ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH

Bài tập lớn 2

THỊ TRƯỜNG GIAO DỊCH NGOẠI HỐI
(Phần 2)

TP. HỒ CHÍ MINH, THÁNG 05/2020

ĐẶC TẢ BÀI TẬP LỚN 2

Phiên bản 1.1

1 Chuẩn đầu ra

Sau khi hoàn thành bài tập lớn này, sinh viên sẽ có khả năng:

- Hiện thực cấu trúc dữ liệu cây, cụ thể là cây nhị phân
- Thực hiện các thao tác cơ bản trên kiểu dữ liệu cây, cụ thể là cây nhị phân

2 Nhiệm vụ

Sinh viên được yêu cầu xây dựng một chương trình trên C++ đọc vào các thông tin về thị trường giao dịch tiền tệ [1] (phần 4), thực hiện các yêu cầu xử lý đơn giản trên các thông tin này và ghi kết quả vào một tập tin văn bản.
Để thực hiện nhiệm vụ này, sinh viên cần phải tải về tập tin assignment2.zip, trong đó có chứa các tập tin:

- src/main.h, src/main.cpp: Thực hiện các điều khiển chính: đọc tập tin văn bản chứa thông tin về thị trường giao dịch tiền tệ, gọi các lệnh xử lý và ghi kết quả trả về vào tập tin kết quả. Sinh viên được phép đọc các tập tin này nhưng không được phép thay đổi bất cứ nội dung nào của các tập tin này
- src/processData.h, src/processData.cpp: Thực hiện xử lý thông tin về giao dịch tiền tệ. Sinh viên thực hiện nhiệm vụ thông qua các tập tin này. Tuy nhiên, sinh viên KHÔNG được phép thêm các #include nào khác với các #include hiện có trong các tập tin này.
- test/input/1.txt, test/input/2.txt: Các tập tin chứa thông tin về giao dịch tiền tệ mẫu, đầu vào của chương trình. Mỗi tập tin chứa nhiều dòng, mỗi dòng là một lệnh xử lý thông tin giao dịch tiền tệ. Yêu cầu xử lý của mỗi lệnh được trình bày chi tiết trong mục 3.
- test/output/1.txt, test/output/2.txt: Các tập tin chứa kết quả xử lý thông tin giao dịch tiền tệ tương ứng với các tập tin đầu vào tương ứng.

Sau khi giải nén, sinh viên phải dịch các mã C++ được cung cấp, sau đó thực thi bằng cách gõ vào dòng lệnh sau trên cửa sổ lệnh (Command Prompt/Terminal):
main test/input/1.txt test/output/0.txt

Sau khi thực thi xong lệnh trên, sinh viên kiểm tra lại bằng cách so sánh nội dung của tập tin **test/output/0.txt** và **test/output/1.txt** bằng cách sử dụng lệnh sau:

- Windows: *FC test/output/0.txt test/output/1.txt*
- Linux/MacOS:*diff test/output/0.txt test/output/1.txt*

Kết quả so sánh phải chỉ ra hai tập tin này không khác nhau.
Sinh viên thực hiện nhiệm vụ bài tập lớn 2 bằng việc hiệu chỉnh các tập tin processData.h và processData.cpp để tổ chức lưu trữ dữ liệu và xử lý thông tin theo yêu cầu, kiểm tra chương trình bằng việc tạo các tập tin đầu vào mới trên thư mục input và tập tin kết quả tương ứng trên thư mục output.

3 Lệnh xử lý dữ liệu

3.1 Hướng dẫn chung

Mỗi lệnh xử lý dữ liệu là một dòng trên tập tin thông tin giao dịch tiền tệ. Mỗi lệnh bắt đầu bằng một từ khoá (từ in đậm trong mô tả) và theo sau là các thông số (từ đặt trong dấu < và > trong mô tả). Giữa lệnh và các thông số cách nhau đúng một khoảng trắng. Không có khoảng trắng nào trước từ khoá lệnh và cũng không có khoảng trắng nào đi sau thông số cuối cùng. Một số thông số cuối có thể tuỳ chọn (có hoặc không có trong lệnh), các thông số này sẽ được đặt trong dấu [và] trong mô tả. Khi một lệnh không cung cấp đúng số thông số hoặc không có kiểu thông só đúng như mô tả hoặc có các khoảng trắng không như mô tả thì lệnh sẽ không đươc xử lý và kết quả trả về là -1. Ngược lại, lệnh sẽ được xử lý và một giá trị nguyên ≥ 0 sẽ được trả về theo mô tả trong mục 3.2.
Ý nghĩa của từ viết tắt và kiểu dữ liệu của các thông số được mô tả như sau:

- TIME: một số nguyên (biểu diễn thời gian theo chuẩn ISO) thể hiện thời điểm giao dịch.
- BP: (bid price) một số thực thể hiện giá cháo mua.
- AP: (ask price) một số thực thể hiện giá chào bán.
- BC: (base currency) một mã (dạng chuỗi) thể hiện mã tiền tệ mua.
- QC: (quote currecy) một mã (dạng chuỗi) thể hiện mã tiền tệ bán.
- LOT: (lot) một số thực thể hiện giá trị giao dịch được tính bằng lot.
- LV: (leverage) đòn bẩy tài chính. Với định dạng đòn bẩy 1:X thì LV = X.
- MN: (money) số tiền được tính bằng USD.
- ID: một số nguyên thể hiện định danh của giao dịch.
- INST: một chuỗi thể hiện từ khoá tên lệnh.

Trong các lệnh có hai thông số thời gian <TIME_A> và <TIME_B> tuỳ chọn (biểu diễn [<TIME_A> [<TIME_B>]]), chương trình sẽ thực thi lệnh với các thông số khác và

- <TIME_A>≤ <TIME>≤ <TIME_B> nếu có cả <TIME_A> và <TIME_B> trong lệnh.
- <TIME> = <TIME_A> nếu chỉ có <TIME_A> trong lệnh.
- mọi <TIME> nếu không có thông số <TIME_A> và <TIME_B>.

Mỗi lệnh cần phải được xử lý với độ phức tạp thời gian không vượt quá qui định ở cột "Độ phức tạp" khi N $\geq$ 100, với N là số nến trung bình cho mỗi cặp giao dịch tiền tệ.

Sau khi xử lý xong tất cả các lệnh trong tập tin đầu vào và ghi kết quả vào tập tin đích, chương trình phải đảm bảo huỷ tất cả các đối tượng dữ liệu được cấp phát động, không để lại rác trong bộ nhớ trước khi kết thúc chương trình.

3.2 Danh sách lệnh

Yêu cầu	**Độ phức tạp**	**Mô tả**
SD <MN>	O(1)	Thiết lập hoặc thay đổi số tiền ký quỹ. Trong trường hợp thay đổi số tiền ký quỹ, cần tuân theo các ràng buộc ở mục 3.3. Nếu lệnh thực hiện thành công, trả về 1, ngược lại trả về 0.
CD	O(1)	Kiểm tra tài khoản ký quỹ và trả về số tiền tương ứng còn lại trong tài khoản ký quỹ.
SL <LV>	O(1)	Thiết lập hoặc thay đổi đòn bẩy tài chính. Trong trường hợp thay đổi đòn bẩy tài chính, cần tuân theo các ràng buộc ở mục 3.3. Nếu lệnh thực hiện thành công, trả về số tiền mà người dùng hiện tại có thể giao dịch, ngược lại trả về 0.

INS <BC> <QC> <TIME> <BP> <AP>	O(log(N))	Thêm tỷ giá mua/bán của một cặp tiền tệ tại một thời điểm vào cấu trúc dữ liệu cây được đề xuất ở mục 4. Nếu đã tồn tại tỷ giá tại thời điểm vừa thêm trong cấu trúc dữ liệu, ta sẽ cập nhật bằng dữ liệu mới. Nếu lệnh được thực hiện thành công, trả về giá trị TIME của node gốc trong cây cấu trúc dữ liệu đang lưu trữ tỷ giá của cặp <BC>/<QC>.
DEL <BC> <QC> [<TIME_A> [<TIME_B>]]	O(log(N))	Xoá các dữ liệu tỷ giá có các giá trị <BC> và <QC> tương ứng với các thông số <BC>, <QC> của lệnh theo thời gian được cung cấp bởi <TIME_A> và <TIME_B>. Lệnh trả về giá trị TIME của node gốc trong cây cấu trúc dữ liệu đang lưu trữ tỷ giá của cặp <BC>/<QC>.
UPD <BC> <QC> <TIME> <BP> <AP>	O(log(N))	Thay đổi các giá trị <BP>, <AP> của tỷ giá có các giá trị <BC>, <QC>, <TIME> tương ứng với thông số. Nếu không có dữ liệu tỷ giá nào khớp với các thông số <BC>, <QC>, <TIME>, chương trình sẽ trả về 0, ngược lại, chương trình sẽ thay đổi các giá trị <BP> <AP> của tỷ giá tương ứng trong cấu trúc dữ liệu và trả về giá trị TIME của node gốc trong cây cấu trúc dữ liệu đang lưu trữ tỷ giá của cặp <BC>/<QC>.
OB <BC> <QC> <TIME> <LOT> <ID> hoặc **OS** <BC> <QC> <TIME> <LOT> <ID>	O(log(N))	Mở lệnh mua (OB) hoặc mở lệnh bán (OS) với cặp tiền tệ <BC>/<QC> tại thời điểm <TIME> với số lot giao dịch là <LOT>. <ID> là mã số để định danh mỗi giao dịch. Lệnh mở giao dịch thành công khi ID không trùng với bất kỳ giao dịch nào trước đó và trả về số tiền đã giao dịch tương ứng. Ngược lại, trả về 0.
CB <BC> <QC> <TIME> <ID> hoặc **CS** <BC> <QC> <TIME> <ID>	O(log(N))	Đóng lệnh mua (CB) hoặc đóng lệnh bán (CS) với cặp tiền tệ <BC>/<QC> tại thời điểm <TIME> với <ID> là mã số định danh của giao dịch mở lệnh tương ứng. Lệnh đóng giao dịch thành công khi ID mở lệnh tồn tại và trả về lợi nhuận tương ứng của giao dịch, tính bằng USD. Ngược lại, trả về 0.

3.3 Các ràng buộc

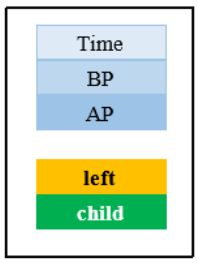
Một số ràng buộc cần thỏa mãn khi xử lý các lệnh trên:

1. Không thể thiết lập hoặc thay đổi đòn bẩy tài chính nếu chưa thiết lập tài khoản ký quỹ.
2. Chỉ có thể tăng số tiền trong tài khoản ký quỹ nếu đang tồn tại các giao dịch (lệnh mở mua/ bán) chưa đóng lệnh.
3. Bất kể khi nào sau khi đóng lệnh, tài khoản ký quỹ bị nhỏ hơn hoặc bằng 0, tất cả các giao dịch (mua/ bán) tiếp theo sẽ không được thực hiện. Tất cả các lệnh còn chưa đóng tại thời điểm đóng buộc phải đóng lại.
4. Tất cả lợi nhuận không được tính bằng USD phải quy đổi thành USD.
5. Trong bài tập lớn này, ta chỉ giao dịch đối với cặp tiền tệ tồn tại USD trong BC hoặc QC.

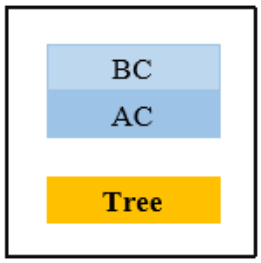
Tất cả các lệnh nếu đi ngược lại với các ràng buộc trên đều trả về kết quả -1.

4 Tổ chức cấu trúc dữ liệu

Để tổ chức dữ liệu cho bài tập lớn, sinh viên yêu cầu phải sử dụng cây nhị phân tự cân bằng AVL để hiện thực như hình vẽ bên dưới (Hình 2). Ta thấy rằng, do ở mỗi lệnh đóng và mở giao dịch, ta cần phải tìm kiếm tỷ giá tại đúng thời điểm đó để ghi nhận nên AVL sẽ là một trong những phương pháp hữu hiệu giúp ta có thể tìm kiếm với độ phức tạp luôn là O(log(N)).

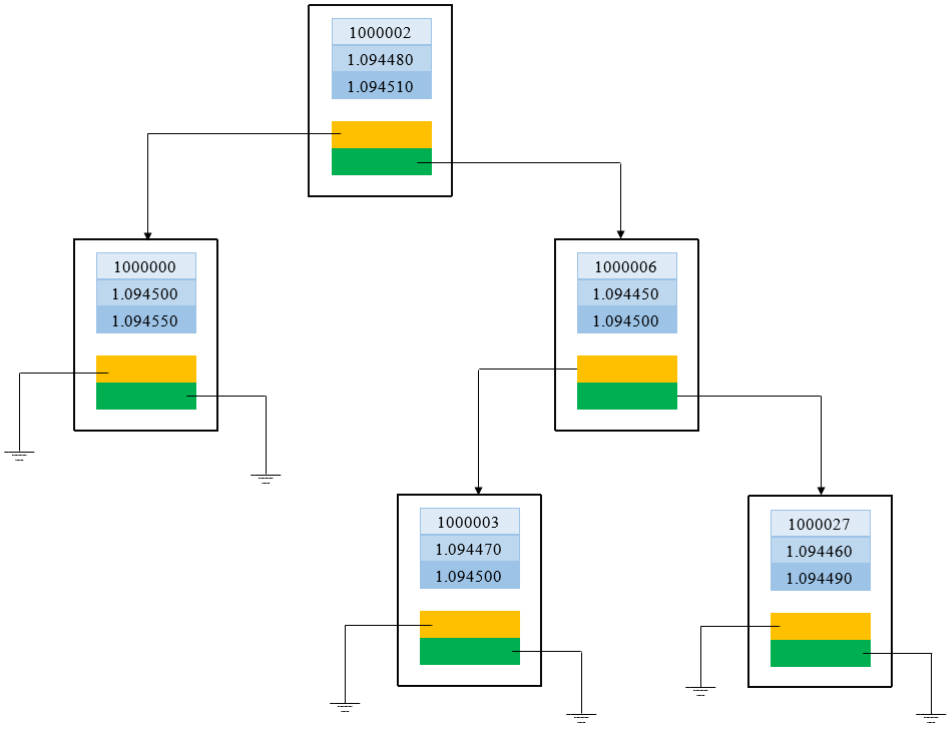


(a) Thông tin của tỷ giá tại một thời điểm



(b) Thông tin của một cặp tiền tệ

Hình 1: Các thông tin lưu trữ



Hình 2: Biểu diễn dữ liệu tỷ giá của một cặp tiền tệ

Để hiện thực bài tập lớn này, một danh sách liên kết đơn nên được sử dụng để lưu trữ thông tin các cặp tiền tệ (Hình 1b), trong đó có các thông tin:

- BC
- QC
- con trỏ trỏ tới cây nhị phân tự cân bằng AVL (với giá trị key là giá trị thời gian TIME) để lữu trữ thông tin tỷ giá.

5 Nộp bài

Sinh viên nộp 2 tập tin: **processData.h** và **processData.cpp** tại "Assignment 2 Submission" trong site "Cấu trúc dữ liệu và giải thuật (CO2003)_CC+CQ_(HK192)" của môn học. Các tập tin phải ở dạng nguyên mẫu (KHÔNG NÉN), đúng tên và có thể dịch được thành công khi kết hợp với các tập tin main.h và main.cpp đã được cung cấp. Nhắc lại, trong hai tập tin processData.h và processData.cpp, sinh viên không được thêm vào bất cứ #include nào khác với các #include hiện có trong các tập tin này.

Thời hạn nộp bài được công bố tại nơi nộp bài trong site nêu trên. Đến thời hạn nộp bài, đường liên kết sẽ tự động khoá nên sinh viên sẽ không thể nộp chậm. Để tránh các rủi ro có

thể xảy ra vào thời điểm nộp bài, sinh viên PHẢI nộp bài trước thời hạn quy định ít nhất **một giờ**.

6 Xử lý gian lận

Bài tập lớn phải được sinh viên TỰ LÀM. Sinh viên sẽ bị coi là gian lận nếu:

- Có sự giống nhau bất thường giữa mã nguồn của các bài nộp. Trong trường hợp này, TẤT CẢ các bài nộp đều bị coi là gian lận. Do vậy sinh viên phải bảo vệ mã nguồn bài tập lớn của mình.
- Sinh viên không hiểu mã nguồn do chính mình viết, trừ những phần mã được cung cấp sẵn trong chương trình khởi tạo. Sinh viên có thể tham khảo từ bất kỳ nguồn tài liệu nào, tuy nhiên phải đảm bảo rằng mình hiểu rõ ý nghĩa của tất cả những dòng lệnh mà mình viết. Trong trường hợp không hiểu rõ mã nguồn của nơi mình tham khảo, sinh viên được đặc biệt cảnh báo là KHÔNG ĐƯỢC sử dụng mã nguồn này; thay vào đó nên sử dụng những gì đã được học để viết chương trình.
- Nộp nhầm bài của sinh viên khác trên tài khoản cá nhân của mình.

Trong trường hợp bị kết luận là gian lận, sinh viên sẽ bị điểm 0 cho toàn bộ môn học (không chỉ bài tập lớn).

KHÔNG CHẤP NHẬN BẤT KỲ GIẢI THÍCH NÀO VÀ KHÔNG CÓ BẤT KỲ NGOẠI LỆ NÀO!

Sau mỗi bài tập lớn được nộp, sẽ có một số sinh viên được gọi phỏng vấn ngẫu nhiên để chứng minh rằng bài tập lớn vừa được nộp là do chính mình làm.

7 Thay đổi so với phiên bản trước

Tài liệu

[1] Trần Ngọc Bảo Duy, Tài liệu phục vụ bài tập lớn môn Cấu trúc dữ liệu và Giải thuật, 4/2020.

——————————HẾT——————————