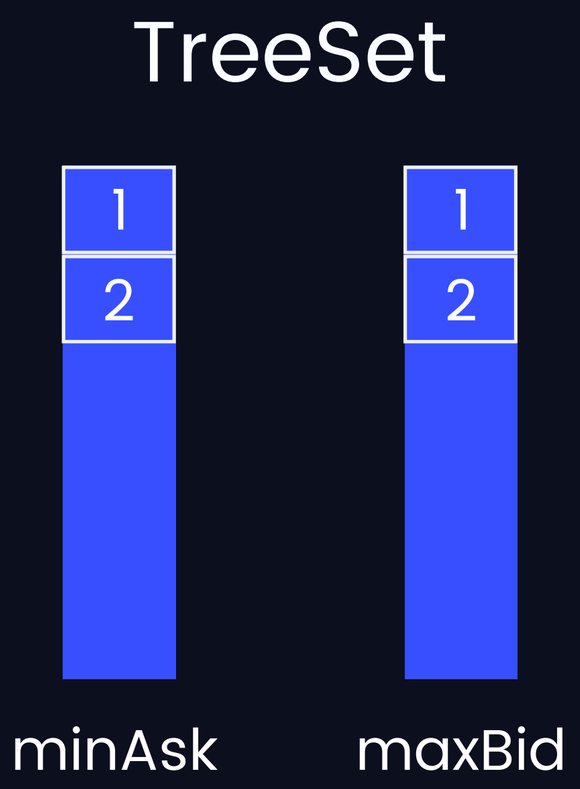
CODE TO CONNECT 2019

F1 - Arbitrage

MATTHEW, ROBERT, YI QIONG

ARBITRAGE OPPORTUNITIES

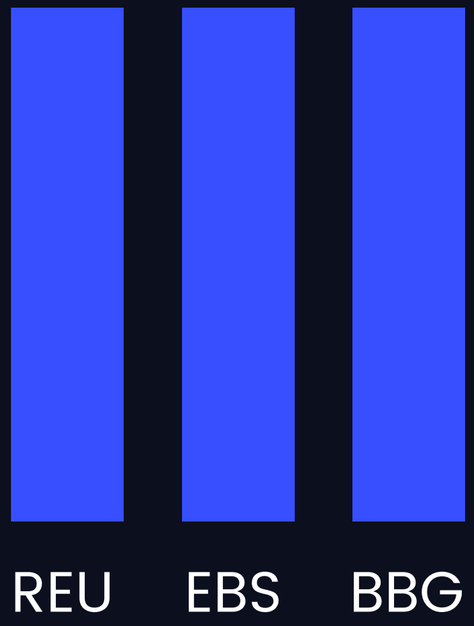


- All providers included in same TreeSet
- Identify as many in-order pairs of (minAsk, maxBid) as possible as long as profit > 0

O (log N)

TIMER

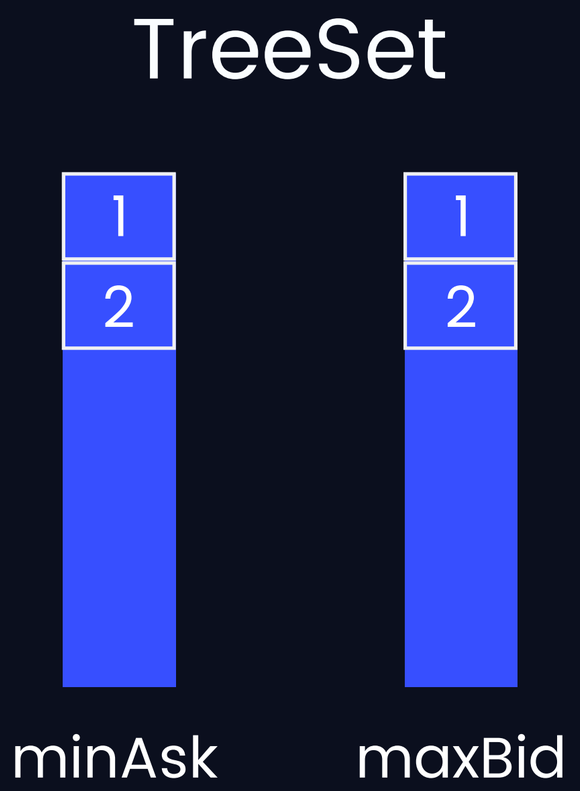
LinkedList



- Remove first element in one second intervals from LinkedList
- Find and remove the element from the TreeSets

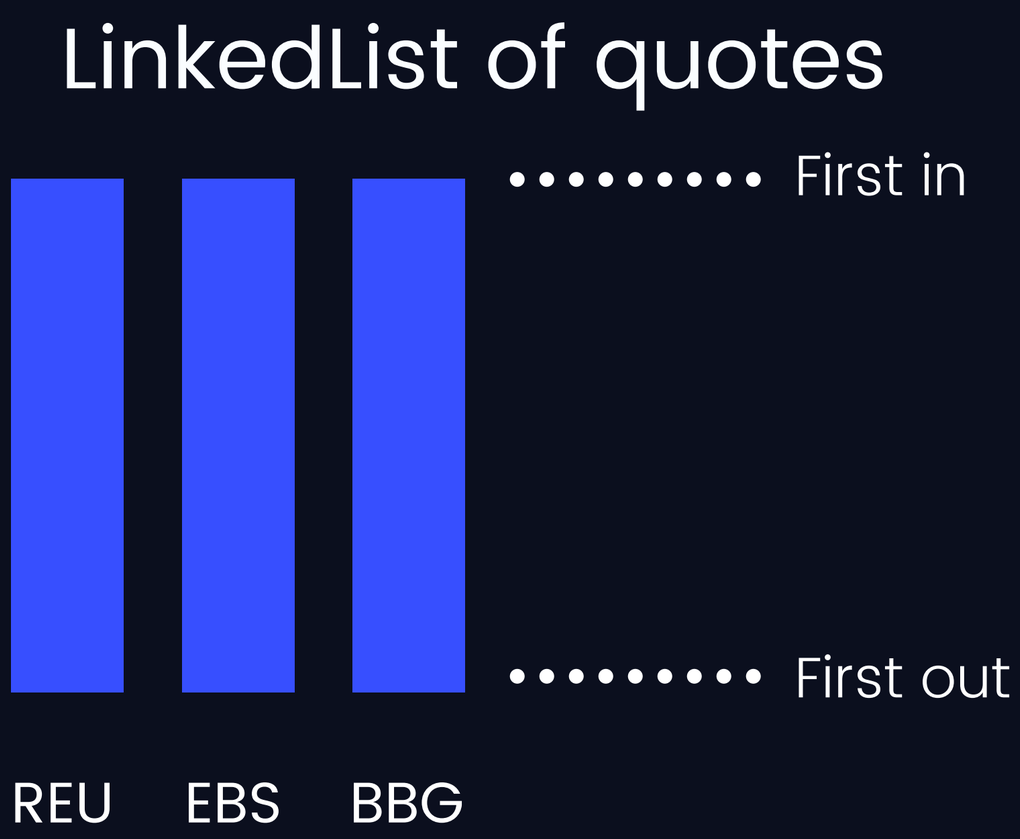
O (1) + O (log N)

EXECUTE ORDER?



- If the pair of (minAsk, maxBid) is not the current first element in the timer LinkedList (not expiring in the next second), then we **do not** execute the order.
- Execute only if some element in the pair is expiring the next second.

INITIAL IDEA



- Remove first quote from LinkedLists in one second intervals
- Traverse through the LinkedLists to find and pair minAsk and maxBid

Issues

- Works for current scale as the LinkedList sizes are all <5 but highly unscalable

O (N)^2

CHALLENGES FACED

- Efficiency of algorithm (scalability)
- Network architecture using web socket
- Synchronising data flow between execution engine and liquidity provider