Quantstamp Security Assessment Certificate

May 27th 2022 — Quantstamp Verified

RageTrade/Core

This audit report was prepared by Quantstamp, the leader in blockchain security.

Executive Summary

Type	
Auditors	Roman Rohleder, Research Engineer Poming Lee, Senior Research Engineer Joseph Xu, Technical R&D Advisor
Timeline	2022-04-25 through 2022-05-27
EVM	Arbitrum VM
Languages	Solidity
Methods	Architecture Review, Unit Testing, Functional Testing, Computer-Aided Verification, Manual Review
Specification	Public documentation
Documentation Quality	Medium
Test Quality	Medium

Source Code

Repository	Commit
core	ea881f6
core	7164563

Total Issues	**15** (15 Resolved)
High Risk Issues	**2** (2 Resolved)
Medium Risk Issues	**2** (2 Resolved)
Low Risk Issues	**7** (7 Resolved)
Informational Risk Issues	**2** (2 Resolved)
Undetermined Risk Issues	**2** (2 Resolved)

High Risk	The issue puts a large number of users' sensitive information at risk, or is reasonably likely to lead to catastrophic impact for client's reputation or serious financial implications for client and users.
Medium Risk	The issue puts a subset of users' sensitive information at risk, would be detrimental for the client's reputation if exploited, or is reasonably likely to lead to moderate financial impact.
Low Risk	The risk is relatively small and could not be exploited on a recurring basis, or is a risk that the client has indicated is low-impact in view of the client's business circumstances.
Informational	The issue does not post an immediate risk, but is relevant to security best practices or Defence in Depth.
Undetermined	The impact of the issue is uncertain.
Unresolved	Acknowledged the existence of the risk, and decided to accept it without engaging in special efforts to control it.
Acknowledged	The issue remains in the code but is a result of an intentional business or design decision. As such, it is supposed to be addressed outside the programmatic means, such as: 1) comments, documentation, README, FAQ; 2) business processes; 3) analyses showing that the issue shall have no negative consequences in practice (e.g., gas analysis, deployment settings).
Fixed	Adjusted program implementation, requirements or constraints to eliminate the risk.
Mitigated	Implemented actions to minimize the impact or likelihood of the risk.

Summary of Findings

We have audited the `core` repository (PerpSwap) of RageTrade, which implements an AMM-based perpetual futures market. During the course of this audit we have found issues spanning all severity levels from high to informational, including issues of yet undetermined severity. Notably, two high severity and two medium severity issues have been identified. We recommend addressing all the findings, before deploying the audited contracts. While covering around 80% of lines and 70% of branches, the test suite is missing some files and some critical lines completely and should be improved before deployment. Overall, the code and the test suite appear to be of high quality, however, the project is very complex due to the large number of non-trivial calculations and logic.

Update: After the re-audit the rage trade team addressed and fixed or mitigated all of the findings. In particular the high severity issues and the two medium severity issues were fixed. All documentation issues and best practices have been fixed and the test suite has been improved. In addition to the findings listed in this report, the rage trade team independently found and fixed 6 issues during the time of this audit, which have been reviewed and acknowledged by the Quantstamp auditors:

- RT-1: Price Caching (https://github.com/RageTrade/core/pull/145)
- RT-2: Liquidity Change Bug (https://github.com/RageTrade/core/pull/155)
- RT-3: Uniswap Force Push Fix (https://github.com/RageTrade/core/pull/137)
- RT-4: Fix circular imports (https://github.com/RageTrade/core/pull/156)
- RT-5: Event Updates (https://github.com/RageTrade/core/pull/144)
- RT-6: Liqudity Change (Close Token Position) (https://github.com/RageTrade/core/pull/148)

ID	Description	Severity	Status
QSP-1	Funds Drainable to `teamMultisig()`	High	Fixed
QSP-2	No Guarantee that the Global Funding State for All Pools Will Be Updated on `pause()`/`unpause()`	High	Fixed
QSP-3	`fixFee` Not Reflected in Liquidation Related Operations	Medium	Fixed
QSP-4	Multicall Cannot Be Used by Keepers to Liquidate	Medium	Fixed
QSP-5	Privileged Roles and Ownership	Low	Mitigated
QSP-6	Use of Unsafe Casts	Low	Fixed
QSP-7	Inverted Sign in the Funding Rate Calculation	Low	Fixed
QSP-8	Token 1 Is Cast to the `IVToken` Interface	Low	Fixed
QSP-9	Governance and Team Multisig Address Transfer Is Finalized Without Requiring Confirmation from the New Governance/Multisig Address	Low	Fixed
QSP-10	`VToken.setVPoolWrapper()` Can Be Called by Anyone	Low	Fixed
QSP-11	Lack of Input Validations	Low	Fixed
QSP-12	Unlocked Pragma	Informational	Mitigated
QSP-13	Settlement Token Oracle Always Returning `1`	Informational	Fixed
QSP-14	Removed `LiquidityPosition` Is Still Calculated in `maxNetPosition`	Undetermined	Fixed
QSP-15	Removed `LiquidityPosition` Is Still Calculated in `longSideRisk`	Undetermined	Fixed

Quantstamp Audit Breakdown

Quantstamp's objective was to evaluate the repository for security-related issues, code quality, and adherence to specification and best practices.

Possible issues we looked for included (but are not limited to):

- Transaction-ordering dependence
- Timestamp dependence
- Mishandled exceptions and call stack limits
- Unsafe external calls
- Integer overflow / underflow
- Number rounding errors
- Reentrancy and cross-function vulnerabilities
- Denial of service / logical oversights
- Access control
- Centralization of power
- Business logic contradicting the specification
- Code clones, functionality duplication
- Gas usage
- Arbitrary token minting

Methodology

The Quantstamp auditing process follows a routine series of steps:

1. Code review that includes the following
   i. Review of the specifications, sources, and instructions provided to Quantstamp to make sure we understand the size, scope, and functionality of the smart contract.
   ii. Manual review of code, which is the process of reading source code line-by-line in an attempt to identify potential vulnerabilities.
   iii. Comparison to specification, which is the process of checking whether the code does what the specifications, sources, and instructions provided to Quantstamp describe.
2. Testing and automated analysis that includes the following:
   i. Test coverage analysis, which is the process of determining whether the test cases are actually covering the code and how much code is exercised when we run those test cases.
   ii. Symbolic execution, which is analyzing a program to determine what inputs cause each part of a program to execute.
3. Best practices review, which is a review of the smart contracts to improve efficiency, effectiveness, clarify, maintainability, security, and control based on the established industry and academic practices, recommendations, and research.
4. Specific, itemized, and actionable recommendations to help you take steps to secure your smart contracts.

Findings

QSP-1 Funds Drainable to `teamMultisig()`

Severity: *High Risk*

Status: Fixed

File(s) affected: `contracts/protocol/clearinghouse/ClearingHouse.sol`

Description: Function `ClearingHouse.withdrawProtocolFee()` accepts an arbitrary list of addresses, on which it calls `collectAccruedProtocolFee()` to retrieve amounts to be withdrawn and then subsequently transfers the accumulated amounts of `protocol.settlementToken` from that contract to `teamMultisig()`. As this function is publicly callable and the provided addresses are not checked in any way (i.e. whitelisted) an attacker could deploy a smart contract implementing a function `collectAccruedProtocolFee()` that returns an arbitrary amount and subsequently drain the `ClearingHouse.sol` contract of funds, transferring them to `teamMultisig()`.

Recommendation: Consider constraining the access to function `withdrawProtocolFee()`, by i.e. adding modifier `onlyGovernanceOrTeamMultisig()` and check the provided address list `wrapperAddresses` against a list of previously whitelisted addresses.

Update: Fixed in PR https://github.com/RageTrade/core/pull/160, by deriving the amount of fees to collect from internal protocol pool addresses, instead of a user supplied list.

QSP-2 No Guarantee that the Global Funding State for All Pools Will Be Updated on `pause()`/`unpause()`

Severity: *High Risk*

Status: Fixed

File(s) affected: `contracts/protocols/clearinghouse/ClearingHouse.sol`

Description: The protocol can be paused/unpaused by the governance or the team multisig by calling `ClearingHouse.pause()` and `ClearingHouse.unpause()` respectively. During the pause/unpause operation, the global funding state must be updated for all the pools to prevent funding payments from accumulating while the protocol is paused. However, the list of pools whose global funding state will be updated on pause/unpause must be supplied as an argument `uint32[] calldata allPoolIds` to each of these functions. This introduces a manual error risk in which some pools' global funding state is not updated on the pause/unpause operation.

Recommendation: Maintain a list of all pools in either `Protocol.sol` or `RageTradeFactory.sol` and reference this list to pause/unpause rather than supplying the pool IDs through the function argument.

Update: Fixed in PR https://github.com/RageTrade/core/pull/143, by iterating based on the full list of existing pools by using `_forEachPoolOnPause` and `_forEachPoolOnUnpause`.

QSP-3 `fixFee` Not Reflected in Liquidation Related Operations

Severity: *Medium Risk*

Status: Fixed

File(s) affected: `contracts/protocols/clearinghouse/ClearingHouse.sol`

Description: Liquidation related mechanisms in the protocol can have an additional fee called `fixFee` that are paid to keepers, in addition to the keeper fee. However, the internal functions that handle the liquidation mechanisms (`ClearingHouse._removeLimitOrder()`, `ClearingHouse._liquidateLiquidityPositions()`, and `ClearingHouse._liquidateTokenPosition()`) all pass `0` as the `fixFee` argument, instead of referencing the `_getFixFee()` function (this function currently returns zero but it can be overriden in specific chain implementations).

Recommendation: Call the `_getFixFee()` function in liquidation related operations to incorporate the `fixFee`.

Update: Fixed in PR https://github.com/RageTrade/core/pull/142, by removing the fixFee parameter and functionality altogether.

QSP-4 Multicall Cannot Be Used by Keepers to Liquidate

Severity: *Medium Risk*

Status: Fixed

File(s) affected: `contracts/protocols/clearinghouse/ClearingHouse.sol`

Description: When users call liquidation related functions in `ClearingHouse.multicallWithSingleMarginCheck()`, the `accountId` variable is always passed as the liquidation target. However, `ClearingHouse.multicallWithSingleMarginCheck()` always enforces the check `ClearingHouse._getAccountAndCheckOwner()` to ensure that `msg.sender` and the variable `accountId` are the same, meaning that the keeper cannot liquidate another account.

Recommendation: Decode the `calldata` array to obtain the liquidation target `accountId` on the liquidation related functions in multicall.

Update: Fixed in PR https://github.com/RageTrade/core/pull/147, by having liquidator to pass in the target account to liquidate.

QSP-5 Privileged Roles and Ownership

Severity: *Low Risk*

Status: Mitigated

File(s) affected: `contracts/protocol/clearinghouse/ClearingHouse.sol`

Description: Smart contracts will often have `owner` variables to designate the person with special privileges to make modifications to the smart contract.
The `ClearingHouse.sol` contract contains the following privileged roles:

- `_governance` and `_teamMultisig`, as initialized during the constructor:
  . Renounce the role(s) and thereby disable all followingly listed actions, by calling `transferGovernance()` and `transferTeamMultisig()` with a known uncontrolled non-zero address.
  . Assign an arbitrary address as new `_governance` or `_teamMultisig`, by calling `transferGovernance()` and `transferTeamMultisig()`, respectively.
  . Modify collateral settings, by calling `updateCollateralSettings()`.
  . Modify pool settings, by calling `updatePoolSettings()`.
  . Modify protocol settings, by calling `updateProtocolSettings()`.
  . Putting the contract in and out of the paused state, by calling `pause()`/`unpause()`.
- `rageTradeFactoryAddress`, as initialized during `initialize()`:
  . Register new pools, by calling `registerPool()`.

Recommendation: This centralization of power needs to be made clear to the users, especially depending on the level of privilege the contract allows to the owner.

Update: Mitigated in PR https://github.com/RageTrade/core/pull/165, by restricting further privileges to governance only, but giving team multisig the ability to override the funding rate setting, additional to governance.

QSP-6 Use of Unsafe Casts

Severity: *Low Risk*

Status: Fixed

File(s) affected: `contracts/libraries/Protocol.sol`

Related Issue(s): SWC-101

Description: In L118 of `Protocol.sol` primitive cast operations (`int256(...)`) are used, which are prone for over-/underflows.

Recommendation: Replace the primitive unsafe cast operations with i.e. OpenZeppelin's SafeCast library alternatives.

Update: Fixed in PR https://github.com/RageTrade/core/pull/149, by manually handling the signedess cases via if-else syntax.

QSP-7 Inverted Sign in the Funding Rate Calculation

Severity: *Low Risk*

Status: Fixed

File(s) affected: `contracts/libraries/FundingPayment.sol`, `contracts/libraries/LiquidityPosition.sol`, `contracts/libraries/VTokenPosition.sol`

Description: The funding rate calculation in `FundingPayment.getFundingRate()` has an inverted sign because the funding rate is computed with the numerator `int256(realPriceX128) - int256(virtualPriceX128)`, as opposed to the usual formula `int256(virtualPriceX128) - int256(realPriceX128)`. While this inverted sign is

corrected by further inverting in `LiquiditionPosition.unrealizedFundingPayment()` and `VTokenPosition.unrealizeFundingPaymment()`, this could lead to other errors in funding rate calculation in future development.

Recommendation: Modify the funding rate calculation to follow the convention.

Update: The funding rate calculation in `FundingPayment.sol` was fixed. Function documentation in `LiquidityPosition.sol` and `VTokenPosition.sol` were adjusted to match the behaviour.

QSP-8 Token 1 Is Cast to the `IVToken` Interface

Severity: *Low Risk*

Status: Fixed

File(s) affected: `contracts/protocols/wrapper/VPoolWrapper.sol`

Description: On `L323` in `VPoolWrapper.uniswapV3SwapCallback()`, `token1` is cast to `IVToken(vPool.token1())` even though `token1` in the Uniswap V3 pool is always the `VQuoteToken`. It should be `IVQuote(vPool.token1())`. While the two interfaces are currently identical, it may lead to other errors in future development.

Recommendation: Correct the code so that the token is cast to the correct interface.

Update: Fixed in PR https://github.com/RageTrade/core/pull/140, by correcting the cast from `IVToken` to `IVQuote`, as suggested and adding comments.

QSP-9 Governance and Team Multisig Address Transfer Is Finalized Without Requiring Confirmation from the New Governance/Multisig Address

Severity: *Low Risk*

Status: Fixed

File(s) affected: `contracts/utils/Governable.sol`

Description: The governance or team multisig privileges can be transferred to another address simply by calling `Governable.transferGovernance()` or `Governanble.transferTeamMultisig()` functions respectively. These two functions immediately transfer a high level of privileges to new addresses in a single transaction, which can be risky from a security perspective.

A more common pattern for governance or team multisig privilege transfers is to require the new pending addresses to issue an `acceptAdmin()` function call before finalizing the transfer. Note that this pattern is common even with the use of timelocks, such as the Compound Timelock contract.

Recommendation: Require the pending new governance/team multisig address to make a `acceptAdmin()` function call before transferring the privileges.

Update: Fixed in PR https://github.com/RageTrade/core/pull/153 and https://github.com/RageTrade/core/pull/170, by implementing a `initiate*Transfer()` and `accept*Transfer()` functions for the `_governance` and `_teamMultisig` roles, as suggested.

QSP-10 `VToken.setVPoolWrapper()` Can Be Called by Anyone

Severity: *Low Risk*

Status: Fixed

File(s) affected: `contracts/protocol/tokens/VToken.sol`

Description: The `VToken` contract has a function `VToken.setVPoolWrapper()` that is used to authorize the `VPoolWrapper.sol` contract so that only it can mint the virtual token. However, this is an external function without permission check, so there is a possibility of the function getting called by another entity after deployment. As of now, this is not an issue since `VToken.setVPoolWrapper()` is called in the same function as the `VToken` deployment (in `RageTradeFactory.initializePool()`). Nevertheless, it is a good security practice to add permission checks on important functions like this.

Recommendation: Add `onlyOwner()` modifier to `VToken.setVPoolWrapper()` so that it is consistent with `VQuote.authorize()`.

Update: Fixed in PR https://github.com/RageTrade/core/pull/136, by making `VToken` inherit from `Ownable` and adding the `onlyOwner` modifier to `setVPoolWrapper()`, as suggested.

QSP-11 Lack of Input Validations

Severity: *Low Risk*

Status: Fixed

File(s) affected: `contracts/protocol/clearinghouse/ClearingHouse.sol`

Description: There are no checks on whether the values being supplied on collateral, pool, or protocol parameter changes are reasonable or not (`ClearingHouse.updateCollateralSettings()`, `ClearingHouse.updatePoolSettings()`, `ClearingHouse.updateProtocolSettings()`). Given the importance of these parameters, it is important to be able to reject unreasonable values. For example, parameter `_removeLimitOrderFee` in `ClearingHouse.updateProtocolSettings()` is unconstrained, leaving users vulnerable to potentially unexpected very high fees.

Recommendation: Define a reasonable range of parameter values for each of these settings and implement checks so that mis-specified values can be rejected.

Update: Fixed in PR https://github.com/RageTrade/core/pull/152 and https://github.com/RageTrade/core/pull/166, by adding parameter sanity checks, as suggested.

QSP-12 Unlocked Pragma

Severity: *Informational*

Status: Mitigated

File(s) affected: `all`

Related Issue(s): SWC-103

Description: Every Solidity file specifies in the header a version number of the format `pragma solidity (^)0.8.*`. The caret (^) before the version number implies an unlocked pragma, meaning that the compiler will use the specified version *and above*, hence the term "unlocked". Further, while for the most part, the pragma is ^0.8.9, the following contracts use a different pragma:

- In `./extloads/ClearingHouseExtsload.sol`, it is `^0.8.0`.
- In `./interfaces/IExtsload.sol`, it is `>=0.5.0`.
- In `./libraries/Account.sol`, it is `^0.8.10`.
- In `./libraries/Bytes32.sol`, it is `^0.8.0`.
- In `./libraries/Protocol.sol`, it is `^0.8.0`.
- In `./libraries/SafeCast.sol`, it is `^0.8.0`.
- In `./libraries/SignedFullMath.sol`, it is `>=0.8.0`.
- In `./utils/Multicall.sol`, it is `>=0.7.6`.
- In `./utils/TimelockControllerWithMinDelayOverride.sol`, it is `^0.8.0`.

Recommendation: For consistency and to prevent unexpected behavior in the future, we recommend to remove the caret to lock the file onto a specific Solidity version.

Update: Mitigated in PR https://github.com/RageTrade/core/pull/164, by locking the main contracts (`./protocol/*`) to `=0.8.14`.

QSP-13 Settlement Token Oracle Always Returning `1`

Severity: *Informational*

Status: Fixed

File(s) affected: `contracts/oracles/SettlementTokenOracle.sol`

Description: The settlement token oracle function `SettlementTokenOracle.getTwapPriceX128()` always returns `1 (X128)`, as a healthy stablecoin value is assumed.

Recommendation: Add publicly facing documentation clearly stating this assumption and dependency and consider having a monitoring system in place, putting the contracts into paused state in case of depegging.

Update: Corresponding documentation was added at https://docs.rage.trade/oracles.

QSP-14 Removed `LiquidityPosition` Is Still Calculated in `maxNetPosition`

Severity: *Undetermined*

Status: Fixed

File(s) affected: `contracts/libraries/LiquidityPositionSet.sol`

Description: `LiquidityPositionSet.maxNetPosition` does not stop even when encountering a removed/uninitialized `LiquidityPosition`.

Recommendation: Should `break` whenever running out of active `LiquidityPosition`s.

Update: Fixed in PR https://github.com/RageTrade/core/pull/135, by adding a `break`, when encountering a removed/uninitialized position, as suggested.

QSP-15 Removed `LiquidityPosition` Is Still Calculated in `longSideRisk`

Severity: *Undetermined*

Status: Fixed

File(s) affected: `contracts/libraries/LiquidityPositionSet.sol`

Description: `LiquidityPositionSet.longSideRisk` does not stop even when encountering a removed/uninitialized `LiquidityPosition`.

Recommendation: Should `break` whenever running out of active `LiquidityPosition`s.

Update: Fixed in PR https://github.com/RageTrade/core/pull/141, by adding a `break`, when encountering a removed/uninitialized position, as suggested.

Code Documentation

(Update: All fixed)

1. Missing or incorrect NatSpec parameter comments:
    1. Missing parameters `protocol` and `checkMargin` in `Account.updateMargin()` and `Account._updateMargin()`.
    2. Missing parameter `checkMargin` in `Account.updateProfit()` and `Account._updateProfit()`.
    3. Missing return value `balanceAdjustments` in `Account._updateVQuoteBalance()`.
    4. Missing return value `value` in `VTokenPosition.marketValue()`.
    5. Missing return value in `VTokenPosition.unrealizedFundingPayment()`.
    6. NatSpec parameter `sqrtPriceX96` in `PriceMath.toPriceX128()` wrongfully states `input price in Q128 format`, however the format is `Q96`, not `Q128`.
    7. Missing parameters `checkMargin` and return values `vTokenAmountOut` and `vQuoteAmountOut` in `Account.swapToken()`.
    8. Missing parameters `fixFee` and return values `keeperFee`, `insuranceFundFee` and `accountMarketValue` in `Account.liquidateLiquidityPositions()`.
    9. Missing parameters `fixFee` and return values `keeperFee` and `insuranceFundFee` in `Account.liquidateTokenPosition()`.
    10. Missing parameter `fees` in `SwapMath.includeFees()`.
    11. Missing parameter `accountId` in `VTokenPositionSet.swapToken()`.
    12. Missing parameter `accountId` in `VTokenPositionSet.liquidateLiquidityPositions()`.
    13. Missing parameter `accountId` in `VTokenPositionSet.liquidityChange()`.

14. Missing parameter `accountId` in `VTokenPositionSet.removeLimitOrder()`.

2. The following typographical error has been noted:
    1. `contracts/protocol/tokens/VQuoteDeployer.sol`: `rsettlementTokenecimals` should be `rsettlementToken[d]ecimals`.

3. On `L80` in `contracts/utils/TimelockControllerWithMinDelayOverride.sol` the code comment and the implementation on `L81` are different. Please correct the code comment.

Adherence to Best Practices

1. There are unused contracts `contracts/utils/ProxyAdmin.sol`, `contracts/utils/TransparentUpgradeableProxy.sol` that simply imports the OpenZeppelin library. These contracts should be removed from the repository.
2. The following dead code has been noted:
    1. `L47` in `contracts/protocol/RageTradeFactory.sol`.

3. The following duplicate line of code has been noted:
    1. `L10` in `contracts/protocol/clearinghouse/ClearingHouseStorage.sol`.

4. The naming in `L9-10` of `contracts/utils/TimelockController` is too similar.
5. Consider using `external` instead of `public` for function `updatePoolSettings()` in `contracts/protocol/clearinghouse/ClearingHouse.sol` to reduce gas cost since this function will not be called by any other functions within the contract.

Test Results

Test Suite Results

Of the 709 tests, 704 passed, while 5 were stuck in pending state.

Update: For the re-audit additional tests have been added. Of the 765 tests, 759 were passing and 6 were stuck in pending state.

```
Clearing House Library
  #Init Params
     ✓ Set Params (57ms)
  #StealFunds
     ✓ Steal Funds (585ms)
  #AccountCreation
     ✓ Create Account - 1
     ✓ Create Account - 1
  #InitializeToken
     ✓ vToken Intialized
     ✓ Other Address Not Intialized
  #TokenSupport
     ✓ Add Token Position Support - Fail - Unauthorized
     ✓ Add Token Position Support - Pass (67ms)
     ✓ Add Token Deposit Support - Fail - Unauthorized
     ✓ AddVQuote Deposit Support  - Pass (45ms)
  #Pause Check
     ✓ Pause (196ms)
     ✓ Create Account
     ✓ Deposit
     ✓ Withdraw
     ✓ Profit
     ✓ Token Position
     ✓ Range Position
     ✓ Token Liquidation
     ✓ Range Liquidation
     ✓ Remove Limit Order
     ✓ UnPause (172ms)
  #Deposit
     ✓ Fail - Access Denied
     ✓ Fail - Uninitialized Token
     ✓ Fail - Unsupported Token
     ✓ Pass (96ms)
  #Withdraw
     ✓ Fail - Access Denied
     ✓ Fail - Uninitialized Token
     ✓ Pass (55ms)
     ✓ Pass - Withdrawal after removal of token support (387ms)
  #Profit
     ✓ Fail - Access Denied
     ✓ Pass - Cover Loss (62ms)
     ✓ Pass - Remove Profit (48ms)
  #InitLiquidity
     ✓ #InitLiquidity (645ms)
  #SwapTokenAmout - Without Limit
     ✓ Fail - Access Denied
     ✓ Fail - Uninitialized Token
     ✓ Fail - Unsupported Token (76ms)
     ✓ Fail - Low Notional Value (68ms)
     ✓ Pass (309ms)
  #SwapTokenNotional - Without Limit
     ✓ Fail - Access Denied
     ✓ Fail - Uninitialized Token
     ✓ Fail - Unsupported Token
     ✓ Fail - Low Notional Value (319ms)
     ✓ Pass (274ms)
  #LiquidityChange - Without Limit
     ✓ Fail - Access Denied
     ✓ Fail - Uninitialized Token
     ✓ Fail - Unsupported Token
     ✓ Fail - Low Notional Value
     ✓ Pass (330ms)
  Withdraw Protocol Fee
     ✓ Valid Pool Fee Withdrawal (40ms)
  Multicall
     ✓ multicallWithSingleMarginCheck (363ms)

  Clearing House Extsload
saved "VQuote": 0xFfefE40b54a0Eb3B35ee345ed356670B5AE50F5a
saved "ClearingHouse": 0x1D6fD8668896dC2FA849F5ef7A510A8C9d8b563C
saved "ProxyAdmin": 0x9230C445Ba467b69C09918E35c233B065F289A39
saved "InsuranceFund": 0x13328E4E2F2819cd49a611930364F1513b658eE7
saved "SettlementTokenOracle": 0xFD471836031dc5108809D173A067e8486B9047A3
No deployment found for: ETH-vToken
saved "ETH-vToken": 0x07daDaae1d4bC632780B06F248F4A0D808e6D55e
saved "ETH-vPool": 0x78196b033298c67C8bfdAAC133aADB177E061A92
saved "ETH-vPoolWrapper": 0xce8E04cadCBF3034b99643415A70751F181B586c
  protocol
     ✓ vPool
     ✓ settings
     ✓ vPool and twapDuration
     ✓ isPoolIdAvailable
     ✓ getPoolInfo
     ✓ getProtocolInfo
     ✓ getCollateralInfo
  account
     ✓ getAccountInfo
     ✓ getAccountInfo
     ✓ getAccountCollateralInfo
     ✓ getAccountCollateralBalance
     ✓ getAccountTokenPositionInfo
     ✓ getAccountPositionInfo
  #slots
     ✓ protocol slot
     ✓ account slot
```

```
Clearing House Scenario 1 (Base swaps and liquidity changes)
  #Init Params
     ✓ Set Params (49ms)
  #Initialize
     ✓ Steal Funds (137ms)
     ✓ Create Account - 1
     ✓ Create Account - 2
     ✓ Create Account - 3 (56ms)
     ✓ Tokens Intialized
     ✓ Add Token Position Support - Pass (75ms)
     ✓ AddVQuote Deposit Support  - Pass
  #Scenario
     ✓ Timestamp And Oracle Update - 0 (47ms)
     ✓ Acct[0] Initial Collateral Deposit = 100K USDC (64ms)
     ✓ Acct[0] Adds Liq b/w ticks (-200820 to -199360) @ tickCurrent = -199590 (478ms)
     ✓ Timestamp and Oracle Update - 600 (55ms)
     ✓ Acct[2] Initial Collateral Deposit = 100K USDC (69ms)
     ✓ Acct[2] Short ETH : Price Changes (StartTick = -199590, EndTick = -199700) (434ms)
     ✓ Acct[1] Initial Collateral Deposit = 100K USDC (49ms)
     ✓ Timestamp and Oracle Update - 1200 (39ms)
     ✓ Acct[1] Adds Liq b/w ticks (-200310 to -199820) @ tickCurrent = -199700 (381ms)
     ✓ Timestamp and Oracle Update - 1900 (46ms)
     ✓ Acct[2] Short ETH : Price Changes (StartTick = -199700, EndTick = -199820) (320ms)
     ✓ Timestamp and Oracle Update - 2600
     ✓ Acct[2] Short ETH : Price Changes (StartTick = -199820, EndTick = -200050 (363ms)
     ✓ Timestamp and Oracle Update - 3300
     ✓ Acct[2] Long  ETH : Price Changes (StartTick = -200050, EndTick = -199820 (355ms)
     ✓ Timestamp and Oracle Update - 4100
     ✓ Acct[2] Long  ETH : Price Changes (StartTick = -199820, EndTick = -199540 (345ms)
     ✓ Timestamp and Oracle Update - 4500 (38ms)
     ✓ Acct[2] Short ETH : Price Changes (StartTick = -199540, EndTick = -199820 (343ms)
     ✓ Timestamp and Oracle Update - 4600 (39ms)
     ✓ Acct[2] Short ETH : Price Changes (StartTick = -199820, EndTick = -200050 (401ms)
     ✓ Timestamp and Oracle Update - 5300
     ✓ Acct[2] Short ETH : Price Changes (StartTick = -200050, EndTick = -200310 (309ms)
     ✓ Timestamp and Oracle Update - 5800
     ✓ Acct[1] Removes Liq b/w ticks (-200310 to -199820) @ tickCurrent = -200310 (304ms)
     ✓ Timestamp and Oracle Update - 6200
     ✓ Acct[2] Short ETH : Price Changes (StartTick = -200310, EndTick = -200460 (308ms)
     ✓ Timestamp and Oracle Update - 6300 (52ms)
     ✓ Acct[2] Short ETH : Price Changes (StartTick = -200460, EndTick = -200610 (320ms)
     ✓ Timestamp and Oracle Update - 7200
     ✓ Acct[2] Short ETH : Price Changes (StartTick = -200610, EndTick = -200750 (293ms)
     ✓ Timestamp and Oracle Update - 7600 (41ms)
     ✓ Acct[2] Short ETH : Price Changes (StartTick = -200750, EndTick = -200800 (280ms)

Clearing House Scenario 2 (Liquidation | Account Position | Slippage Bound Full Liquidations Only)
  #Init Params
     ✓ Set Params
  #Initialize
     ✓ Steal Funds (210ms)
     ✓ Create Account - 1
     ✓ Create Account - 2
     ✓ Create Account - 3
     ✓ Create Account - Keeper (47ms)
     ✓ Tokens Intialized
     ✓ Add Token 1 Position Support - Pass (73ms)
     ✓ Add Token 2 Position Support - Pass (53ms)
     ✓ AddVQuote Deposit Support  - Pass
  #Collateral Deposit
     ✓ Acct[0] Initial Collateral Deposit = 2M USDC (62ms)
     ✓ Acct[1] Initial Collateral Deposit = 100K USDC (59ms)
     ✓ Acct[2] Initial Collateral Deposit = 10m USDC (57ms)
     ✓ Keeper Initial Collateral Deposit = 1m USDC (62ms)
  #Scenario Liquidation
     ✓ Timestamp And Oracle Update - 0 (63ms)
     ✓ Acct[0] Adds Liq to BTC Pool b/w ticks (60000 to 68000) @ tickCurrent = 64197 (597ms)
     ✓ Timestamp And Oracle Update - 100 (136ms)
     ✓ Acct[0] Adds Liq to ETH Pool b/w ticks (-190000 to -196000) @ tickCurrent = -194365 (537ms)
     ✓ Timestamp and Oracle Update - 600 (45ms)
     ✓ Acct[1] Short BTC : Price Changes (StartTick = 64197, EndTick = 64000) (308ms)
     ✓ Timestamp and Oracle Update - 1000 (65ms)
     ✓ Acct[1] Adds Liq to BTC Pool b/w ticks (63000 to 64400) @ tickCurrent = 64000 (324ms)
     ✓ Timestamp and Oracle Update - 1500 (60ms)
     ✓ Acct[1] Short ETH : Price Changes (StartTick = -194365, EndTick = -194430) (535ms)
     ✓ Timestamp and Oracle Update - 2000 (63ms)
     ✓ Acct[1] Adds Liq to ETH Pool b/w ticks (-195660 to -193370) @ tickCurrent = -194430 (499ms)
     ✓ Timestamp and Oracle Update - 2500 (65ms)
     ✓ Acct[2] Long BTC : Price Changes (StartTick = 64000, EndTick = 64400) (362ms)
     ✓ Timestamp and Oracle Update - 2600 (60ms)
     ✓ Acct[2] Long ETH : Price Changes (StartTick = -194430, EndTick = -193370) (521ms)
     ✓ Timestamp and Oracle Update - 3000 (62ms)
     ✓ Acct[2] Long BTC : Price Changes (StartTick = 64400, EndTick = 66000) (446ms)
     ✓ Timestamp and Oracle Update - 3500 (61ms)
     ✓ Acct[1] Underwater : Liquidate Ranges @ current tickBTC = 66000, current tickETH = -193370 (619ms)
     ✓ Timestamp and Oracle Update - 4000 (61ms)
     ✓ Acct[1] Underwater : Liquidate ETH Token Positions @ current tickETH = -193370 (660ms)
     ✓ Timestamp and Oracle Update - 4500 (59ms)
     ✓ Acct[1] Underwater : Liquidate BTC Token Positions @ current tickBTC = 66000 (625ms)

Clearing House Scenario 3 (Liquidation | Account Negative | Slippage Bound & Unbounded Full Liquidations)
  #Init Params
     ✓ Set Params (79ms)
  #Initialize
     ✓ Steal Funds (233ms)
     ✓ Create Account - 1
     ✓ Create Account - 2
     ✓ Create Account - 3
     ✓ Create Account - Keeper
     ✓ Tokens Intialized
     ✓ Add Token 1 Position Support - Pass (44ms)
     ✓ Add Token 2 Position Support - Pass (39ms)
     ✓ AddVQuote Deposit Support  - Pass
  #Scenario Underwater Liquidation
     ✓ Acct[0] Initial Collateral Deposit = 2M USDC (58ms)
     ✓ Acct[1] Initial Collateral Deposit = 100K USDC (1201ms)
     ✓ Acct[2] Initial Collateral Deposit = 10m USDC (76ms)
     ✓ Keeper Initial Collateral Deposit = 1m USDC (68ms)
     ✓ Timestamp And Oracle Update - 0 (71ms)
     ✓ Acct[0] Adds Liq to BTC Pool b/w ticks (60000 to 68000) @ tickCurrent = 64197 (855ms)
     ✓ Timestamp And Oracle Update - 100 (61ms)
     ✓ Acct[0] Adds Liq to ETH Pool b/w ticks (-190000 to -196000) @ tickCurrent = -194365 (524ms)
     ✓ Timestamp and Oracle Update - 600 (51ms)
     ✓ Acct[1] Short BTC : Price Changes (StartTick = 64197, EndTick = 64000) (335ms)
     ✓ Timestamp and Oracle Update - 1000 (54ms)
     ✓ Acct[1] Adds Liq to BTC Pool b/w ticks (63000 to 64400) @ tickCurrent = 64000 (287ms)
     ✓ Timestamp and Oracle Update - 1500 (62ms)
     ✓ Acct[1] Short ETH : Price Changes (StartTick = -194365, EndTick = -194430) (490ms)
     ✓ Timestamp and Oracle Update - 2000 (54ms)
     ✓ Acct[1] Adds Liq to ETH Pool b/w ticks (-195660 to -193370) @ tickCurrent = -194430 (470ms)
     ✓ Timestamp and Oracle Update - 2500 (636ms)
     ✓ Acct[2] Long BTC : Price Changes (StartTick = 64000, EndTick = 64400) (416ms)
     ✓ Timestamp and Oracle Update - 2600 (72ms)
     ✓ Acct[2] Long ETH : Price Changes (StartTick = -194430, EndTick = -193370) (497ms)
     ✓ Timestamp and Oracle Update - 3000 (46ms)
     ✓ Acct[2] Long BTC : Price Changes (StartTick = 64400, EndTick = 67000) (416ms)
     ✓ Timestamp and Oracle Update - 3500 (55ms)
     ✓ Acct[1] Underwater : Liquidate Ranges @ current tickBTC = 66000, current tickETH = -193370 (590ms)
     ✓ Timestamp and Oracle Update - 4000 (52ms)
     ✓ Acct[1] Underwater : Liquidate Partial ETH Token Positions @ current tickETH = -193370 (592ms)
     ✓ Timestamp and Oracle Update - 4400 (56ms)
     ✓ Acct[1] Underwater : Liquidate Full ETH Token Positions @ current tickETH = -193073 (537ms)
     ✓ Timestamp and Oracle Update - 4500 (64ms)
     ✓ Acct[1] Underwater : Liquidate Partial BTC Token Positions @ current tickBTC = 67000 (392ms)
     ✓ Timestamp and Oracle Update - 4900 (952ms)
     ✓ Acct[1] Underwater : Liquidate Full BTC Token Positions @ current tickBTC = 67297 (383ms)

Clearing House Scenario 4 (Partial Swaps & Notional Swaps)
  #Init Params
     ✓ Set Params (76ms)
  #Initialize
     ✓ Steal Funds (734ms)
     ✓ Create Account - 1
     ✓ Create Account - 2
     ✓ Create Account - 3
     ✓ Create Account - Keeper
     ✓ Tokens Intialized
     ✓ Add Token 1 Position Support - Pass (41ms)
     ✓ Add Token 2 Position Support - Pass (45ms)
     ✓ AddVQuote Deposit Support  - Pass
  #Scneario
     ✓ Acct[0] Initial Collateral Deposit = 2M USDC (68ms)
     ✓ Acct[1] Initial Collateral Deposit = 100K USDC (67ms)
     ✓ Acct[2] Initial Collateral Deposit = 10m USDC (61ms)
     ✓ Keeper Initial Collateral Deposit = 1m USDC (66ms)
     ✓ Timestamp And Oracle Update - 0 (69ms)
     ✓ Acct[0] Adds Liq to BTC Pool b/w ticks (60000 to 68000) @ tickCurrent = 64197 (486ms)
     ✓ Timestamp And Oracle Update - 100 (57ms)
     ✓ Acct[0] Adds Liq to ETH Pool b/w ticks (-190000 to -196000) @ tickCurrent = -194365 (603ms)
     ✓ Timestamp and Oracle Update - 600 (43ms)
     ✓ Acct[1] Short BTC : Price Changes (StartTick = 64197, EndTick = 64000) (322ms)
     ✓ Timestamp and Oracle Update - 1000 (57ms)
     ✓ Acct[1] Adds Liq to BTC Pool b/w ticks (63000 to 64400) @ tickCurrent = 64000 (304ms)
     ✓ Timestamp and Oracle Update - 1500 (54ms)
     ✓ Acct[1] Short ETH : Price Changes (StartTick = -194365, EndTick = -194430) (514ms)
     ✓ Timestamp and Oracle Update - 2000 (54ms)
```

```
      ✓ Acct[1] Adds Liq to ETH Pool b/w ticks (-195660 to -193370) @ tickCurrent = -194430 (479ms)
      ✓ Acct[1] Adds Liq to ETH Pool b/w ticks (-195660 to -193370) @ tickCurrent = -194430 (FAIL - Slippage Beyond Tolerance) (62ms)
      ✓ Timestamp and Oracle Update - 2500 (905ms)
      ✓ Acct[2] Long BTC : Price Changes (StartTick = 64000, EndTick = 64400) (502ms)
      ✓ Timestamp and Oracle Update - 2600 (77ms)
      ✓ Acct[2] Long ETH : Price Changes (StartTick = -194430, EndTick = -193370) (725ms)
      ✓ Timestamp and Oracle Update - 3000 (61ms)
      ✓ Acct[2] Long BTC (Partial Swap = False): Price Changes (StartTick = 64400, EndTick = 65500) (485ms)
      ✓ Acct[2] Long BTC (Partial Swap = True): Price Changes (StartTick = 64400, EndTick = 65500) (445ms)
      ✓ Timestamp and Oracle Update - 3500 (59ms)
      ✓ Acct[2] Long BTC (isNotional = True) : Price Changes (StartTick = 65499, EndTick = 65999) (441ms)
      ✓ Timestamp and Oracle Update - 4000 (61ms)
      ✓ Acct[2] Short BTC (isNotional = True) : Price Changes (StartTick = 65999, EndTick = 65499) (436ms)

  Clearing House Scenario 5 (Liquidation | Account Position | Full Liquidation Without Slippage Bound)
    #Init Params
      ✓ Set Params (38ms)
    #Initialize
      ✓ Steal Funds (181ms)
      ✓ Create Account - 1
      ✓ Create Account - 2 (557ms)
      ✓ Create Account - 3
      ✓ Create Account - Keeper
      ✓ Tokens Intialized
      ✓ Add Token 1 Position Support - Pass (51ms)
      ✓ Add Token 2 Position Support - Pass (45ms)
      ✓ AddVQuote Deposit Support  - Pass
    #Collateral Deposit
      ✓ Acct[0] Initial Collateral Deposit = 5M USDC (70ms)
      ✓ Acct[1] Initial Collateral Deposit = 100K USDC (62ms)
      ✓ Acct[2] Initial Collateral Deposit = 10m USDC (61ms)
      ✓ Keeper Initial Collateral Deposit = 1m USDC (57ms)
    #Scenario Partial Token Liquidation
      ✓ Timestamp And Oracle Update - 0 (61ms)
      ✓ Acct[0] Adds Liq to ETH Pool b/w ticks (-200000 to -195000) @ tickCurrent = -197830 (792ms)
      ✓ Timestamp and Oracle Update - 1000 (239ms)
      ✓ Acct[1] Short ETH : Price Changes (StartTick = -197830, EndTick = -198080) (360ms)
      ✓ Timestamp and Oracle Update - 1500 (48ms)
      ✓ Acct[2] Long ETH : Price Changes (StartTick = -198080, EndTick = -196505) (349ms)
      ✓ Timestamp and Oracle Update - 2000 (42ms)
      ✓ Acct[1] Underwater : Liquidate ETH Token Positions @ current tickETH = -196505 (302ms)

  Clearing House Scenario 6
    #Init Params
      ✓ Set Params (52ms)
    #Initialize
      ✓ Steal Funds (153ms)
      ✓ Create Account - 1
      ✓ Create Account - 2
      ✓ Create Account - 3
      ✓ Tokens Intialized
      ✓ Add Token Position Support - Pass (47ms)
      ✓ AddVQuote Deposit Support  - Pass
    #Scenario 1
      ✓ Timestamp And Oracle Update - 0 (54ms)
      ✓ Acct[0] Initial Collateral Deposit = 100K USDC (67ms)
      ✓ Acct[0] Adds Liq b/w ticks (-200820 to -199360) @ tickCurrent = -199590 (460ms)
      ✓ Timestamp and Oracle Update - 600
      ✓ Acct[2] Initial Collateral Deposit = 100K USDC (47ms)
      ✓ Acct[2] Short ETH : Price Changes (StartTick = -199590, EndTick = -199700) (353ms)
      ✓ Acct[1] Initial Collateral Deposit = 1mil USDC (1146ms)
      ✓ Timestamp and Oracle Update - 1200
      ✓ Acct[1] Adds Liq b/w ticks (-200310 to -199820) @ tickCurrent = -199700 (524ms)
      ✓ Timestamp and Oracle Update - 1300 (51ms)
      ✓ Acct[1] Adds Liq b/w ticks (-200200 to -199900) @ tickCurrent = -199700 (462ms)
      ✓ Timestamp and Oracle Update - 1400 (50ms)
      ✓ Acct[1] Adds Liq b/w ticks (-200100 to -200000) @ tickCurrent = -199700 (509ms)
      ✓ Timestamp and Oracle Update - 1900
      ✓ Acct[2] Short ETH : Price Changes (StartTick = -199700, EndTick = -200050) (366ms)
      ✓ Timestamp and Oracle Update - 2500
      ✓ Acct[2] Long ETH : Price Changes (StartTick = -200050, EndTick = -199850 (359ms)

  Clearing House Scenario 7 (Liquidation | Account Positive | Account Value > .75 * MM & PositionNotional < MinNotionalLiquidatable )
    #Init Params
      ✓ Set Params (46ms)
    #Initialize
      ✓ Steal Funds (179ms)
      ✓ Create Account - 1
      ✓ Create Account - 2
      ✓ Create Account - 3
      ✓ Create Account - Keeper
      ✓ Tokens Intialized
      ✓ Add Token 1 Position Support - Pass (41ms)
      ✓ Add Token 2 Position Support - Pass (47ms)
      ✓ AddVQuote Deposit Support  - Pass
    #Collateral Deposit
      ✓ Acct[0] Initial Collateral Deposit = 5M USDC (70ms)
      ✓ Acct[1] Initial Collateral Deposit = 50 USDC (62ms)
      ✓ Acct[2] Initial Collateral Deposit = 10m USDC (63ms)
      ✓ Keeper Initial Collateral Deposit = 1m USDC (59ms)
    #Scenario Partial Token Liquidation
      ✓ Timestamp And Oracle Update - 0 (64ms)
      ✓ Acct[0] Adds Liq to ETH Pool b/w ticks (-198500 to -191200) @ tickCurrent = -198045 (491ms)
      ✓ Timestamp and Oracle Update - 1000 (61ms)
      ✓ Acct[1] Short ETH : Price Changes (StartTick = -198045, EndTick = -198080) (471ms)
      ✓ Timestamp and Oracle Update - 1500
      ✓ Acct[2] Long ETH : Price Changes (StartTick = -198080, EndTick = -191380) (379ms)
      ✓ Timestamp and Oracle Update - 2000 (602ms)
      ✓ Acct[1] Underwater : Liquidate ETH Token Positions @ current tickETH = -191380 (452ms)

  Clearing House Scenario 8 (Liquidation | Account Position | Partial Liquidation (Account Value > .75 * MM & PositionNotional > MinNotionalLiquidatable)
    #Init Params
      ✓ Set Params
    #Initialize
      ✓ Steal Funds (169ms)
      ✓ Create Account - 1
      ✓ Create Account - 2
      ✓ Create Account - 3
      ✓ Create Account - Keeper
      ✓ Tokens Intialized
      ✓ Add Token 1 Position Support - Pass
      ✓ Add Token 2 Position Support - Pass (45ms)
      ✓ AddVQuote Deposit Support  - Pass
    #Collateral Deposit
      ✓ Acct[0] Initial Collateral Deposit = 5M USDC (61ms)
      ✓ Acct[1] Initial Collateral Deposit = 100K USDC (47ms)
      ✓ Acct[2] Initial Collateral Deposit = 10m USDC (55ms)
      ✓ Keeper Initial Collateral Deposit = 1m USDC (67ms)
    #Scenario Partial Token Liquidation
      ✓ Timestamp And Oracle Update - 0 (59ms)
      ✓ Acct[0] Adds Liq to ETH Pool b/w ticks (-200000 to -195000) @ tickCurrent = -197830 (480ms)
      ✓ Timestamp and Oracle Update - 1000 (44ms)
      ✓ Acct[1] Short ETH : Price Changes (StartTick = -197830, EndTick = -198080) (414ms)
      ✓ Timestamp and Oracle Update - 1500 (51ms)
      ✓ Acct[2] Long ETH : Price Changes (StartTick = -198080, EndTick = -196850) (399ms)
      ✓ Timestamp and Oracle Update - 2000
      ✓ Acct[1] Underwater : Liquidate ETH Token Positions @ current tickETH = -196750 (369ms)

  Clearing House Scenario 9 (Settle Profit)
    #Init Params
      ✓ Set Params (38ms)
    #Initialize
      ✓ Steal Funds (168ms)
      ✓ Create Account - 1
      ✓ Create Account - 2
      ✓ Create Account - 3
      ✓ Create Account - Keeper
      ✓ Tokens Intialized
      ✓ Add Token 1 Position Support - Pass (45ms)
      ✓ Add Token 2 Position Support - Pass (39ms)
      ✓ AddVQuote Deposit Support  - Pass
    #Collateral Deposit
      ✓ Acct[0] Initial Collateral Deposit = 1M USDC (58ms)
      ✓ Acct[1] Initial Collateral Deposit = 1M USDC (51ms)
      ✓ Acct[2] Initial Collateral Deposit = 1M USDC (59ms)
      ✓ Keeper Initial Collateral Deposit = 1M USDC (55ms)
    #Scenario Liquidation
      ✓ Timestamp And Oracle Update - 0 (54ms)
      ✓ Acct[0] Adds Liq to BTC Pool b/w ticks (60000 to 68000) @ tickCurrent = 64197 (449ms)
      ✓ Timestamp and Oracle Update - 600 (3375ms)
      ✓ Acct[1] Long BTC : Price Changes (StartTick = 64197, EndTick = 65000) (544ms)
      ✓ Timestamp and Oracle Update - 1000 (58ms)
      ✓ Acct[2] Long BTC : Price Changes (StartTick = 65000, EndTick = 66000) (554ms)
      ✓ Timestamp and Oracle Update - 2000 (3321ms)
      ✓ Acct[1] Partial Close Position, Short BTC : Price Changes (StartTick = 66000, EndTick = 65500) (585ms)
      ✓ Timestamp And Oracle Update - 3000 (60ms)
      ✓ Acct[0] Remove Liq from BTC Pool b/w ticks (60000 to 68000) @ tickCurrent = 65500 (669ms)

  Market Value and Required Margin
    Base Case
      ✓ Scenario 1 - Add Range (428ms)
      ✓ Scenario 2 - Price Moves (332ms)
      ✓ Scenario 3 - Add Range Outside (409ms)
      ✓ Scenario 4 - Price Moves (389ms)
      ✓ Scenario 5 - Add Range Outside (402ms)
      ✓ Scenario 6 - Price Moves (2353ms)
```

```
  Additional Cases
    ✓ Scenario 1 - Full Range (440ms)
    ✓ Scenario 2 - Concentrated Range (Same Liquidity as full range) (2409ms)
    ✓ Scenario 3 - Concentrated Range (Same notional value of assets as full range) (2784ms)
    ✓ Scenario 4 - Short Trade Position + Long Range) (2924ms)
    ✓ Scenario 5 - Long Trade Position + Short Range) (690ms)
    ✓ Scenario 6 - Long Range + Short Range) (745ms)

Account Library Test Basic
  #Initialize
    ✓ Init
  #Margin
    ✓ Add Margin
    ✓ Remove Margin
  #Trades
    ✓ Swap Token (Token Amount) (263ms)
    ✓ Swap Token (Token Notional) (246ms)
    ✓ Liqudity Change (326ms)
  #Remove Limit Order
    Not limit order
      ✓ Remove Failure - Inside Range (No Limit) (41ms)
      ✓ Remove Failure - Below Range (No Limit)
      ✓ Remove Failure - Above Range (No Limit)
    Lower limit order
      ✓ Remove Failure - Inside Range (Lower Limit)
      ✓ Remove Failure - Above Range (Lower Limit)
      ✓ Remove Success - Below Range (Lower Limit) (63ms)
    Upper limit order
      ✓ Remove Failure - Inside Range (Upper Limit)
      ✓ Remove Failure - Below Range (Upper Limit)
      ✓ Remove Success - Above Range (Upper Limit) (55ms)
  #Liquidation
    ✓ Liquidate Liquidity Positions - Fail
    ✓ Liquidate Token Positions - Fail

Account Library Test Realistic
  #Initialize
    ✓ Init
  Account Market Value and Required Margin
    ✓ No Position
    ✓ Single Position (474ms)
  #Margin
    ✓ Add Margin (580ms)
    ✓ Remove Margin - Fail (794ms)
    ✓ Remove Margin - Pass (618ms)
  #Profit
    #Token Position Profit
      ✓ Remove Profit - Fail (No Profit | Enough Margin) (242ms)
      ✓ Remove Profit - Fail (Profit Available | Not Enough Margin) (646ms)
      ✓ Remove Profit - Pass (1929ms)
      ✓ Deposit Loss - Pass (575ms)
  #Trade - Swap Token Amount
    ✓ Successful Trade (281ms)
  #Trade - Swap Token Notional
    ✓ Successful Trade (287ms)
  Limit Order Removal
    ✓ Limit Order Removal (Upper) with Fee - No Price Change (921ms)
    ✓ Limit Order Removal (Lower) with Fee - No Price Change (912ms)
    ✓ Limit Order Removal (Lower) with Fee - Price Change (479ms)
    ✓ Limit Order Removal Fail - Inactive Range (438ms)
  #Single Range Position Liquidation
    ✓ Liquidation - Fail (Account Above Water) (543ms)
    ✓ Liquidation - Success (Account Positive) (4995ms)
    ✓ Liquidation - Success (Account Negative) (2752ms)
  #Multiple Range Position Liquidation
    ✓ Liquidation - Fail (Account Above Water) (569ms)
    ✓ Liquidation - Success (Account Positive) (789ms)
    ✓ Liquidation - Success (Account Negative) (1220ms)
  #Trade- Liquidity Change
    ✓ Successful Add (937ms)
    ✓ Successful Remove (No Net Position) (2075ms)
    ✓ Successful Remove And Close (No Net Position) (99ms)
    ✓ Successful Add (Non-Zero Net Position) (1392ms)
    ✓ Successful Remove (Non-Zero Net Position) (322ms)
    ✓ Successful Add And Close (Non-Zero Net Position) (3820ms)
    ✓ Successful Remove And Close (Non-Zero Net Position) (630ms)
  #Trade- Multiple Liquidity Add & Remove
    ✓ Test #1 (9, 35, 39) (87348ms)
    ✓ Test #2 (4, 5, 49) (57416ms)
    ✓ Test #3 (10, 41, 20) (77012ms)
    ✓ Test #4 (11, 30, 27) (74399ms)
    ✓ Test #5 (3, 8, 41) (60296ms)
    ✓ Test #6 (3, 33, 36) (64917ms)
    ✓ Test #7 (10, 12, 4) (24695ms)
    ✓ Test #8 (5, 45, 13) (62569ms)
    ✓ Test #9 (20, 29, 3) (65878ms)
    ✓ Test #10 (19, 21, 29) (77995ms)

BatchedLoop
  Start Index 0
    ✓ should work if empty array
    ✓ should work if single element in array (40ms)
    ✓ should do entire iterations when passed 0
    ✓ should do partial iterations (75ms)
    ✓ should work if bad inputs 1
    ✓ should work if bad inputs 2 (38ms)
  Start Index Non Zero
    ✓ should work for small batch size
    ✓ should work for huge batch size
    ✓ should not do anything if end before start

Bisection
  ✓ works
  ✓ reverts when target is out of bounds

ChainlinkPriceFeed Spec
  Error Handling
    ✓ Not Enough History
  Chainlink failure handling
    ✓ Aggregator getRoundData reverts
    ✓ Aggregator getRoundData reverts after 1 call
  Different Timestamps for each round
    ✓ Twap Duration = History
    ✓ Twap Duration > History
    ✓ Twap Duration < History
    ✓ Twap Duration = History
    ✓ (Now - Twap Duration) > Last Update TS
    ✓ Latest Price is Negative
    ✓ Middle Price is Negative
    ✓ Twap Duration = 0
  Same Timestamps for multiple rounds
    ✓ Twap Duration = History
    ✓ Twap Duration > History

Collateral Deposit Set Library
  #Single Token
    ✓ Add Margin
    ✓ Remove Margin
    ✓ Deposit Market Value
  #Multiple Tokens
    ✓ Add Margin (54ms)
    ✓ Deposit Market Value (Price1)
    ✓ Deposit Market Value (Price2)

Extsload
  ✓ reads single
  ✓ reads multiple

FundingPayment
  #a
    ✓ rp=101 vp=100 dt=10
    ✓ rp=99 vp=100 dt=10
    ✓ rp=1.01 vp=1 dt=10
  #update
    ✓ initial
    ✓ one long
    ✓ one short
    ✓ two longs
    ✓ two shorts

GoodAddressDeployer
  #deploy
    ✓ no constructor arg (3790ms)
    ✓ with constructor arg (1580ms)
    ✓ payable with constructor arg (1379ms)

Governable
  #deploy
    ✓ sets variables (6674ms)
  #transfer
    ✓ initiates governance transfer (154ms)
    ✓ initiates teamMultisig transfer
    ✓ accepts governance transfer (53ms)
    ✓ accepts teamMultisig transfer (41ms)
    ✓ can change pending address to something again
    ✓ can cancel transfer
```

```
insuranceFund
  Functions
    ✓ Is initilized Correctly
    ✓ Deposit 1:1 (55ms)
    ✓ Withdraw 1:1 (43ms)
    ✓ Deposit after : Reward, ratio 2 USDC: 1 Share (84ms)
    ✓ Withdraw after : Reward, ratio 2 USDC: 1 Share
    ✓ Claim, ratio 1 USDC : 2 shares (41ms)
    ✓ Deposit after : Claim, ratio 1 USDC: 2 Share (46ms)
    ✓ Withdraw after : Claim, ratio 1 USDC: 2 Share (45ms)

LiquidityPosition Library
  #initialize
    ✓ first works
    ✓ again reverts
  #checkpoints
    ✓ zero chkpts (51ms)
    ✓ non-zero chkpts (114ms)
  #liquidityChange
    ✓ increase (43ms)
    ✓ decrease (81ms)
    ✓ overflow
  #marketValue
    ✓ zero
    ✓ ticks -1 1 | current 0 | amounts 100 100 (55ms)
    ✓ ticks -100 100 | current 99 | amounts 100 100 (50ms)
    ✓ ticks -100 100 | current 100 | amounts 200 100 (49ms)
    ✓ ticks -100 100 | current 101 | amounts 200 100 (54ms)
    ✓ ticks -100 100 | current -100 | amounts 200 100 (48ms)
    ✓ ticks -100 100 | current -99 | amounts 200 100 (50ms)
    ✓ ticks -100 100 | current -101 | amounts 200 100 (53ms)
    ✓ ticks -100 100 | current 9001 | amounts 200 100 (54ms)
    ✓ ticks -100 100 | current -9001 | amounts 200 100 (49ms)
  #maxNetPosition
    ✓ zero
    ✓ ticks -1 1 | current 0 | amounts 100 100 (43ms)
    ✓ ticks -100 100 | current 99 | amounts 100 100 (47ms)
    ✓ ticks -100 100 | current 100 | amounts 200 100 (39ms)
    ✓ ticks -100 100 | current 101 | amounts 200 100 (46ms)
    ✓ ticks -100 100 | current -100 | amounts 200 100 (51ms)
    ✓ ticks -100 100 | current -99 | amounts 200 100 (43ms)
    ✓ ticks -100 100 | current -101 | amounts 200 100 (42ms)
    ✓ ticks -100 100 | current 9001 | amounts 200 100 (48ms)
    ✓ ticks -100 100 | current -9001 | amounts 200 100 (42ms)

  LiquidityPositionSet Library
    #create
      ✓ empty (74ms)
      ✓ invalid

  Price Math
    #toPriceX128
      ✓ 79228162514264337593543950336(X96) == 340282366920938463463374607431768211456(X128)
      ✓ 158456325028528675187087900672(X96) == 136112946768375385385349842972707284582 4(X128)
      ✓ 396140812571321687967719751 68(X96) == 85070591730234615865843651857942052864(X128)
      ✓ 0(X96) reverts
    #toSqrtPriceX96
      ✓ 340282366920938463463374607431768211456(X128) == 79228162514264337593543950336(X96) (68ms)
      ✓ 136112946768375385385349842972707284582 4(X128) == 158456325028528675187087900672(X96) (67ms)
      ✓ 85070591730234615865843651857942052864(X128) == 396140812571321687967719751 68(X96) (69ms)
      ✓ 142724769270595988105828596944949513638274662400(X128) == 16225927682921336339157801028812 80(X96) (77ms)
      ✓ 0(X128) reverts
      ✓ 2**256-1 reverts
      ✓ fuzz perfect square (7733ms)
      ✓ fuzz non perfect square (7465ms)

  RageTradeFactory
    #constructor
      ✓ deploys VQuote at good address (19602ms)
      ✓ initializes values (5833ms)
      ✓ governance and teamMultisig is deployer (2604ms)
      ✓ sets proxyAdmin owner (2462ms)
    #initializePool
      ✓ deploys vToken at good address (2836ms)
    #upgradability
      ✓ upgrades clearing house logic (3595ms)
      ✓ upgrades vpoolwrapper (7632ms)

  SignedFullMath
    #mulDiv(uint256,uint256,uint256)
      ✓ 1 * 2 / 2
      ✓ 160693804425899027554196209234116260252220299378279283530137 6 * 160693804425899027554196209234116260252220299378279283530137 6 / 180925139433306555349329664076074856020734351040063381311652475012364265062 4
      ✓ 414483806531955393238086080633490537769116951023545309921836316752188859475 * 254617334707634142805908766695966164585966939630087287033970223131704874 52 /
853057593808315090632009031047011662268295284266263825956061526302542890982 6
    #mulDiv(int256,uint256,uint256)
      ✓ 1 * 2 / 2
      ✓ 160693804425899027554196209234116260252220299378279283530137 6 * 160693804425899027554196209234116260252220299378279283530137 6 / 180925139433306555349329664076074856020734351040063381311652475012364265062 4
      ✓ 297451053434110898026990128712272340709007620172860942821548389230733139 99 * 148271918446338095858440064400208047349298026316367463517847454879180786501 /
601593083663836053454506520528021427246104517455032516637848968543216090439 74
      ✓ -160693804425899027554196209234116260252220299378279283530137 6 * 160693804425899027554196209234116260252220299378279283530137 6 / 180925139433306555349329664076074856020734351040063381311652475012364265062 4
    #mulDivRoundingDown(int256,uint256,uint256)
      ✓ 1 * 2 / 2
      ✓ -1 * 2 / 2
      ✓ -2 * 1 / 1
      ✓ 160693804425899027554196209234116260252220299378279283530137 6 * 160693804425899027554196209234116260252220299378279283530137 6 / 180925139433306555349329664076074856020734351040063381311652475012364265062 4
      ✓ -160693804425899027554196209234116260252220299378279283530137 6 * 160693804425899027554196209234116260252220299378279283530137 6 / 180925139433306555349329664076074856020734351040063381311652475012364265062 4
      ✓ -160693804425899027554196209234116260252220299378279283530137 6 * 160693804425899027554196209234116260252220299378279283530137 6 / 180925139433306555349329664076074856020734351040063381311652475012364265062 3
    #mulDiv(int256,int256,int256)
      ✓ 1 * 2 / 2
      ✓ -160693804425899027554196209234116260252220299378279283530137 6 * 160693804425899027554196209234116260252220299378279283530137 6 / 180925139433306555349329664076074856020734351040063381311652475012364265062 4
      ✓ 160693804425899027554196209234116260252220299378279283530137 6 * -160693804425899027554196209234116260252220299378279283530137 6 / 180925139433306555349329664076074856020734351040063381311652475012364265062 4
      ✓ 160693804425899027554196209234116260252220299378279283530137 6 * 160693804425899027554196209234116260252220299378279283530137 6 / -180925139433306555349329664076074856020734351040063381311652475012364265062 4
      ✓ -160693804425899027554196209234116260252220299378279283530137 6 * 160693804425899027554196209234116260252220299378279283530137 6 / 180925139433306555349329664076074856020734351040063381311652475012364265062 3
      ✓ -160693804425899027554196209234116260252220299378279283530137 6 * 160693804425899027554196209234116260252220299378279283530137 6 / -180925139433306555349329664076074856020734351040063381311652475012364265062 4
      ✓ 160693804425899027554196209234116260252220299378279283530137 6 * 160693804425899027554196209234116260252220299378279283530137 6 / 180925139433306555349329664076074856020734351040063381311652475012364265062 4
      ✓ -160693804425899027554196209234116260252220299378279283530137 6 * 160693804425899027554196209234116260252220299378279283530137 6 / 180925139433306555349329664076074856020734351040063381311652475012364265062 3
      ✓ 160693804425899027554196209234116260252220299378279283530137 6 * -160693804425899027554196209234116260252220299378279283530137 6 / 180925139433306555349329664076074856020734351040063381311652475012364265062 3
      ✓ 160693804425899027554196209234116260252220299378279283530137 6 * 160693804425899027554196209234116260252220299378279283530137 6 / -180925139433306555349329664076074856020734351040063381311652475012364265062 3
      ✓ 160693804425899027554196209234116260252220299378279283530137 6 * 160693804425899027554196209234116260252220299378279283530137 6 / 180925139433306555349329664076074856020734351040063381311652475012364265062 3
      ✓ -999956398182672334764352444574694867837014009380178578648576586343306193 * 173222625028426592203669520984162604581238309531404027707371234109426757 2 /
211096248595112144546496893956168110660368101226809400715178045366493416824
    #mulDivRoundingDown(int256,int256,int256)
      ✓ 1 * 2 / 2
      ✓ -1 * 2 / 2
      ✓ 1 * -2 / 2
      ✓ 1 * 2 / -2
      ✓ -2 * 1 / 1
      ✓ 2 * -1 / 1
      ✓ 2 * 1 / -1
      ✓ 2 * -1 / -1
      ✓ 160693804425899027554196209234116260252220299378279283530137 6 * 160693804425899027554196209234116260252220299378279283530137 6 / 180925139433306555349329664076074856020734351040063381311652475012364265062 4
      ✓ -160693804425899027554196209234116260252220299378279283530137 6 * 160693804425899027554196209234116260252220299378279283530137 6 / 180925139433306555349329664076074856020734351040063381311652475012364265062 4
      ✓ -160693804425899027554196209234116260252220299378279283530137 6 * 160693804425899027554196209234116260252220299378279283530137 6 / 180925139433306555349329664076074856020734351040063381311652475012364265062 3
      ✓ 160693804425899027554196209234116260252220299378279283530137 6 * -160693804425899027554196209234116260252220299378279283530137 6 / 180925139433306555349329664076074856020734351040063381311652475012364265062 3
      ✓ 160693804425899027554196209234116260252220299378279283530137 6 * 160693804425899027554196209234116260252220299378279283530137 6 / -180925139433306555349329664076074856020734351040063381311652475012364265062 3
      ✓ 160693804425899027554196209234116260252220299378279283530137 6 * 160693804425899027554196209234116260252220299378279283530137 6 / 180925139433306555349329664076074856020734351040063381311652475012364265062 3

  SignedMath
    #abs
      ✓ abs(1) = 1
      ✓ abs(-1) = 1
      ✓ abs(3) = 3
      ✓ abs(-3) = 3
      ✓ abs(0) = 0
    #absUint
      ✓ absUint(1) = 1
      ✓ absUint(-1) = 1
      ✓ absUint(3) = 3
      ✓ absUint(-3) = 3
      ✓ absUint(0) = 0
    #sign
      ✓ sign(1) = 1
      ✓ sign(-1) = -1
      ✓ sign(3) = 1
      ✓ sign(-3) = -1
      ✓ sign(0) = 1
    #extractSign
      ✓ extractSign(1) = [1,true]
      ✓ extractSign(-1) = [1,false]
      ✓ extractSign(3) = [3,true]
      ✓ extractSign(-3) = [3,false]

  SimulateSwap
network block skew detected; skipping block events (emitted=13075889 blockNumber13555700)
network block skew detected; skipping block events (emitted=13075889 blockNumber13555700)
network block skew detected; skipping block events (emitted=13075889 blockNumber13555700)
network block skew detected; skipping block events (emitted=13075890 blockNumber13555700)
network block skew detected; skipping block events (emitted=13075891 blockNumber13555701)
    #amounts
      ✓ swap 1.0 WETH for USDC (491ms)
      ✓ swap 100.0 WETH for USDC (218ms)
      ✓ swap 10000.0 WETH for USDC (538ms)
      ✓ swap 1.0 USDC for WETH (1166ms)
      ✓ swap 10000000.0 USDC for WETH (728ms)
      ✓ swap 50000000.0 USDC for WETH (2210ms)
    #onSwapStep
      ✓ swap 1.0 WETH for USDC (94ms)
      ✓ swap 100.0 WETH for USDC (220ms)
      ✓ swap 10000.0 WETH for USDC (815ms)
```

```
      ✓ swap 1.0 USDC for WETH (620ms)
      ✓ swap 10000000.0 USDC for WETH (227ms)
      ✓ swap 50000000.0 USDC for WETH (1602ms)

  StorageLayout
    #ClearingHouse
      ✓ protocol is at 100
      ✓ numAccounts is at 210
      ✓ accounts is at 211
      ✓ rageTradeFactoryAddress is at 212
      ✓ insuranceFund is at 213
      ✓ _paused is at 368
      ✓ _governance is at 418
      ✓ _teamMultisig is at 419
    #VPoolWrapper
      ✓ clearingHouse is at 0
      ✓ vToken is at 1
      ✓ vQuote is at 2
      ✓ vPool is at 3
      ✓ liquidityFeePips is at 3
      ✓ protocolFeePips is at 3
      ✓ accruedProtocolFee is at 4
      ✓ fpGlobal is at 5
      ✓ sumFeeGlobalX128 is at 9
      ✓ fundingRateOverrideX128 is at 10
      ✓ ticksExtended is at 11

  TickExtended
    #fundingPaymentGrowthInside
      ✓ zero
      ✓ current price inside, global increase
      ✓ current price outside, global increase
      ✓ current price inside, global increase with ticks (52ms)
      ✓ current price outside, global increase with ticks (46ms)
      ✓ current price outside, global increase with ticks and extrapolation (54ms)
    #extendedFeeGrowth
      ✓ zero
      ✓ current price inside, global increase
      ✓ current price outside, global increase
      ✓ current price inside, global increase with ticks (38ms)
      ✓ current price outside, global increase with ticks (38ms)
    #cross
      ✓ first cross
      ✓ second cross (85ms)

  TimelockControllerWithMinDelayOverride
    #getSelector
      ✓ getSelector(0x0000000000000000000000000000000000000000000000000000000000000001) == 0x00000000
      ✓ getSelector(0x095ea7b3000000000000000000000000000000000000000000000000000000000000000068b3465833fb72a70ecdf485e0e4c7bd8665fc45ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff) == 0x095ea7b3
      ✓ getSelector(0xa9059cbb0000000000000000000000000000000000000000000000000000000000000004c4add20570dc96ccc87b02bf8cf4ef5e02b6f7400000000000000000000000000000000000000000000000000000000000000000000000000000000000014dc9380) == 0xa9059cbb
      ✓ getSelector(0x12345678) == 0x12345678
      ✓ getSelector(0x1234567890) == 0x12345678
      ✓ getSelector(0x) reverts
      ✓ getSelector(0x12) reverts
      ✓ getSelector(0x0000) reverts
      ✓ getSelector(0x000000) reverts
      ✓ fuzz (550ms)
    #getKey
      ✓ getKey(0x2222222222222222222222222222222222222222, 0x11111111) == 0x0000000000000000111111112222222222222222222222222222222222222222
      ✓ fuzz (487ms)
    #minDelayOverride
      ✓ can not just set minDelayOverride immediately from admin wallet
      ✓ set works (67ms)
      ✓ unset works (57ms)

  Uint32L8Set Library
    #include
      ✓ single element
      ✓ prevents 0
      ✓ repeated
      ✓ multiple
      ✓ limits to 8 includes (126ms)
    #exclude
      ✓ single element
      ✓ multiple elements (57ms)
      ✓ non existant element
      ✓ middle element (41ms)
      ✓ zero element
      ✓ remove element from full array (90ms)
    #numberOfNonZeroElements
      ✓ works

  Uint48 Library
    #concat
      ✓ positive.positive#1
      ✓ positive.positive#2
      ✓ positive.negative#1
      ✓ positive.negative#2
      ✓ negative.positive#1
      ✓ negative.positive#2
      ✓ negative.negative#1
      ✓ negative.negative#2

  Uint48L5Array Library
    #include
      ✓ single element
      ✓ prevents 0
      ✓ repeated
      ✓ multiple
      ✓ limits to 5 includes (75ms)
    #exclude
      ✓ single element
      ✓ multiple elements (61ms)
      ✓ non existant element
      ✓ middle element (39ms)
      ✓ zero element
      ✓ remove element from full array (55ms)
    #numberOfNonZeroElements
      ✓ works

  PoolWrapper
    #liquidityChange
      ✓ adds liquidity (82ms)
      ✓ removes liquidity (71ms)
      ✓ sets tick state to global when tickLower <= tickCurrent (156ms)
    #getValuesInside
      values
        ✓ no trade (71ms)
        ✓ small trade (240ms)
      lp sumB
        no tick cross
          ✓ buy | exactIn vQuote (110ms)
          ✓ buy | exactOut vToken (120ms)
          ✓ sell | exactIn vToken (135ms)
          ✓ sell | exactOut vQuote (137ms)
        single tick cross
          ✓ buy | exactIn vQuote (204ms)
          ✓ buy | exactOut vToken (203ms)
          ✓ sell | exactIn vToken (231ms)
          ✓ sell | exactOut vQuote (226ms)
      lp fees
        no tick cross
          ✓ buy | exactIn vQuote (143ms)
          ✓ buy | exactOut vToken (149ms)
          ✓ sell | exactOut vQuote (168ms)
          ✓ sell | exactIn vToken (171ms)
        single tick cross
          ✓ buy | exactIn vQuote (230ms)
          ✓ buy | exactOut vToken (222ms)
          ✓ sell | exactIn vToken (252ms)
          ✓ sell | exactOut vQuote (277ms)
    #fundingRateOverrideX128
      ✓ should use actual prices in getFundingRate() when fundingRateOverrideX128 is null (53ms)
      ✓ should use fundingRateOverrideX128 in getFundingRate() when fundingRateOverrideX128 is not null (40ms)
      ✓ should not allow fundingRateOverrideX128 to be updated if it is more than 100% per hour
    #admin
      ✓ setLiquidityFee
      ✓ setProtocolFee
      ✓ fundingRateOverrideX128
      ✓ setLiquidityFee owner check
      ✓ setProtocolFee owner check
      ✓ setFundingRateOverride owner check (39ms)

  VPoolWrapper.swap
    exactIn 1 ETH
      ✓ amountSpecified (133ms)
      ✓ swapped amount (56ms)
      ✓ mint and burn (154ms)
      ✓ fee (187ms)
      - sumB
    exactIn 2000 USDC
      ✓ amountSpecified (148ms)
      ✓ swapped amount (68ms)
      ✓ mint and burn (176ms)
      ✓ fee (196ms)
      - sumB
```

```
  exactOut 1 ETH
     ✓ amountSpecified (159ms)
     ✓ swapped amount (66ms)
     ✓ mint and burn (180ms)
     ✓ fee (207ms)
    - sumB
  exactOut 2000 USDC
     ✓ amountSpecified (189ms)
     ✓ swapped amount (74ms)
     ✓ mint and burn (198ms)
     ✓ fee (242ms)
    - sumB

VQuote
  #decimals
     ✓ sets decimals correctly
  #authorise
     ✓ works (39ms)
     ✓ onlyOwner
  #mint
     ✓ works
     ✓ unauthorised
  #burn
     ✓ works (45ms)

VToken contract
  #decimals
     ✓ sets decimals correctly
  #mint
     ✓ works
     ✓ unauthorised
    #burn
       ✓ works (41ms)

VTokenPosition Library
  Functions
     ✓ unrealizedFundingPayment
     ✓ marketValue
     ✓ riskSide

VTokenPositionSet Library
  Functions
     ✓ Activate
     ✓ Update (66ms)
     ✓ Realized Funding Payment
  Token Swaps (Token Amount)
     ✓ Token1 (73ms)
     ✓ Token2 (75ms)
     ✓ Token1 Partial Close (49ms)
     ✓ Token1 Close (55ms)
  Token Swaps (Token Notional)
     ✓ Token1 (73ms)
     ✓ Token2 (73ms)
     ✓ Token1 Partial Close (64ms)
     ✓ Token1 Close (59ms)
  Liquidity Change
     ✓ Add Liquidity
     ✓ Remove Liquidity
  Liquididate Liquidity Positions (For a token)
     ✓ Liquidate Liquidity Position (94ms)

WordHelper
  #slice
     ✓ works
  #keccak256
     ✓ keccak256One
  #pop
     ✓ works
     ✓ works 2
     ✓ popAddress
     ✓ popAddress full
     ✓ popUint8
     ✓ popUint8 full
     ✓ popUint16
     ✓ popUint16 full
     ✓ popUint32
     ✓ popUint32 full
     ✓ popUint64
     ✓ popUint64 full
     ✓ popUint128
     ✓ popUint128 full
     ✓ popBool
     ✓ popBool bad value
     ✓ popBool full
  #convertToUint32Array
     ✓ convertToUint32Array empty
     ✓ convertToUint32Array one
     ✓ convertToUint32Array two
     ✓ convertToUint32Array eight
  #convertToTickRangeArray
     ✓ convertToTickRangeArray empty
     ✓ convertToTickRangeArray one positive
     ✓ convertToTickRangeArray two positive
     ✓ convertToTickRangeArray five positive
    - convertToTickRangeArray one negative
    - convertToTickRangeArray one negative 2


759 passing (20m)
6 pending
```

Code Coverage

Overall the line and branch coverage are improvable, with around 80% and 70%, respectively. One file was not tested at all (`contracts/utils/Multicall.sol`), while others do have unit tests but were not run during coverage testing (`contracts/oracles/ChainlinkOracle.sol` and `contracts/utils/TimelockControllerWithMinDelayOverride.sol`) and several lines in `contracts/protocol/clearinghouse/ClearingHouse.sol` (the main contract) have not been covered. Notably, of the untested lines, several issues had been identified (L150-L157: QSP-1, L290-303: QSP-4 and L528-L530: QSP-3), which may have been identified beforehand given corresponding tests. We recommend adding tests to cover the missing file (or remove it, if unused), missing lines and improve the line and branch coverages to be above 90%.

Update: After the re-audit the coverage has been improved by at least 10% in both line and branch coverages to 90% and 80%, respectively. As some tests were failing under coverage due to timeouts or being stuck in a pending state, the coverage may be higher. Noteably, the previously untested lines that led to the high severity issue QSP-1 in L150-L157 of `ClearingHouse.sol` have now been covered by tests (now corresponding to L183-L190).

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Lines
extsloads/	80	100	68.75	80.14	
ClearingHouseExtsload.sol	80	100	68.75	80.14	… 528,549,570
interfaces/	100	100	100	100	
IClearingHouse.sol	100	100	100	100	
IExtsload.sol	100	100	100	100	
IGovernable.sol	100	100	100	100	
IInsuranceFund.sol	**100**	**100**	**100**	**100**	
IOracle.sol	100	100	100	100	

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Lines
IVPoolWrapper.sol	100	100	100	100	
IVQuote.sol	100	100	100	100	
IVToken.sol	100	100	100	100	
interfaces/clearinghouse/	100	100	100	100	
IClearingHouseActions.sol	100	100	100	100	
IClearingHouseCustomErrors.sol	100	100	100	100	
IClearingHouseEnums.sol	100	100	100	100	
IClearingHouseEvents.sol	100	100	100	100	
IClearingHouseOwnerActions.sol	100	100	100	100	
IClearingHouseStructures.sol	100	100	100	100	
IClearingHouseSystemActions.sol	100	100	100	100	
IClearingHouseView.sol	100	100	100	100	
lens/	15	0	28.57	14.63	
ClearingHouseLens.sol	37.5	100	37.5	37.5	… 116,124,145
SwapSimulator.sol	0	0	0	0	… 149,160,167
libraries/	95.77	87.4	92.79	95.58	
Account.sol	96.23	82.61	88.46	94.5	… 437,444,657
AddressHelper.sol	100	100	100	100	
BatchedLoop.sol	100	100	100	100	
Bisection.sol	100	100	100	100	
Block.sol	75	50	100	75	20
CollateralDeposit.sol	100	83.33	100	100	
FundingPayment.sol	100	100	100	100	
GoodAddressDeployer.sol	100	100	100	100	
LiquidityPosition.sol	97.8	85	100	98.94	391
LiquidityPositionSet.sol	93.55	77.78	93.75	90.77	… 348,349,350
PriceMath.sol	100	100	100	100	
Protocol.sol	91.07	77.78	94.12	91.07	… ,90,169,234
SafeCast.sol	100	50	100	100	
SignedFullMath.sol	100	100	100	100	
SignedMath.sol	100	100	100	100	
SimulateSwap.sol	95	69.23	100	94.44	148,150
SwapMath.sol	100	100	100	100	
TickBitmapExtended.sol	100	100	100	100	
TickExtended.sol	100	100	100	100	
Uint32L8Array.sol	96.88	100	100	97.3	114
Uint48.sol	100	100	100	100	
Uint48L5Array.sol	96.88	100	100	97.3	114
UniswapV3PoolHelper.sol	93.33	75	100	93.75	44
VTokenPosition.sol	80	100	83.33	80	90,91
VTokenPositionSet.sol	99.01	90.91	100	99.01	87
WordHelper.sol	77.5	100	75	82.35	… 188,192,212
oracles/	100	75	100	98.18	
ChainlinkOracle.sol	100	75	100	98.15	63
SettlementTokenOracle.sol	**100**	**100**	**100**	**100**	
protocol/	100	100	100	100	

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Lines
RageTradeFactory.sol	100	100	100	100	
protocol/clearinghouse/	83.58	60.91	90.7	84.46	
ClearingHouse.sol	83.59	60.91	91.89	84.49	… 502,504,526
ClearingHouseDeployer.sol	100	100	100	100	
ClearingHouseStorage.sol	100	100	100	100	
ClearingHouseView.sol	80	100	80	80	25
protocol/insurancefund/	100	100	100	100	
InsuranceFund.sol	100	100	100	100	
InsuranceFundDeployer.sol	100	100	100	100	
protocol/tokens/	100	70	100	91.3	
VQuote.sol	100	75	100	88.89	29
VQuoteDeployer.sol	100	100	100	100	
VToken.sol	100	66.67	100	90	45
VTokenDeployer.sol	100	100	100	100	
protocol/wrapper/	100	88	100	98.59	
VPoolWrapper.sol	100	87.5	100	98.54	81,102
VPoolWrapperDeployer.sol	100	100	100	100	
utils/	84.62	68.18	96	86.67	
Extsload.sol	100	100	100	100	
Governable.sol	100	87.5	100	100	
Multicall.sol	0	0	0	0	… 19,20,23,26
ProxyAdminDeployer.sol	100	100	100	100	
TimelockControllerWithMinDelayOverride.sol	100	80	100	100	
constants.sol	100	100	100	100	
All files	**90.65**	**79.05**	**88.55**	**90.58**	

Appendix

File Signatures

The following are the SHA-256 hashes of the reviewed files. A file with a different SHA-256 hash has been modified, intentionally or otherwise, after the security review. You are cautioned that a different SHA-256 hash could be (but is not necessarily) an indication of a changed condition or potential vulnerability that was not within the scope of the review.

Contracts

```
6784546c137dd294a6b4b1232156ee9ce56c154a164b11101cccb4c07e118108 ./contracts/oracles/SettlementTokenOracle.sol
fcb84b672b1aea23889be9809a61bd315a6f114188ea8757e8302a5d1879e021 ./contracts/oracles/ChainlinkOracle.sol
8eb390948013699b39ea5a1fde9852e0cde084a90d0e0595b93ae139b0e9ef0a ./contracts/libraries/VTokenPosition.sol
fe34be37551b2adc55555f57cb3406f3e76cd4ae769fcdef6415d13ec5a07295 ./contracts/libraries/CollateralDeposit.sol
12bc967d24376f2e0c46d118d387f16b4a5546f1a1a3359d1d7fb632a437ad3d ./contracts/libraries/SimulateSwap.sol
13b953433ff3d8cdbfb4bf66960f447b0c3696b38a06f70f2df3cf709943179f ./contracts/libraries/Bisection.sol
b37f149d14cc18f739a3c8167fdfbc3dfd97d4f930e17a998e28a0c3b08c944f ./contracts/libraries/TickExtended.sol
b1402d9e4a1fe806a2562bdf80211c8bb6888b37d8d6e3ee81c9600d48e80979 ./contracts/libraries/Block.sol
ce3af755b1d7402690eed9aa47ce75833283cb003582bb37e81a28403799c735 ./contracts/libraries/Uint48L5Array.sol
5138f9a307d5414498236aeb628979f9326e605940af02d6d41948c245ea2feb ./contracts/libraries/SafeCast.sol
145138a588b021047d23ce032714699ce0adffa41967d49b0e9839c6862a33bb ./contracts/libraries/GoodAddressDeployer.sol
c11699a8cd28fb85a2f384d7b810d497c89606bad6db59c9b7f1ed8f5757d7f5 ./contracts/libraries/LiquidityPositionSet.sol
99a7b44b1296dc17e3ecb968dfbdbc88264dc2a219e0c752f51ba341bed62e0a ./contracts/libraries/Account.sol
2293fd0dca2872b9587e27a658b9715f0e1a8733e90d05d9fe98b1bfd7cf19e6 ./contracts/libraries/SignedFullMath.sol
b17013f9e64034316b91d46969d2a905f6f920032c1a77b77b02454b166ebfd4 ./contracts/libraries/Protocol.sol
011f6eed49e385727d1f21fb70cef2af837d32f310e788e4c2c1799fe80b6324 ./contracts/libraries/WordHelper.sol
717c455603b8c0639bce26ade03e7eff3d3554e8f5aa0c9d3ccc0b69779bf026 ./contracts/libraries/TickBitmapExtended.sol
451986ec88d7efa03ee49b8664c25bb21c0507d46d6180306f1a5beb3187bbd8 ./contracts/libraries/UniswapV3PoolHelper.sol
4b9495d0850a501cd42c54ca2e62f5cb200037939e5564229b9cfca77815e323 ./contracts/libraries/AddressHelper.sol
91ad238d92c078fbf57dc2a11b95cd081f4164dc1c064124d6968550c1244ef0 ./contracts/libraries/Uint48.sol
```

e6dc1e29a85ad838a0ae061fe1d620718274ceb884c4739033aa700297a589eb ./contracts/libraries/PriceMath.sol

736d9f5d211c93fb1a546c3edcd041438cd99439c63f33c36afbc3a6111cd786 ./contracts/libraries/SwapMath.sol

b7c0b96c15fe4b3d6544d3d0f1f9e74f51699112a0da694e610a66d1378bb742 ./contracts/libraries/Uint32L8Array.sol

e2112af52700edd9c461e8294a358bbf42b168a06f73355a75ab7c6d020a9f66 ./contracts/libraries/SignedMath.sol

b11c8b37892bb81b9a2e60305a8125c58f29001b766a82e75832b12763aab368 ./contracts/libraries/VTokenPositionSet.sol

b273186318b120dfecb9399a9af22612430f139cbb100fd230de721b69ed9b84 ./contracts/libraries/FundingPayment.sol

71cc233b2d745636c83ba524297ceeb42f0494a4fb94c38166676bd7e04e652e ./contracts/libraries/LiquidityPosition.sol

f7de70851950838c2280582bf232d5dea91f098db48e78bba5a00457d80a1909 ./contracts/libraries/BatchedLoop.sol

3ffb97256ebd3e8e35bab713ce3b413330302f38ce8a58d04414dfb9ddd586b0 ./contracts/utils/TimelockControllerWithMinDelayOverride.sol

b0a056862fafbe94e3ff748de40dd50dda61e47ce83e4e5bf8446ff9b6c91d95 ./contracts/utils/Governable.sol

6e4360697adc1d1ddda0155f8f0cc8c4f5eb4237f484780972de56a8f9affcb9 ./contracts/utils/constants.sol

04fc0f6cf5fd6a39585a0ac726673e2d6f71c4be7ed7bce24bf9185e080302fc ./contracts/utils/Multicall.sol

91c2a0491818ceb51b0d531c88a9b2697ed090ee6adffa9883d0509190f47d70 ./contracts/utils/Extsload.sol

3f52935a5961c735f385bcfd10667361c9d55335201632d0bb9fc01bac290fb0 ./contracts/utils/ProxyAdminDeployer.sol

83290cbee4fb9feae763b99b77a7c29b744ad3d57fbeeea9de9109bad300d1b0 ./contracts/protocol/RageTradeFactory.sol

af0b5cc34b7084aab94ec284ffe107b2491aff309ef99e88e9f9168f6206ef0c ./contracts/protocol/wrapper/VPoolWrapperDeployer.sol

deba406e942e8eab469b4b793b9169e358ae329e449a55925cf58cb884abad3c ./contracts/protocol/wrapper/VPoolWrapper.sol

f36e3b7ff5db8dc75f0c1fa79847ca8d7222401ef1024b955c2b0ef380b50fbb ./contracts/protocol/tokens/VQuoteDeployer.sol

64af9079ca028358c1adbe3a439bfaea06266d46117f5c332969d71198fbb2d4 ./contracts/protocol/tokens/VToken.sol

860c445bb2063708e49f3cb3c65d2ad4a5157375c49618691f65fade23719fe1 ./contracts/protocol/tokens/VTokenDeployer.sol

9ad7a5450668509d23cd69b9bea1871d52838e1b820bba486cc9ab8070f8a37f ./contracts/protocol/tokens/VQuote.sol

a4fe76681ab82f9670db7399229325637ec73eb5be672898c70e608b6dd70572 ./contracts/protocol/insurancefund/InsuranceFundDeployer.sol

8e3add4800f71ae7f2f4ac065f2b113cc14a592a1989f7acbe8c194fd5181450 ./contracts/protocol/insurancefund/InsuranceFund.sol

ae6700cb2b098c560a9d961eafc049df49ba2f99899f5668d7dccf1d863e8e73 ./contracts/protocol/clearinghouse/ClearingHouseStorage.sol

187fec0f5623c80028f887aeaac1df65b39d0eb08a5b5cbbb53647ca2480e4e4 ./contracts/protocol/clearinghouse/ClearingHouse.sol

cd9ba696a8d0297fa77a20acd569e9176cc2970ac521af9ddd1f13aebc0d18ba ./contracts/protocol/clearinghouse/ClearingHouseDeployer.sol

1d11644f8e33adf23a20e215a5ec1d04437716756320eb5cf38309dee6b34dce ./contracts/protocol/clearinghouse/ClearingHouseView.sol

9a55bd64b2f0f1e5b4f3b9281c9b67b468ffb9cf899e29043512eb927d39d045 ./contracts/interfaces/IOracle.sol

d7ce59dafbc6eb0fcf93b0d570ed0c63338a08f514e01b0bcc20441166bf95a3 ./contracts/interfaces/IVQuote.sol

4c9ac6cf8d21c94ae6d15ea8983474f8b028beb14fff72a129d019777e06f41f ./contracts/interfaces/IInsuranceFund.sol

8ef5e08a186cbdf08dfd3d59104391fc081adb8af0791603ac1e94a808536e9e ./contracts/interfaces/IGovernable.sol

8d4bbe9e891f640ea98cf8aea4b1394d8053a5188f877c8aae0a1ce6a37ed22f ./contracts/interfaces/IExtsload.sol

2163757b443f7a6f5c69d1389fe8802606e82dbb27026d5a15a33d40b9a81299 ./contracts/interfaces/IVPoolWrapper.sol

2f660d15cdcb45a12fabcb0354cff264e49e488582d7033f230e89a53538bd3b ./contracts/interfaces/IClearingHouse.sol

cf8a52f283f41df0c75037e37c720a572ec1534f1a97cd0df406717be739578d ./contracts/interfaces/IVToken.sol

bf65408cb896033729f64dc2919f7cd5b3594c49885bbc7eb1697763b33c3ea3 ./contracts/interfaces/clearinghouse/IClearingHouseEnums.sol

d8f6034c09e8c16e945dc48d6f55de853f3aad652cc2190fc6286f142564329b ./contracts/interfaces/clearinghouse/IClearingHouseOwnerActions.sol

cba9a0313610b1dad85b8a53b973bdeb2f1d09aac847c3a3279a1eddc4b953d4 ./contracts/interfaces/clearinghouse/IClearingHouseView.sol

b36942f14f624b13dd33f6f1264b7423bb7e648820301952d0332188be18a84e ./contracts/interfaces/clearinghouse/IClearingHouseSystemActions.sol

77904983999183d71107900a9dbd61e3e292c3332bfa2da3b2c8a5585c5ccecb ./contracts/interfaces/clearinghouse/IClearingHouseEvents.sol

35a42cec9d953b1f3cd802fa1b83547bbafdadfbebc6b6a6388aa8bf24621453 ./contracts/interfaces/clearinghouse/IClearingHouseStructures.sol

364225cec23e353fa8e33b20fa837d2bf193f3e0f5965345615d1557b13c5054 ./contracts/interfaces/clearinghouse/IClearingHouseCustomErrors.sol

6e0977e46a23e7739a0cdb52c865625a3ba55014701d9995f582491f95bc092a ./contracts/interfaces/clearinghouse/IClearingHouseActions.sol

b6e2a2e2e2fdba8e86e87866235babead4877d8af5b5f7500c036d4b651fcfff ./contracts/extsloads/ClearingHouseExtsload.sol

eb778063a5244d7c26b8b16ac9de6e1ced60d011a0c00bc9de7ce6661b3d3ee2 ./contracts/lens/ClearingHouseLens.sol

0120716b958be7602b0df4a0854b908b780c988d4453b333dc23f7fd1a842cac ./contracts/lens/SwapSimulator.sol

eb3bbdd29e3208482f7d2e7caab630748c00103e221f4afd4a801a69019479bb ./contracts/test/TickExtendedTest.sol

4f27e3594d0bdce8fa7572d090e9ff4c1c68b6ad99c0e7fe55dcce17587c63cb ./contracts/test/GoodAddressDeployerTest.sol

516e822908211f9d3dd7fdf359255d4c010aef3fc444f7113256fa6d3e83f549 ./contracts/test/FundingPaymentTest.sol

7de3a8123b5f4ce52fc137b0501f5744b6a0e4b0b80f026e3b5b1c46a6ae606e ./contracts/test/VTokenPositionTest.sol

946421f27bae6e908e09a8a721ea742d3f1f844d7efccba86f849d30635371b5 ./contracts/test/Uint48L5ArrayTest.sol

1318691a6479eb76a9a1be905aab5921576d9338eec5ba4de8b66ccaec497564 ./contracts/test/Uint48Test.sol

1d1cdbb3f213d7573579fcfb5cb91612724aae6b430874d7e530f20c9c527567 ./contracts/test/Governable.sol

dfc376a3bec50e5d354150539d6751ea3d74313bf6e623a47a2b2e7b6e3dfe39 ./contracts/test/VTokenPositionSetTest.sol

3a4fc2dd6ff08dce2b6034637ec33b13cd2846c30791926c93362ad772b51bb3 ./contracts/test/ClearingHouseExtsloadTest.sol

800e7f6542d858b4e0c58a473eccd5b182df860a7e0a25fd4678ab0472d83f03 ./contracts/test/PriceMathTest.sol

bce17ec023de5329f1e8bc33733f2d77a941e5614f0748022f9da25367ef3047 ./contracts/test/ClearingHouseTest.sol

f7732d64c27ffe5c4352c86e4632c032d05a3e51507ca0b46c0311ae3abc7bd8 ./contracts/test/ExtsloadTest.sol

aeadce832e112ddce75683b346f33e048a09a03e226e346b5a51360af7c94a39 ./contracts/test/WordHelperTest.sol

550d1fbdb5b2a95a63ca9cc7f0f0120c3d1d744242d480386c6e593840c77912 ./contracts/test/SignedMathTest.sol

47806d643aa14a740d371d92bcf8baba61851dc4a8f734cf8a3aa03dd2235b2c ./contracts/test/SignedFullMathTest.sol

59f140feeebc9f6541e621287b2fccaf8a15f805063cab0a8498ca890b264f11 ./contracts/test/AccountTest.sol

7d0d9e4d7984a8e572db2824e24a328f2cd5ce2273280f60ae141c07aa9203f3 ./contracts/test/TimelockControllerWithMinDelayOverrideTest.sol

2fae4f478dc3c909a122761295cf656d57426096d13848dd3c15b3e2861b9260 ./contracts/test/BisectionTest.sol

665991e6a02bfc52241f5561827a4ab37d5f2c44042ec170cf5855ec14ad19d3 ./contracts/test/LiquidityPositionSetTest.sol

e7866e97e6572d9c6507c8d0e4d6615c8d80acfbb03cf6f86f6a681e0730456e ./contracts/test/CollateralDepositSetTest.sol

c0bf8fa7a55acc210fca45969564d5e6b439d8db8a39ab34311219835672a3ae ./contracts/test/SimulateSwapTest.sol

836142db408a4c8631411bc3c8d887c5d4dc40a0e4788912f7eda9cb9ce472b7 ./contracts/test/VTokenPositionSetTest2.sol

08575952b2045ff9044a4b476df01f8a6e274b5fa9123a896e698516c7247db8 ./contracts/test/Uint32L8ArrayTest.sol

3f82215b4143efc26f585e262809c20c32ed8e71dcbbf4a01767a876e9109b22 ./contracts/test/LiquidityPositionTest.sol

8fb5c3fb9e21d8de01fc8e4564020038e4862205394efada9a4c356e458b5f61 ./contracts/test/BatchedLoopTest.sol

622bf0ea4757b6cd86953f697954f6564ba7cd8084731d2dd4e201bb93e6763f ./contracts/test/mocks/ArbSysMock.sol

fdb2a871c96c3ccb30cf189fef41e4e0b98c42e6dc93bce348ab0ce725c6a850 ./contracts/test/mocks/AccountProtocolInfoMock.sol

f1b244c7cef3aa19634fbbe75e6b2c3449cb4391c37e767057a3a98a1653d3a8 ./contracts/test/mocks/ClearingHouseDummy.sol

8f0e4b1bbc1a510ab628844bf2e665eda88b3bb37ea9acc9fa42e97b448c9b84 ./contracts/test/mocks/VPoolWrapperMock.sol

20a187dcb4be63a348ca7194505a5e42ef3f161c4036fb3a5bdd08accc5028c9 ./contracts/test/mocks/RealTokenMock1.sol

80e69a27b7da3b341a77296d379a8a13adc264a7b78bb05fdbe091641ca1eff0 ./contracts/test/mocks/SettlementTokenMock.sol

677b9caedcf20373e9083234eeb48893edab60ab36eb7003a6b5fe961ff62c37 ./contracts/test/mocks/MockAggregatorV2.sol

f910a0476a753caeceb5902f68539db92f0ae0d58fd20b853da528175b8bd0d2 ./contracts/test/mocks/VPoolWrapperMock2.sol

e5233764827fe123b037d67557b998d0912ce8b3d93cf2378cceee69c37c1e90 ./contracts/test/mocks/RealTokenMock.sol

efe4e1bb241712f91096e2398fdd0c1497ad6a619dca4e896c7b098f0399d445 ./contracts/test/mocks/OracleMock.sol

a3ba7c34de526beceecb41ec497ae0fb711e9f208467a5599faa8269a0fedb7b ./contracts/test/mocks/UniswapV3PoolMock.sol

4f9f25c0ea255206f78ced532025f86361c4425af135342aa767c1dc4f21cd60 ./contracts/test/mocks/VPoolWrapperMockRealistic.sol

Tests

95468a66d4888df398098ae44cff4a1478d8b29c730ad948038a2a305ae61030 ./test/helpers/dummy-constants.ts

437159e0e7ff9a6d2c01a682fb1fc6f7a46a857307e56db2f2288f475cfe71c1 ./test/helpers/steal-funds.ts

40d50645da35e57d3fc8de4ccb4fdba3f05561e4ce99ab70f295a7190e2deb5e ./test/helpers/impersonate-account.ts

d82176249e2be643280377bdbd37f00559a6e25693a29c9747060b01efa3898f ./test/helpers/setup-vPool.ts

a4ffdd881e9ee561f08fc174326808701ce7db69c9f2e3e86aae3fac646805ab ./test/helpers/setup-clearinghouse.ts

1e498510e808984d57b4d536613d49cf7901a7d60c4c6bd7f987a410da1210ed ./test/helpers/setup-general.ts

bb53bd8c5646ef6349749b6df6e905714e35d6dc677302f59694d217d7012b48 ./test/helpers/setup-wrapper.ts

bf8bd1083548dac17aacfb27e644dff1ec771f9bb9cacbbef7e57ef6ae5a82b8 ./test/helpers/real-constants.ts

e511909b05bc56806c078847c580b9738c326c345eef3ccb6c98ec03b8f629ea ./test/helpers/get-storage-layout.ts

93dd9fd7fe948738ccd5bf1fc5b1b6191f5c900ec1f091854c2d680c6a9c1ee5 ./test/helpers/mainnet-fork.ts

731f252b210ec0ce1402335cb675d0c2b0991049fef181b60614deba4d17dd2e ./test/units/Bisection.spec.ts

d2724a65741f3aea9a48f8939f708361367dc113435dc23505e00851cffd08f8 ./test/units/AccountBasic.spec.ts

35855527def1b1cc06bad649a39291f5bc0e74082505bd0c27cd7a4c4d861e27 ./test/units/VToken.spec.ts

bd79a74596246f3f6330a0955a5636c9273b07139c52701d66ab0b36d9752ebd ./test/units/AccountRealistic.spec.ts

9f960bfdb4999724ef05d71bfcfc86b45ec35d19e0b9d311f34c3d13050c6588 ./test/units/Uint48.spec.ts

8e898d0bda3d412859c2c8929667a5997adf868c30f4ea0ca575fd7d6d734cb9 ./test/units/Governable.spec.ts

0f1f6f2e67628bcfaef3a3fd20a4c7e27111c9a1f8632a2fdb9c6f5a05f94aec ./test/units/ChainlinkOracle.spec.ts

c1a76e58e9e544783279096356cea769b4553ce699be15a140c7f4e547392f29 ./test/units/VQuote.spec.ts

eec22fff9604f45e9be2473963fe7aa35ac4e9e914392d2c1c6fbc2a94e84ff0 ./test/units/BatchedLoop.spec.ts

9713c66bc5e8bec07ed1e1ac9d1a189ab222913314e526e980248356aa2063f4 ./test/units/SignedMath.spec.ts

64f98d4c50a0721d4092f41ee9f3272a1179e7bf07d98fad195f8f5f49df8ab8 ./test/units/Uint32L8Array.spec.ts

52d78fab2fc61d19f15a913a1c015284aa60c154412625bf1534009df096cb8b ./test/units/Extsload.spec.ts

cc27941df82539d8aeb4cfb9eff30ecc0a3806b6019064fad2ca92546890818f ./test/units/StorageLayout.spec.ts

2811fd367781aabb263cf709638fe54b603b51163263346825268e993d0f747e ./test/units/WordHelper.spec.ts

22c0b306f5786eccea2f97032af59cb3254bab125b19904e7588706a0951f7c9 ./test/units/RageTradeFactory.spec.ts

c7667b5e50fe43fe22411c71b511dbc74999b5fdbdc0c1a63a9f6db33830fe36 ./test/units/SimulateSwap.spec.ts

bb4a9ffdbe7a4e071b4ef22a3214a50aa920489bd8601a8bc601e24f2831b46a ./test/units/VTokenPositionSet.spec.ts

88c1c0540dc1c15cb4e2697bd280d0f2990f0c4893d3171c451a89c0e77c0b02 ./test/units/VPoolWrapper.spec.ts

09f02a8917d30aaa6a39f121ce7186b2f4b6c2bcfb6616c25a59d00c1f07385a ./test/units/TickExtendedTest.spec.ts

67ca5d2baa2f94eff4cae6a431382ef66c4b19c4f2f0557303670ca3f8e03e9e ./test/units/SignedFullMath.spec.ts

b473d1094f2461a451a3b994c6230da130ac1d1334886617da698a389c4408a4 ./test/units/VPoolWrapper.swap.spec.ts

a5bf4ad34e3e99ccd104368adb1acc575cf37ddd89a396c9b1fdbb1f77d62805 ./test/units/FundingPayment.spec.ts

0f8964a58d82bc39c60e7c9a77d00f5dfc873692651db074adff9317ff4bf9b1 ./test/units/LiquidityPositionSet.spec.ts

3fc9cc25fdc52b31ec6dc28c8f001e10972a4f019ab5561b107106a116266c3e ./test/units/PriceMath.spec.ts

2afcdcbb5fa9dabe3c4827aa35ed9a88f4abb65f666163a15e3f36f4e7b04721 ./test/units/LiquidityPosition.spec.ts

3811b5f147693fbb28b6b7c4eb0aa27daff17f4ccdf90011f89c2e68681b763e ./test/units/TimelockControllerWithMinDelayOverride.spec.ts

d6dbe35d9469bc0261062eeb12f21d964445000957308469143c549451550264 ./test/units/VTokenPosition.spec.ts

568b1bb60e21c5822f7b752cd7c4525597d7b514bf2149cc01256f605b6d787b ./test/units/GoodAddressDeployer.spec.ts

bd349f0474200ce94d6fddf0a0c382e62b5dab467438189dd8c1ecd83b43d02f ./test/units/CollateralDepositSet.spec.ts

f65a113a7f65f2792c676094d51a1657ab40d27b25622820976508e326d67aff ./test/units/Uint48L5Array.spec.ts

6b5a7af8f3e152e0e44b25f1686b77de35d41668c24552b5be8aed7f84f95197 ./test/units/InsuranceFund.spec.ts

29aae5a6d0fa276b8095e9275ebcfe37a6a69f03ee427d08405cdeb65f87d0c2 ./test/fixtures/vETH.ts

e6e15314dd215660cc116c1e0855ad6ebc80cab41fd35f61dae0c17ff433067e ./test/scenarios/ClearingHouseScenario9.spec.ts

6f746019e00b2ad3538d0375bd8428118b03d4695b8f4d182676991da83b802b ./test/scenarios/ClearingHouseScenario8.spec.ts

c70de4898a0d07726b5fb5e710ea50690c8d40a084a32e3f3f1374f9334f8368 ./test/scenarios/ClearingHouseScenario6.spec.ts

c96311cfd1d09c006794fe2845976ebf379ca1612bde45ed2fb444b76bfeaffe ./test/scenarios/ClearingHouseScenario2.spec.ts

9d45f162d0e9006075252872622286d4907077fe7fe7311e2148236f9bf13508 ./test/scenarios/ClearingHouseScenario5.spec.ts

259e54def521fa02caa6f560cc4d6c054bd3e833c04e9e384fd92c34b5f57d13 ./test/scenarios/ClearingHouseExtsload.spec.ts

3508f0528f6b745f3ef39b1766489a8fd870b5440c9d15c593f273e9e1358fda ./test/scenarios/ClearingHouseState.spec.ts

0528dd77768c114aa4b26ae22cff59b3ff23e4af109973800fa49115307f1b40 ./test/scenarios/ClearingHouse.spec.ts

b5a74b4a3c8f24f6213d7959fa601e7e88ee6183f466b5d2b00b9637cc1f4038 ./test/scenarios/ClearingHouseScenario4.spec.ts

569fc773404f70c2390aaf8adb6c6095bba977ad676dab7169eaf99feb5b402a ./test/scenarios/MarketValueAndReqMargin.spec.ts

e8b2b5fa73fc267cc937f3f2637fdcacfdd0ff42f43beb12ab025676972a01e3 ./test/scenarios/ClearingHouseScenario1.spec.ts

1cd0a918c398a23e5e1967a41df6811e6f31161d25fa7a59154176c308ad1893 ./test/scenarios/ClearingHouseScenario7.spec.ts

386c0c6661162f80e137bfa78b9eb276cc8b61975e060f00a7b5e0d651658b5b ./test/scenarios/ClearingHouseScenario3.spec.ts

Changelog

- 2022-05-27 - Initial report
- 2022-06-15 - Final report

About Quantstamp

Quantstamp is a Y Combinator-backed company that helps to secure blockchain platforms at scale using computer-aided reasoning tools, with a mission to help boost the adoption of this exponentially growing technology.

With over 1000 Google scholar citations and numerous published papers, Quantstamp's team has decades of combined experience in formal verification, static analysis, and software verification. Quantstamp has also developed a protocol to help smart contract developers and projects worldwide to perform cost-effective smart contract security scans.

To date, Quantstamp has protected $5B in digital asset risk from hackers and assisted dozens of blockchain projects globally through its white glove security assessment services. As an evangelist of the blockchain ecosystem, Quantstamp assists core infrastructure projects and leading community initiatives such as the Ethereum Community Fund to expedite the adoption of blockchain technology.

Quantstamp's collaborations with leading academic institutions such as the National University of Singapore and MIT (Massachusetts Institute of Technology) reflect our commitment to research, development, and enabling world-class blockchain security.