Homework Assignments for

SORT

This document is provided to SJTU for implementing a Smart Order RouTer.

Contents

[Course Requirement: 2](#_Toc482197800)

[Minimum Features Requirement: 2](#_Toc482197801)

[Extra Features Requirement: 2](#_Toc482197802)

[Homework 1 Basics: 3](#_Toc482197803)

[Function requirements: 3](#_Toc482197804)

[Homework 2 Aggressive Take: 4](#_Toc482197805)

[Function requirements: 4](#_Toc482197806)

[Homework 3 Passive Placement: 5](#_Toc482197807)

[Function requirements: 5](#_Toc482197808)

[Homework 4 Performance : 6](#_Toc482197809)

[Function requirements: 6](#_Toc482197810)

# Course Requirement:

Minimum Features Requirement:

Homework 1 - 3: With those three completed, it can be called an SORT.

## Extra Features Requirement:

Homework 4: Best to have, a SORT with bad performance is useless.

# Homework 1 Basics:

## Function requirements:

Read the given order csv file and construct orders from them.

* Make the profile interactive and support read orders from console.

Read the given quote csv file and construct quotes from them.

,

# Homework 2 Aggressive Take:

## Function requirements:

Implement the Aggressive model

* When a new quote [each line is a quote] is read, check all the current orders one by one if they are marketable. If yes, consider the order has got a fill from this quote, record the time, fill quantity and fill price.
  + For example, for an existing order **BUY 1000 AAPL@150**
  + For quote **NYSE Ask AAPL 100@149.5**, the order will book a fill of [100@149.5](mailto:100@149.5) from NYSE.
  + For quote NASDAQ Ask AAPL 100@151, the order will ignore it.
* Print order fill report [the quantity and average price on each exchange] and the total quantity and average price for each order after all quotes are processed.

# Homework 3 Passive Placement:

## Function requirements:

Implement the passive placement model

* When a passive quote [no order would cross with it] is read, use it to construct the level1 book.
  + Level1 book is empty at the beginning, and there’s an existing order **BUY 1000 AAPL@150**
  + Step1, For quote **NYSE Ask AAPL** [**100@150.5**](mailto:100@150.5); it will go to the level1 ask book:

|  |  |  |
| --- | --- | --- |
| NYSE | NASDAQ | IEX |
| [**100@150.5**](mailto:100@150.5) |  |  |

* + Step2, quote **NYSE Ask AAPL** [**100@149.5**](mailto:100@149.5) because existing order will cross with it, it’s ignored in level1 book.
  + Step3, quote **NYSE Ask AAPL** [**100@151.5**](mailto:100@151.5), it will update the level1 ask book:

|  |  |  |
| --- | --- | --- |
| NYSE | NASDAQ | IEX |
| [**100@151.5**](mailto:100@151.5) |  |  |

* + Step4, quote **IEX Ask AAPL** [**300@150.8**](mailto:300@150.8)**,** it will update the level1 ask book:

|  |  |  |
| --- | --- | --- |
| NYSE | NASDAQ | IEX |
| [**100@151.5**](mailto:100@151.5) |  | [**300@150.8**](mailto:300@150.8) |

* + Step5, quote **NASDAQ Ask AAPL 1200@150**, it will fill the order and update the level1 as book:

|  |  |  |
| --- | --- | --- |
| NYSE | NASDAQ | IEX |
| [**100@151.5**](mailto:100@151.5) | **300@150** | [300@150.8](mailto:300@150.8) |

* Based on the level1 book, decide for each order, how to split the quantity among exchanges.
  + Based on previous example, after step1, the order will be fully (1000@150) on NYSE
  + After step2, the order will be 900@150 on NYSE (because 100 was filled)
  + After step3, the order will be 900@150 on NYSE
  + After step4, the order will be 300@150 on NYSE and 600@150 on IEX
  + After step5, the order will be already totally filled. But if there’s another order, its quantity will be split into three exchanges with ratio 1:3:2 (目前售货量)
* Print the status of passive placement of each step to a file.

# Homework 4 Performance :

## Function requirements:

Performance tuning

* Move the quote processing logic into a dedicated thread;
* Add a command to display the order status from main thread;
* Add a command to display the level1 book from main thread;
* Keep adding symbols and orders into the system; measure the latency of processing each quote update precisely;
* Improve the performance;

**刘未鹏**

**Too big too fail**