Blockchain Foundations **Spring, NTUA 2026**

Theory Exercise 2 Solutions

Due: 1:15pm, Monday, Feb 13, 2023

Problem 1

Consider the blocktree of Figure 1. Blue blocks are honestly mined blocks, whereas red blocks are adversarially mined blocks. Hypothetical blockids are shown within the squares.

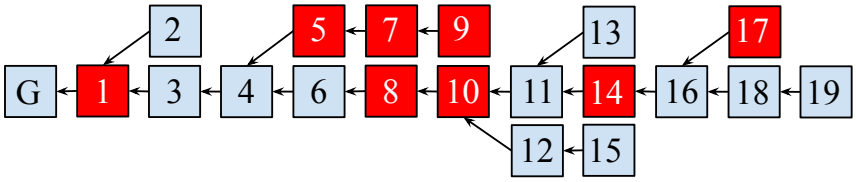


Figure 1: The blocktree.

Let $\mathcal{C}_1, \mathcal{C}_2$ indicate the chains whose tips are blocks 9 and 19 respectively.

1. What is the chain quality of $\mathcal{C}_1[-4:]$, $\mathcal{C}_1[:4]$ and $\mathcal{C}_2$ respectively?
2. Suppose an honest party had adopted $\mathcal{C}_1$ at time 100 and $\mathcal{C}_2$ at time 200. What is the velocity τ of the chain between those two times?

1. $\mathcal{C}_1[-4:]$ has 1 honest block and 3 adversarial blocks, resulting in a chain quality of $\frac{1}{4}$.
   $\mathcal{C}_1[:4]$ has 3 honest blocks and 1 adversarial block, resulting in a chain quality of $\frac{3}{4}$.
   $\mathcal{C}_2$ has 8 honest blocks and 4 adversarial blocks, resulting in a chain quality of $\frac{8}{12} = \frac{2}{3}$.
2. $\mathcal{C}_1$ has 7 blocks, and $\mathcal{C}_2$ has 12 blocks. Hence,

$$\tau = \frac{|\mathcal{C}_2| - |\mathcal{C}_1|}{200 - 100} = \frac{12 - 7}{100} = \frac{1}{20}.$$

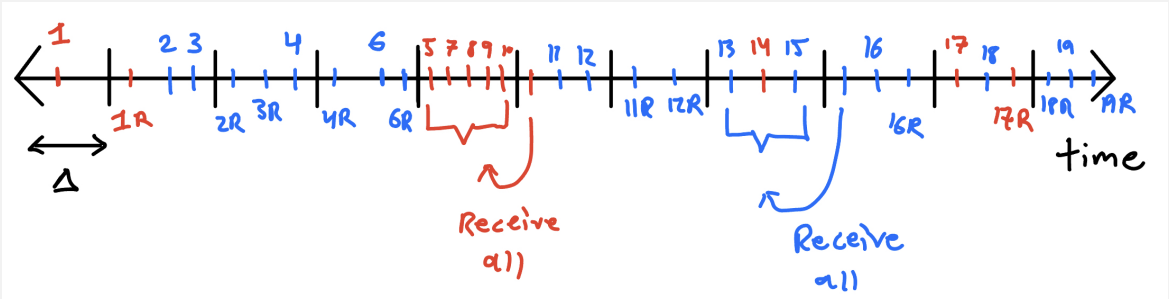
Problem 2

Consider $\Delta = 1, n = 5, t = 2$. Draw a timeline of successful queries that could have caused Figure 1 to appear. For each successful query, indicate:

1. The time at which it took place.
2. Whether the query was honest or adversarial.

3. The time at which each honest party received the block produced by the query.

For the timeline you drew, what is the *minimum* $k \in \mathbb{N}$ for which Common Prefix holds between *all* honest parties and across *all* time?



There are many possible timelines you can construct for this problem. The main idea is that your timeline should have less than Δ time between honest blocks in a fork and less than Δ time between any block and the time at which it was received by all honest parties.

In the timeline above, blue indicates honest blocks and red indicates adversarial blocks. The letter "R" after a blockid indicates the time at which the blockid was received by all honest parties. The minimum value of k for which Common Prefix holds between all honest parties and all time is $k = 3$. The longest temporary fork occurs after block 10 is received. Honest parties will disagree over whether blocks $5, 7, 9$ or blocks $6, 8, 10$ are the last 3 blocks in the longest chain.

Problem 3

Consider the UTXO transaction graph illustrated in Figure 2. Hypothetical txids are shown within the circles. The value of an output is indicated above its respective arrow.

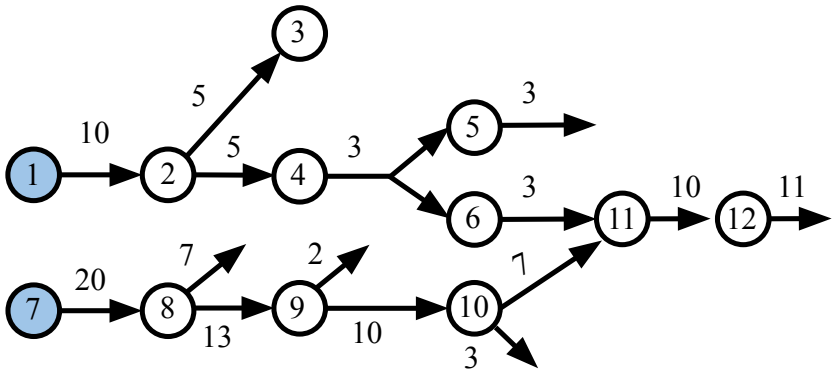


Figure 2: The transaction graph for Problem 3.

1. Honest party P has adopted a chain containing genesis (which has no transactions) and blocks B_1 (containing transaction 1 only) and B_2 (containing transaction 7 only) and is receiving the transactions from the network in this order: 2, 8, 9, 3, 4, 6, 5, 10, 11, 12. No other blocks beyond those three are mined. Which transactions will the mempool of this party contain?

2. The honest party managed to find a block B containing no new coinbase transactions and confirming its mempool on top of B_2. The rest of the honest parties then mine another k sequential blocks on top of B. No other blocks are mined in the meantime. What is the ledger L^P reported by the honest party P at the end of this process?

3. How much monetary value remains unspent in the system in the view of party P?

1. The mempool consists of transactions $2, 8, 9, 3, 4, 6, 10, 11$ in that order. It contains all the transactions except for $5, 12, 1, 7$ because 5 would be a double spend after 6 has been applied, and 12 violates the Weak Law of Conservation. 1 and 7 are coinbase transactions that are already a part of mined blocks B_1 and B_2.

2. The ledger consists of the transactions in B_1, B_2, and B because at least k blocks have been mined on top of them each. As a result, the ledger will contain transactions $1, 7, 2, 8, 9, 3, 4, 6, 10, 11$.

3. The UTXO set will be
$$\{(11, 0), (10, 1), (8, 0), (9, 0)\}$$
which amounts to $10 + 3 + 7 + 2 = 22$ of unspent output.

Problem 4

Consider the timeline of successful queries of Figure 3. The network delay is $\Delta = 1$, and we have $n = 3$ and $t = 0$.

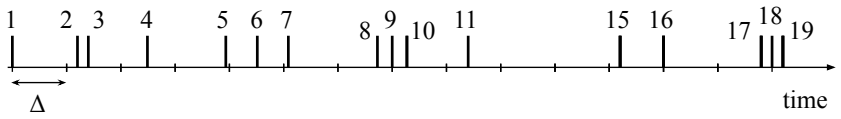


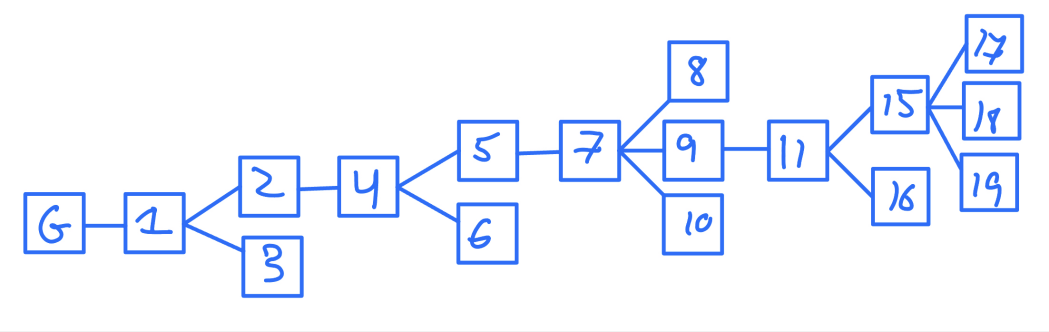
Figure 3: The timeline for Problem 4.

1. Indicate which among these successful queries are convergence opportunities.

2. Draw a blocktree that could have resulted from this timeline. For each block in the chain, indicate the successful query during which it was produced.

3. What is the height of the tip of the longest chain? What is the chain quality of the longest chain?

1. Queries 1, 4, and 11 are convergence opportunities because they are Δ-separated from all other successful queries.

2. 

A wide variety of blocktrees are possible here (e.g., one long chain with all the blocks also works). The key idea is that honest parties always build on top of a longest chain, so two Δ-separated blocks might force one to built on top of the other. For example, block 2 is more than Δ away from block 1 which is the unique chaintip, forcing block 2 to be built on top of block 1.

3. We have multiple longest chains in the blocktree above all of which have height 9. All blocks are honest, so the chain quality is 1.

Problem 5

I was using the AI program ChatGPT to save some time while preparing the lecture notes for this course. As I was working on them, ChatGPT autocompleted my notes with the following text:

> It seems that all three properties, collision resistance, preimage resistance, and second preimage resistance, are desirable. However, it is not possible to have all three at the same time. In fact, the following theorem shows that it is impossible to have collision resistance and second preimage resistance at the same time.
>
> **Theorem 1** (Krawczyk's Theorem)**.** *Let $H : \{0,1\}^* \longrightarrow \{0,1\}^\kappa$ be a hash function. Then, H is collision resistant if and only if it is not second preimage resistant.*

Prove or disprove the above theorem. You may use ChatGPT and all the theorems we have proven in class. Good luck!

Recall from class that collision resistance implies second-preimage resistance. Let H be a collision-resistant hash function. Then, H is also second-preimage resistant, disproving the theorem. Alternatively, one can construct a hash function that is neither collision resistant nor second preimage resistant. One example of such a hash function would be

$$H_\kappa(x) = 0^\kappa.$$

Algorithm 2 The second-preimage-finding game for a hash function H.

function $\mathsf{2PRE}_{\mathcal{A},H}(\kappa)$:
    $x_1 \xleftarrow{\$} \{0,1\}^{2\kappa+1}$
    $x_2 \leftarrow \mathcal{A}(x_1)$
    return $x_1 \neq x_2 \wedge H_\kappa(x_1) = H_\kappa(x_2)$
end function

Reference

Some helpful definitions are provided below. For the full definitions, consult the lecture notes.

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

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

The game is defined in Algorithm 1.

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

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

Definition (2nd Preimage Resistance)**.** A hash function $H : \{0,1\}^* \longrightarrow \{0,1\}^\kappa$ is *2nd preimage resistant* if for all PPT adversaries $\mathcal{A}$,

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

The game is defined in Algorithm 2.

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

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

Definition (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 (Common Prefix)**.** The Common Prefix virtue, parametrized by $k \in \mathbb{N}$, is satisfied if for all honest parties P_1, P_2 and for all times $r_1 \leq r_2$, it holds that $\mathcal{C}^{P_1}_{r_1}[:-k] \preceq \mathcal{C}^{P_2}_{r_2}$.

Definition (Chain Quality)**.** The Chain Quality of a chain chunk $\mathcal{C}$ is defined as the ratio of the honestly produced blocks divided by the total blocks within that chain chunk.

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.