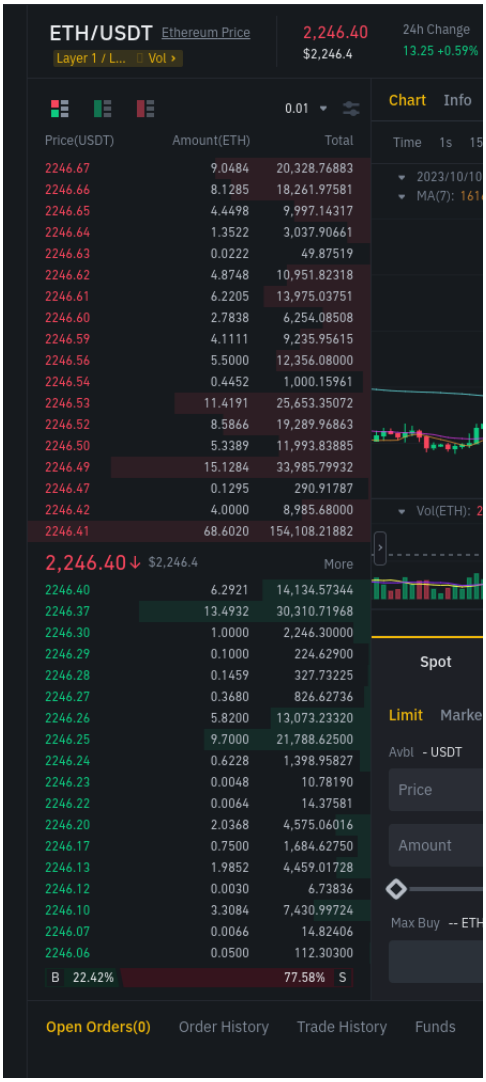
Take-home task
E4 Capital
Spring 2024
Estimated completion time: 2.5 hours

Building a Binance order book

1 Introduction

A centralised exchange (**CEX**) helps to match orders between traders. When traders submit orders that cannot be matched against existing orders immediately, their orders are entered into an **order book**. This is a data structure that collates the prices and quantities at which people want to buy or sell a cryptocurrency.

The following is a screen capture for the order book for Ethereum, with prices quoted in USDT:



The greens are the **bid** prices for Ethereum (the prices at which people want to buy Ethereum), and the reds are the **ask** prices for Ethereum (the prices at which people want to sell Ethereum). The middle column refers to the quantity of Ethereum available to bid/ask for at the given price. The big number $2,246.40$ represents the mid-price, which is the price between the highest bid on the book and the lowest ask. Prices are ordered from "most attractive" (highest bids, lowest asks) near the mid-price to "least attractive" as you move from the mid-price towards the periphery.

For traders that want to maintain a local copy of the order book for trading, Binance offers two types of order book updates:

1. Snapshots: You can get the current top few hundred price levels for a symbol (top meaning the highest bid prices or lowest ask prices). You do this through an HTTP GET request.

2. Depth updates: You can open a WebSocket stream to receive updates about the latest changes in the quantity at various price levels. For example, if you receive an update that the quantity for the bid price $2246.26 is now 6.0, that means there are 6.0 Ethereum coins available to buy at the price of $2246.26 USDT. This quantity refers to the absolute quantity available - it is not a delta.

2 The task

Write a program that asks the user for a Binance-style symbol on startup, like `BTCUSDT`, `BNBBTC` etc, and prints out the top 5 levels of the orderbook as and when a snapshot or a depth update comes in. The program should run until it is stopped by `Ctrl-C`.

The method you should use to maintain your orderbook is described in detail in the Binance docs. The presentation format of the orderbook is up to you. We will not penalise numerical imprecisions due to finite-precision numerical parsing. You may use any third party libraries you like.

Write your program in C++. Partial submissions accompanied by clear documentation about their implementation progress **will** also be considered. Prioritise correctness and extensibility. Candidates with correct submissions will be invited for an interview.