alice and Bob may do instantaneous transactions.

Question 13 [12 marks] Bitcoin allows you to define the script that must be satisfied by using only its hash. This is a scheme called Pay-to-Script-Hash. Explain how this scheme works and give an example of a case in which Alice wants to lock 5BTC to Bob's pubkey.

Question 14 [4+4+4+8 marks] These questions are related to bitcoin signatures:

a) When computing the signature, why is signing *just* the parent transaction a bad idea? Bitcoin allows you to pick which parts of a transaction should be signed. These are specified with particular "sighash" flags.

b) What does $SIGHASH\_SINGLE$ do and when is it useful?

c) What does $SIGHASH\_NONE$ do and when is it useful?

d) You want to gather 50BTC in order to buy food for your entire workspace. You know the (very public) bitcoin address of a popular restaurant that all your colleagues would be willing to eat from. However, you do not have 50BTC – you must ask your colleagues (such as Bob) to donate. Bob, however, is worried that he will donate the BTC to the restaurant and he will lose his bitcoin if the total 50BTC required are not collected. This is because the transaction is irreversible and the restuarant may not be trusted to return the Bitcoin. Considering this, how will you ensure that Bob gets his bitcoin back if the total goal of 50BTC is not reached?

Question 15 [5 marks] In bitcoin, the difficulty is the ratio of the average number of tries required for a block compared to the average number of tries required for the genesis block. What would be an analogue in Ethereum (which relies on Proof-of-Stake)?