Blockchain Foundations
Midterm Exams
Spring 2026

1. The exam has 4 questions with a total of 100 points. You have 80 minutes to take the exam. Questions have different numbers of points so please allocate your time to each question accordingly.
2. Please write the answer in the designated area underneath each question. If you need more room for your answer, please indicate as such, and continue your response on the blank page at the end of all questions. No additional pages will be allowed.
3. Scratch paper will be provided and collected at the end of the exam, but will not be graded.
4. All answers should be justified, unless otherwise stated.
5. The exam is closed book but you are allowed one double-sided sheet of notes. A list of variables and reference information is provided at the end of the exam. No other materials are allowed.

Good luck!

Name:
Student ID:

(22 points) Problem 1

For the following questions, choose the one most fitting answer among the four choices. No justifications are required for this problem. 2 points for a correct answer, 0 points for an incorrect answer. 0.5 point for leaving the answer blank. Knowing you don't know something has value.

1. In a system with constant n, t, Δ, q, which reconfiguration increases the density of honestly successful queries in the unit of time?
   (a) Increase k
   (b) Decrease k
   (c) Increase T
   (d) Decrease p
2. If an adversary alters the public key of a coinbase transaction of a proof-of-work protocol in transit during gossiping, this will invalidate:
   (a) The signature on the coinbase transaction
   (b) The signature on the block
   (c) The suitability of the block nonce
   (d) The weak conservation law
3. In the UTXO model, an input of a transaction tx contains an outpoint that always points to:
   (a) A coinbase transaction tx'
   (b) A tuple (blockid, index) of the previous block on the chain
   (c) The output of a different transaction tx'
   (d) The output of a different transaction tx', or an output of that same transaction tx
4. At the epoch boundary, if the difference between the timestamp recorded on the tip and the timestamp recorded m blocks ago is larger than the expected epoch time duration:
   (a) The parameter p is indirectly increased by adjusting the target.
   (b) The parameter Δ is decreased so that more blocks have sufficient time to be transmitted across the diameter of the gossiping network.
   (c) The parameter k is decreased to maintain the same liveness level, since successful queries are becoming rarer.
   (d) A request is placed towards the honest participants to increase their q value so that mining power is appropriately increased to keep f constant.
5. A *correct* signature scheme must necessarily:

(a) Ensure the adversary cannot guess the secret key with non-negligible probability.

(b) Ensure the adversary cannot forge a signature on a new message that she did not request from the oracle except with negligible probability.

(c) Ensure the honest party can verify honestly produced signatures on messages that start and end with the 0 bit.

(d) Ensure that the key generation function is collision-resistant.

6. When a rational miner holds a mempool that is smaller than the block size limit:

   (a) He includes all the transactions of his mempool into the new block.

   (b) He waits for the mempool to fill up before he begins mining a block, to save on mining resources.

   (c) He includes only the transactions with the top fee/byte score into the new block.

   (d) He places half of the transactions in the current new block, and saves the rest for the next block to minimize the variance of his profits.

7. If the recent chunks of an honestly adopted chain have small chain quality:

   (a) Ledger safety has been violated.

   (b) Common Prefix has been violated for sufficiently large values of k.

   (c) Chain Growth must have recently dropped to a velocity of $\tau = 0$.

   (d) Ledger liveness may be deteriorating.

8. How can a non-temporary probabilistic polynomial majority adversary spend an honest party's money?

   (a) By issuing coinbase transactions that violate the system's macroeconomic policy.

   (b) By issuing double spending transactions that spend the honest party's money, while mining using the Nakamoto Race to ensure a ledger safety violation.

   (c) By creating a transaction that spends the honest party's money and ensuring it is confirmed in a timely manner.

   (d) She cannot.

9. The fan-out situation, in which numerous forks are created even though there is an honest majority, occurs in a proof-of-work system when:

   (a) The network delay Δ is extremely large.

   (b) The system is misconfigured with a too high target T.

   (c) The system is misconfigured with a too high query success probability p.

   (d) All of the above.

10. If the k Common Prefix chain virtue is violated in a proof-of-work longest chain protocol, it *must* be that:

(a) Ledger safety has been violated.
(b) Ledger liveness has been violated.
(c) Chain quality of recent chunks has been reduced to $\mu = 0$.
(d) None of the above.

11. A majority selfish miner, over the long run, can achieve a chain quality of:

(a) $\mu = 0$
(b) $\mu = \frac{1}{3}$
(c) $\mu = \frac{1}{2}$
(d) $\mu = 1$

(23 points) Problem 2

Let $H_\kappa : \{0,1\}^* \longrightarrow \{0,1\}^\kappa$ be a family of collision-resistant hash functions. Define $G_\kappa : \{0,1\}^* \longrightarrow \{0,1\}^\kappa$ to be the family of hash functions obtained by invoking $H_\kappa(x)$, flipping the last bit of $H_\kappa(x)$, and then returning the result. Namely, $G_\kappa(x) = H_\kappa(x)[:-1] \,\|\, (H_\kappa(x)[-1] \oplus 1)$. Examine whether the function family G_κ is collision-resistant.

Hint: If you choose to prove collision resistance, use a computational reduction. We suggest that you write out the exact pseudocode of the reduction and calculate the relationship of the probabilities of success between the adversaries. If you choose to disprove collision resistance, construct a pathological case in which H_κ is collision-resistant, but G_κ is not collision-resistant, and prove the latter assuming the former.

(25 points) Problem 3

We are working in a UTXO longest chain system with a block reward of 50 units and a confirmation rule of $k = 6$. Consider the transaction graph illustrated in Figure 1. Transaction ids are displayed in the circles, values of outputs are written above the outputs, and owners of outputs are written below the outputs. Any outputs with no owners indicated belong to the adversary. Transactions with no inputs are coinbase transactions. For the coinbase transactions, you can assume that no "height" is needed to be included in them for validity, and that there is no maturation restriction, so they can be spent within the same block. In the questions below, please denote your outpoints in $(\mathsf{txid}, \mathsf{index})$ notation.

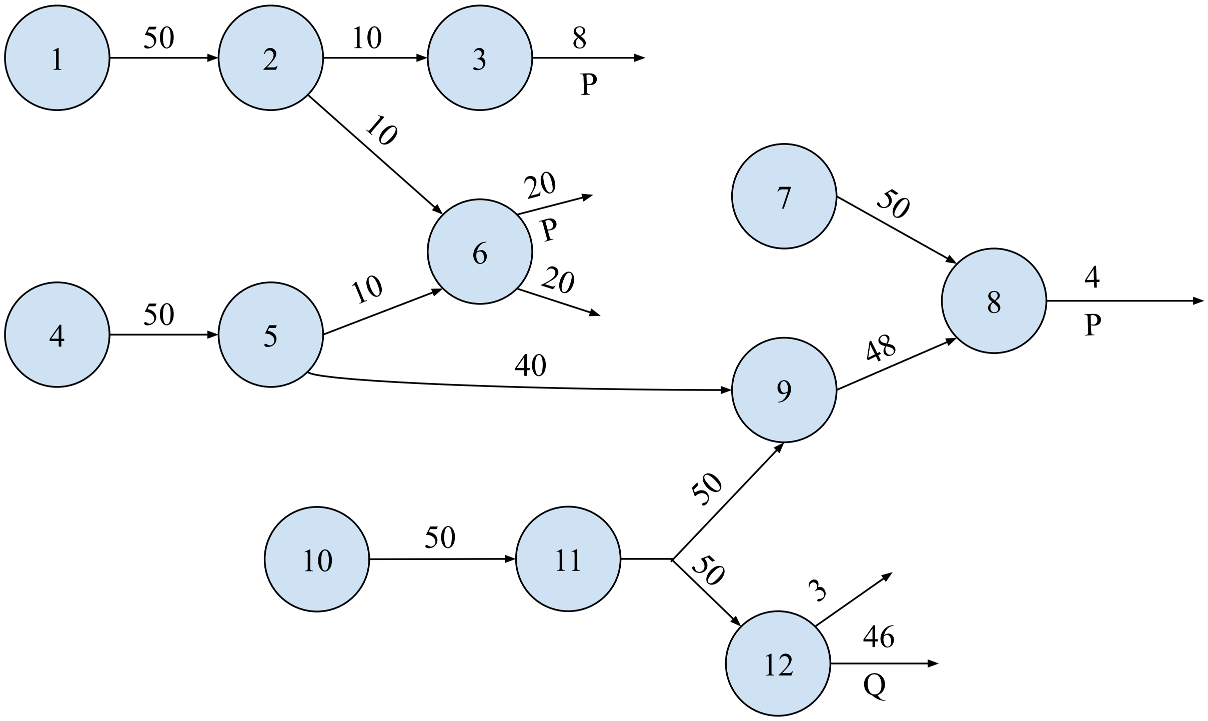

Figure 1: The transaction graph.

Remark. The second input to transaction 9 and the first input to transaction 12 are the same outpoint $(11, 0)$.

1. (7 points) Honest party P has adopted a chain $C_P = (\mathcal{G}, B_1, B_2, B_3)$. Genesis has no transactions, block B_1 only contains transaction 1, block B_2 only contains transaction 4, and block B_3 only contains transaction 10. Party P receives the transactions in the order 2, 3, 5, 6, 7, 8, 9, 11, 12 from the network. The party does not receive any new blocks for now. According to the state derived from P's new mempool, what is the UTXO set and how much money do P and Q have respectively?

2. (7 points) A rational party R (who may play dishonestly and is trying to maximize his profits) has also adopted the same chain $C_R = C_P$ and attempts to mine a block on top of $C_R[-1]$, having seen all the 12 transactions in the graph. The block size limit is such that up to 5 transactions (1 coinbase + 4 other transactions) can fit per block. Which transactions will R choose to include in the block he creates? According to the state derived from R's newly mined block B_4, how much money does R have?

3. (5 points) After block B_4 is mined, another 10 *empty* blocks get mined and broadcast sequentially by the adversary on top of B_4 and P adopts the chain C'_P containing all of them. What is the k-confirmed ledger according to P?

4. (6 points) Assuming no further blocks or transactions have arrived in the meantime, what is P's new mempool now?

Extra page for answers

(30 points) Problem 4

Consider the blocktree illustrated in Figure 2. The picture shows all the blocks that have been successfully mined in a static difficulty proof-of-work longest chain protocol during the time interval $[0, 31]$. The chains C_1, C_2, C_3, C_4, C_5 are marked at their tips on the diagram. Block $C_1[0]$ is the genesis block. Blue blocks have been mined by an honest party, whereas red blocks (blocks at times $3, 8, 8, 17, 20, 25, 25$) have been mined adversarially. Within each box, the time at which the block was mined is displayed. Consider a scenario where the network delay is $\Delta = 1.5$, there are $n = 5$ parties and $q = 1000$ is the mining power. We are using a $\kappa = 256$ bit function to mine. The adversary is constantly mining with her maximum mining power. The confirmation and Common Prefix parameter is $k = 2$.

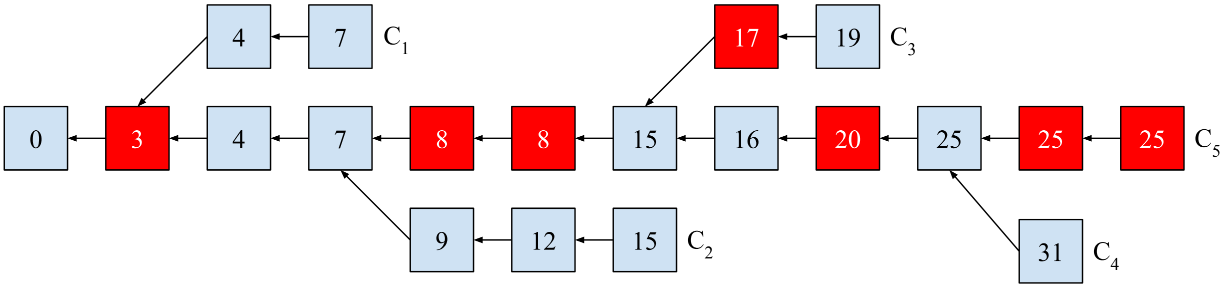

Figure 2: The blocktree graph.

1. (4 points) How many successful queries have occurred? Of those, which ones are convergence opportunities?

2. (4 points) Calculate the chain quality of C_1 and C_5.

3. (7 points) Draw an example timeline during which Common Prefix with $k = 2$ was violated. In this timeline, highlight when each party received each block. Write out the concrete predicate describing Common Prefix that was violated. At which times was it violated, and between the chains of which parties? Why had the particular honest parties adopted these tips at the particular times, and which other chains had they seen at those times?

4. (7 points) For the above Common Prefix violation, describe an example with the ledgers and transactions adopted by the honest parties in which safety was violated. Write out the concrete predicate describing safety that was violated. During which times was safety violated and among which honest parties? What was the ledger of each honest party at those times?

5. (8 points) Based on the observed blocktree, give your best estimation for the values of t and T. You should calculate the exact values for t and T by plugging in the numbers, but you don't need to do the final numerical calculation if it requires a calculator.

Extra page for answers

Reference

Some helpful definitions are provided below.

Definition 1 (Collision Resistance). *A hash function* $H : \{0,1\}^* \longrightarrow \{0,1\}^\kappa$ *is* collision resistant *if for all PPT adversaries* $\mathcal{A}$*,*

$$\Pr[\textsf{\textit{collision-game}}_{H,\mathcal{A}(\kappa)} = 1] = \mathrm{negl}(\kappa)\,.$$

The game is defined in Algorithm 1.

Algorithm 1 The collision-finding game for a hash function H.

```
function COLLISION-GAME_{H,A}(κ)
    $x_1, x_2 \leftarrow \mathcal{A}(1^\kappa)$
    return $H_\kappa(x_1) = H_\kappa(x_2) \wedge x_1 \neq x_2$
end function
```

Definition 2 (Correct Signature). *A signature scheme* $(\textsf{\textit{Gen}}, \textsf{\textit{Sig}}, \textsf{\textit{Ver}})$ *is* correct *if, for all* $m \in \{0,1\}^*$*, whenever* $(sk, pk) \leftarrow \textsf{\textit{Gen}}(1^\kappa)$*, we have that* $\textsf{\textit{Ver}}(pk, m, \textsf{\textit{Sig}}(sk, m)) = 1$*.*

Definition 3 (Secure Signature). *A signature scheme* $(\textsf{\textit{Gen}}, \textsf{\textit{Sig}}, \textsf{\textit{Ver}})$ *is* secure *if for all PPT adversaries* $\mathcal{A}$*,*

$$\Pr[\textsf{\textit{existential-forgery-game}}_{(\textsf{\textit{Gen}}, \textsf{\textit{Sig}}, \textsf{\textit{Ver}}),\mathcal{A}}(\kappa) = 1] = \mathrm{negl}(\kappa)\,.$$

The game is defined in Algorithm 2.

Algorithm 2 The existential forgery game for a signature scheme (Gen, Sig, Ver).

```
function existential-forgery-game_{(Gen,Sig,Ver),A}(κ)
    $(pk, sk) \leftarrow \mathsf{Gen}(1^\kappa)$
    $M \leftarrow \emptyset$
    function $\mathcal{O}$(m)
        $M \leftarrow M \cup \{m\}$
        return $\mathsf{Sig}(sk, m)$
    end function
    $m, \sigma \leftarrow \mathcal{A}^{\mathcal{O}}(pk)$
    return $\mathsf{Ver}(pk, \sigma, m) \wedge m \notin M$
end function
```

Definition 4 (Weak Conservation Law). *A transaction* tx *satisfies the Weak Conservation Law if*

$$\sum_{i \in \mathsf{tx}.\textsf{\textit{ins}}} i.v \geq \sum_{o \in \mathsf{tx}.\textsf{\textit{outs}}} o.v\,.$$

Definition 5 (Velocity). *The* velocity τ *of a chain of an honest party* P *between times* $r_1 < r_2$ *is the ratio* $\frac{|\mathcal{C}^P_{r_2}|-|\mathcal{C}^P_{r_1}|}{r_2-r_1}$.

Definition 6 (Common Prefix). *A system is said to satisfy* Common Prefix *with parameter* $k \in \mathbb{N}$ *if for all honest parties* P_1, P_2 *and for all times* $r_1 \leq r_2$*, the chains adopted by the honest parties satisfy the property that*

$$\mathcal{C}^{P_1}_{r_1}[:-k] \preceq \mathcal{C}^{P_2}_{r_2} \,.$$

Definition 7 (Chain Quality). *A system is said to satisfy* Chain Quality *with parameters* $\ell \in \mathbb{N}, \mu \in [0,1]$ *if for all honest parties* P *and all times* r*, for all* $i, j \in \mathbb{N}$ *such that* $j - i \geq \ell$*, we have*

$$\frac{|\mathcal{H}(C^P_r[i{:}j])|}{j-i} \geq \mu \,.$$

Definition 8 (Chain Growth). *A system is said to satisfy* Chain Growth *with parameters* $s \in \mathbb{N}, \tau \in \mathbb{R}^+$ *if for all honest parties* P *and all times* $r_1 \leq r_2$ *such that* $r_2 - r_1 \geq s$*, we have*

$$|C^P_{r_2}| - |C^P_{r_1}| \geq \tau(r_2 - r_1) \,.$$

A ledger L is a sequence of transactions.

Definition 9 (Ledger Liveness (informal)). *A ledger system is* live *if, whenever an honest party attempts to inject a transaction to the ledger, then this transaction will make it to all honest ledgers "soon".*

Definition 10 (Ledger Safety). *A ledger system is* safe *if, for all honest parties* P_1, P_2 *and for all times* r_1, r_2*, it holds that* $L^{P_1}_{r_1} \preceq L^{P_2}_{r_2}$ *or* $L^{P_2}_{r_2} \preceq L^{P_1}_{r_1}$.

Our variables.

- κ: The security parameter
- $\mathcal{A}$: The uniform PPT adversary
- Π: The honest protocol
- H: The hash function
- $\mathcal{G}$: The genesis block, an *honestly* mined reference block with 0 height
- Δ: The maximum network delay
- T: The mining target
- p: The probability of a successful query
- n: The total number of parties (includes both honest and adversarial)
- t: The number of adversarial parties

- q: The hashing power of a single party per unit of time
- k: The Common Prefix parameter, in blocks
- μ: The Chain Quality parameter, as a proportion
- τ: The velocity, in blocks per unit of time
- m: Epoch duration, in blocks

Terminology.

- The proof-of-work inequality: $H(B) < T$.
- A *query* is a fresh call to the hash function H.
- A *successful query* is a fresh (honest or adversarial) query to the random oracle H that satisfies the proof-of-work inequality.
- A *convergence opportunity* is an *honest* successful query which is spaced at least Δ apart from all other *honest* successful queries. The genesis block was computed during a convergence opportunity.
- A *negligible function* is eventually smaller than all inverse polynomials.
- A block tree has the *Common Prefix* virtue with parameter k if, for any two chains C_1, C_2 currently adopted by honest parties, $C_1[:-k]$ is a prefix of C_2.
- The *k-confirmation rule* reports the transactions in $C[:-k]$ as confirmed.
- The xor operator $b_1 \oplus b_2$ between two bits b_1 and b_2 returns 1 iff $b_1 \neq b_2$.

Algorithms.

Algorithm 3 The mining algorithm.

```
function Mine($s, \overline{x}$)
    $ctr \xleftarrow{\$} \{0,1\}^\kappa$
    while true do
        $B \leftarrow s \,\|\, \overline{x} \,\|\, ctr$
        if $H(B) < T$ then
            return $B$
        end if
        $ctr \leftarrow ctr + 1$
    end while
end function
```

Chain addressing notation.

- $|\mathcal{C}|$: Chain length

- $\mathcal{C}[i]$: i^{th} block in the chain (0-based). The block height is i.
- $\mathcal{C}[-i]$: i^{th} block from the end.
- $\mathcal{C}[0]$: Genesis (by convention honest).
- $\mathcal{C}[-1]$: The tip.
- $\mathcal{C}[i{:}j]$: Chain chunk from block i (inclusive) to j (exclusive).
- $\mathcal{C}[{:}j]$: Chain chunk from the beginning and up to block j (exclusive).
- $\mathcal{C}[i{:}]$: Chain chunk from block i (inclusive) onwards.
- $\mathcal{C}[{:}-k]$: The stable chain.