

**Blockchain Foundations**  
**Practice Midterm 1**  
**2026**

1. (24 points) True-false questions (no explanations required). 2 points for a correct answer, 0 points for an incorrect answer. 1 point for leaving the answer blank. Knowing you don't know something has value.

Please shade your answer in completely to receive full credit.

- |  | T                        | F                        |
|--|--------------------------|--------------------------|
| (a) While two conflicting transactions cannot both appear in the same valid block, they can appear in different blocks of a valid chain.   | <input type="checkbox"/> | <input type="checkbox"/> |
| (b) A correctly parametrized proof-of-work inequality ensures that all successful queries are always spaced at least $\Delta$ apart.   | <input type="checkbox"/> | <input type="checkbox"/> |
| (c) As the network delay $\Delta$ decreases, the mining target $T$ should be made more difficult.  | <input type="checkbox"/> | <input type="checkbox"/> |
| (d) The probability that two different honest miners choose the same nonce to mine with is negligible in $\kappa$ (see the MINE algorithm in the Reference section).                             | <input type="checkbox"/> | <input type="checkbox"/> |
| (e) The genesis block is anchored at a particular point in time in the real world by including real world data from a newspaper or other publicly verifiable source that cannot easily be faked. | <input type="checkbox"/> | <input type="checkbox"/> |
| (f) Changing the coinbase transaction public key during gossiping will fail because it will invalidate the coinbase signature.   | <input type="checkbox"/> | <input type="checkbox"/> |
| (g) While honest miners mine blocks at a bounded rate, an adversary can mine as many blocks as she likes.  | <input type="checkbox"/> | <input type="checkbox"/> |
| (h) An adversary who manages to violate ledger safety can issue a transaction spending the money of an honest party.   | <input type="checkbox"/> | <input type="checkbox"/> |
| (i) A ledger liveness violation implies a ledger safety violation.   | <input type="checkbox"/> | <input type="checkbox"/> |
| (j) An execution with $n = 1$ , $t = 0$ , $q = 1$ and $\Delta = 1$ sec has no temporary forks whatsoever.  | <input type="checkbox"/> | <input type="checkbox"/> |
| (k) In the UTXO model under longest chain rule, a block in the chain can extend multiple parent blocks.  | <input type="checkbox"/> | <input type="checkbox"/> |

2. (16 points) Let  $H_\kappa : \{0, 1\}^* \rightarrow \{0, 1\}^\kappa$  be a family of collision-resistant hash functions and define  $G_{2\kappa} : \{0, 1\}^* \rightarrow \{0, 1\}^{2\kappa}$  as  $G_{2\kappa}(x) = H_\kappa(x) \parallel H_\kappa(x)$ . Show that  $G_{2\kappa}$  is a collision-resistant family of hash functions.

3. (30 points) 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.

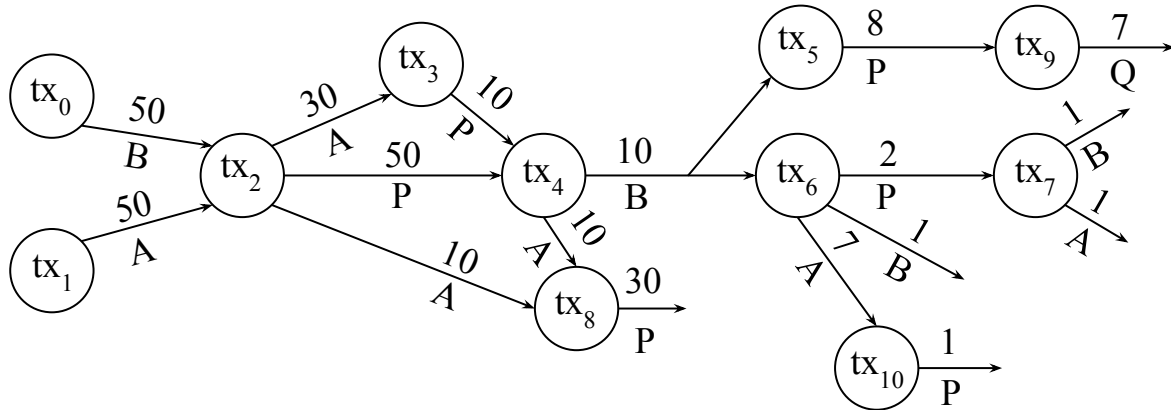


Figure 1: The transaction graph.

The party  $Q$  has adopted a chain  $C_Q$  such that  $\mathbb{L}_Q = (\text{tx}_0, \text{tx}_1, \text{tx}_2)$ , while the transactions recorded in  $C_Q$  are  $(\text{tx}_0, \text{tx}_1, \text{tx}_2, \text{tx}_3, \text{tx}_4)$ . The party  $Q$  is a miner who collects transactions into a mempool to create a template block to mine on.

- (a) (6 points) In what order should the rest of the transactions ( $\text{tx}_5$  through  $\text{tx}_{10}$ ) be arranged into a block by  $Q$  so that  $Q$ 's coinbase proceeds are maximized?
- (b) (6 points) What is the output value in  $Q$ 's new coinbase transaction?

- (c) (4 points) If  $Q$  successfully mined a block in part (a) and the block is buried under  $k = 6$  other blocks, what does the new ledger  $\mathbb{L}_Q$  report?
- (d) (6 points) Which transactions (among  $\text{tx}_5$  through  $\text{tx}_{10}$ ) are missing from  $Q$ 's ledger and why?
- (e) (4 points) How much unspent money does  $Q$  have in the system, if we also include his mining proceeds?
- (f) (4 points) How much money do the parties  $A$ ,  $B$ , and  $P$  have, in  $Q$ 's view, provided they are not mining any blocks?

4. (30 points) Consider the sequence of successful *honest* party queries in Figure 2. Recall that a successful query satisfies the proof-of-work inequality  $H(B) < T$ .

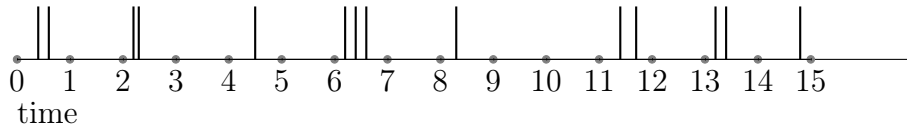


Figure 2: A sequence of honest successful query events.

Consider executions<sup>1</sup> with *maximum* network delay  $\Delta = 1$ . You are a powerful rushing adversary and you have 7 successful queries at your disposal. You are given the (fictitious) ability to place your successful queries at whichever points on the timeline you prefer. Honest parties gossip blocks, but you can schedule the delay of each honest message freely, as long as it is within a maximum of  $\Delta$ . As the adversary, you are also allowed to choose how each honest party will choose to break ties among competing chains of the same length.

Describe the following executions, consistent with the above timeline. For each of the below executions, draw the block tree. For each block in the block tree, indicate whether it was honestly or adversarially mined, and what time it was mined at. You can use just the integer part of the time (for example, you can write “1” for a block that was mined at time “1.4”). Your three executions do not all have to be different.

- (a) (10 points) An execution in which Common Prefix with  $k = 7$  is violated. What is the adopted chain tip of each honest party in your execution?

---

<sup>1</sup>An *execution* is the transcript of everything that happened, including *who* mined each block, when each block was mined, what the whole private and public blocktree looks like, when and if each block was broadcast, and when it was received, including all adversarial actions.

(b) (10 points) An execution in which three different honest parties adopt chains  $C_1$ ,  $C_2$  and  $C_3$  such that  $C_1[: -k]$ ,  $C_2[: -k]$ ,  $C_3[: -k]$  are different from each other for  $k = 4$ . ( $C[: -k]$  means the chain resulting from removing the last  $k$  blocks in a chain  $C$ .)

(c) (10 points) An execution *minimizing* Chain Quality (across all executions) of the whole chain for *some* honest party. What is the Chain Quality of the chain adopted by your chosen honest party?

Extra page for answers

# Reference

Some helpful definitions are provided below.

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

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

The game is defined in Algorithm 1.

---

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

---

```
1: function COLLISION-GAME $_{H, \mathcal{A}(\kappa)}$ 
2:    $x_1, x_2 \leftarrow \mathcal{A}(1^\kappa)$ 
3:   return  $H_\kappa(x_1) = H_\kappa(x_2) \wedge x_1 \neq x_2$ 
4: end function
```

---

**Definition 2** (Weak Conservation Law). A transaction  $\text{tx}$  satisfies the Weak Conservation Law if

$$\sum_{i \in \text{tx.ins}} i.v \geq \sum_{o \in \text{tx.outs}} o.v.$$

**Definition 3** (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

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

**Definition 4** (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

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

**Definition 5** (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

$$|\mathcal{C}_{r_2}^P| - |\mathcal{C}_{r_1}^P| \geq \tau(r_2 - r_1).$$

**Our variables.**

- $\kappa$ : The security parameter
- $\mathcal{A}$ : The uniform PPT adversary
- $\Pi$ : The honest protocol

- $H$ : The hash function
- $G$ : The genesis block, an *honestly* mined reference block
- $\Delta$ : 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
- $\mu$ : The Chain Quality parameter, as a proportion
- $\tau$ : The velocity, in blocks per unit of time

### Terminology.

- The proof-of-work inequality:  $H(B) < T$ .
- A *successful query* is a fresh 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  $\Delta$  apart from all other *honest* successful queries.
- 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  $C_2$ .

### Algorithms.

#### Chain addressing notation.

- $|\mathcal{C}|$ : Chain length
- $\mathcal{C}[i]$ :  $i^{\text{th}}$  block in the chain (0-based). The block height is  $i$ .
- $\mathcal{C}[-i]$ :  $i^{\text{th}}$  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).

---

**Algorithm 2** The mining algorithm.

---

```
1: function MINE( $s, \bar{x}$ )
2:    $ctr \stackrel{\$}{\leftarrow} \{0, 1\}^\kappa$ 
3:   while true do
4:      $B \leftarrow s \parallel \bar{x} \parallel ctr$ 
5:     if  $H(B) < T$  then
6:       return  $B$ 
7:     end if
8:      $ctr \leftarrow ctr + 1$ 
9:   end while
10: end function
```

---

- $\mathcal{C}[:j]$ : 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.