

交易系统开发测试题

本测试包含两道题目，你有一周的时间来完成这些的题目。

你可以使用任何网络资源，但需要独立完成这些题目，不得向他人求助。

请不要将题目，题目中的数据或者你的答案分享给任何人。

完成以后请花一点时间对题目进行总结，阐述你对题目的看法和你对自己的解决方案的思路和想法。

题目一：行情转发

将交易所发布的交易行情以网络方式发送给下游程序。请实现数据发送和接受的代码。

说明

行情数据是指交易所按照一定频率发送的反映了当前交易所内部交易情况的一个快照。

其中包含了行情价格信息和买卖量的信息。比如

Side（买卖标志）	Price（价格）	Volume（量）
Sell	11.7	20
Sell	11.0	101
Sell	10.2	200
Sell	10.1	62
Sell	10	85
buy	9.9	77
buy	9.8	76
buy	9.5	90
buy	9.3	20
buy	8	1

其中每一行标识了在该价格上有多少量的买或者卖的股票在等待交易。该行情仅包含买方和卖方最优的五档。而且按照价格从高到底排列。

随着交易的进行，交易所发布的行情会随之发生变化。比如在上面的情况中，下一次发布的行情可能变为

Side (买卖标志)	Price (价格)	Volume (量)
Sell	11.8	33
Sell	11.7	20
Sell	11	101
Sell	10.2	200
Sell	10.1	40
buy	10	15
buy	9.9	77
buy	9.8	76
buy	9.5	90
buy	9.3	20

其中有人在100元的价格上买入了100手，于是与之前卖方的85手成交，并在100元上面留下了剩余的15手买入。并且在101元的价格上，有人撤销了22手，使原来的62手变化为40手。又由于价格的整体升高，导致买方80元价格挡位移出了前五档，而卖方118元价位变得可见。

请根据行情的特性设计传输的协议，将从交易所收到的行情以最快的方式发送给下游的程序。设计并实现发送以及接收的程序。

程序输入

全深度行情列表文件 snap.csv（见附件）。其中每一行数据代表一次行情。

字段说明：

- TIME：当前深度行情对应的时间戳
- BidPrice1~5：第1~5档买方价格
- BidVol1~5：第1~5档购买量
- AskPrice1~5：第1~5档卖方价格
- AskVol1~5：第1~5档卖出量

其中某一价格字段为空时表示本次行情不足五档，该档行情为空。

程序输出

接收程序能够正确接收并输出完整的行情，复原snap.csv中的内容。

要求

1. 逐条发送深度行情（csv文件的每一行），要求每一次传输从发送到接收的延迟最短。延迟时间包括从单条行情被读入到内存（不包含）以后开始，数据协议化处理，数据序列化，网络传输，数据反序列化，数据协议化还原处理的过程。不包含文件读取和写入时间。
2. 为保证低网络延迟，数据传输的总量要控制在5MB以内，优化目标为2MB以内。单次行情传输，不计入网络传输时间的情况下，要求均值至少控制在5微秒以内。
3. 本题目不考察socket编程。网络传输部分可以用变量传递代替，但是传递的内容必须是序列化以后的纯数据（如char数组），不能传递对象或其他内存结构。
4. 允许使用开源方案实现数据传输的序列化和反序列化。
5. 考虑实际情况中，行情深度可以达到上百层，进行程序设计。
6. 考虑每一次深度行情发布之间有较大时间间隔，本题目考察分离的单次行情的发布延迟，并非文件传输速度。

可选功能

1. 实现传输过程中的断线重连。重连以后接收最新发布的数据，丢失数据不需要重传。
 2. 实现多客户端接入
-

题目二：自成交保护模块

在交易当中，当自己的委托单与另一个自己的委托单成交时，我们称之为自成交。自成交往往被交易所和监管机构禁止。在实际的交易过程中，如果发现当前的下单有可能与之前尚未成交的委托单成交时，我们要先撤销之前的委托单，然后才能将最新的委托单发出。侦测并自动撤销可能自成交的程序模块称为自成交保护模块。请编程实现这一功能。

说明

交易过程中会收到如下信息

委托单：向交易所发出的买卖请求。信息包括：

- 方向：买或者卖
- 价格：委托单价格
- 委托量：想要交易的手数
- 订单号：全局唯一号码，用于标识该订单

撤单请求：向交易所发出的撤销委托单的请求。信息包括：

- 撤销单号：与委托单的订单号对应

成交回报：在委托单成交时交易所发回的成交信息。信息包括：

- 成交单号：与委托单的订单号对应
- 成交价格
- 成交量

当买单价格大于或等于卖单价格时，交易就有可能发生。

当收到成交回报显示的成交量小于委托单的委托量时，委托单剩余部分会继续留在市场上等待成交。

当成交回报的累积成交量显示委托单已经全部成交，那么委托单完成，不再继续成交。

为保证不发生自成交，需要跟踪自己所发委托单在交易所的状态，并计算可能自成交的委托单。并在发出新委托单之前发出撤单请求。

示例

测试用例：

	输入	输出
委托单1：	方向-买 价格-100 委托量-20 订单号-1	保护模块：发出1号订单
委托单2：	方向-卖 价格-101 委托量-30 订单号-2	保护模块：发出2号订单
委托单3：	方向-买 价格-99 委托量-10 订单号-3	保护模块：发出3号订
撤单请求1：	订单号-1	保护模块：撤销1号订单
成交回报：	订单号-3 价格-99 成交量-5	保护模块：（内部处理）
委托单4：	方向-买 价格-99 委托量-20 订单号-4	保护模块：发出4号订单
委托单5：	方向-卖 价格-98 委托量-40 订单号-5	保护模块：撤销3号订单，撤销4号订单，发出5号订单

程序接口

solution.cpp（见附件）

要求

- solution.cpp需要使用到gtest和gmock，学习[googletest](#)搭建编译运行环境。
- 请实现源码中“*class MySolution*”部分的函数，并修复测试用例“ExampleCase”
- 设计更多的测试用例
- 考虑实际情况中可能有上千个买卖委托单并分散在几十个不同价格上，针对此情形对指定函数进行优化。