

Stock Broker

In this assignment, we simulate how stock trading works and for simplicity, we consider stocks of the same company. A stock broker uses the priority queue to organize a bid queue and an ask queue. A bid queue is used to organize all the buying orders of a stock, and an ask queue is used to organize all the selling orders of this stock. Each bid or ask request has a suggested price and the number of shares. A trade will occur if the highest bid price exceeds or meets the lowest ask price. When this happens, the broker will step in and buy the stock from the one offers the lowest ask price, and sell the stock to the one who offers the highest bid price and make profit when these two prices are different. If the highest bid and the lowest ask have different number of shares, then the broker splits the larger request and continues the trading.

The Request ADT

We store each bid or ask in an object of `Request` type. The `Request` class has the following ADT.

- `__init__(self, price, share)` - Receives the price and the number of shares for a bid or ask request.
- `__eq__(self, other)` - Receives a `Request` and returns `True` if both requests have the same price and shares.

The Broker ADT

We use `Broker` with the following ADT to simulate the stock broker.

- `__init__(self, bids, asks)` - Receives a list of bid requests and a list of ask requests and store them internally.
- `trade(self)` - Returns a tuple containing the bid request with the highest price and the ask request with the lowest price if the ask price is not greater than the bid price, and `None` otherwise.
- `tradeall(self)` - Trades all stocks as long as the ask price is not greater than the bid price and returns a list of trades (tuples of bid and ask requests).
- `getprofit(self)` - Returns the obtained profit at this moment.

What to do

Implement `__init__(self, bids, asks)`, `trade(self)`, `tradeall(self)`, and `getprofit(self)` methods of `Broker` class in `stockbroker.py` file. Let asks and bids be of n_a and n_b size, respectively. Then, the time complexity of your `__init__`, `trade`, `tradeall`, and `getprofit` methods should be $O(n_a + n_b)$, $O(\log n_a + \log n_b)$, $O(n_a \log n_a + n_b \log n_b)$, and $O(1)$, respectively. submit your `priorityqueue.py` and `stockbroker.py` files to Mimir.