

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 1

**THỊ TRƯỜNG
GIAO DỊCH NGOẠI HỐI
(Phần 1)**

TP. HỒ CHÍ MINH, THÁNG 05/2019

Đặc Tả Bài Tập Lớn 1

Phiên bản 1.3

1 Chuẩn đầu ra

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

- Hiện thực cấu trúc dữ liệu danh sách
- Thực hiện các thao tác cơ bản trên kiểu dữ liệu danh sách

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], 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 assignment1.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 1 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 tùy 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 số đúng như mô tả hoặc có các khoảng trắng không như mô tả thì lệnh sẽ không được 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 mở cửa của phiên giao dịch.
- **OP**: (opening price) một số thực thể hiện giá mở cửa.
- **HP**: (high price) một số thực thể hiện giá cao nhất.
- **LP**: (low price) một số thực thể hiện giá thấp nhất.
- **CP**: (close price) một số thực thể hiện giá đóng cửa.
- **BC**: (Base Currency) một mã (dạng chuỗi) thể hiện mã tiền tệ mua.
- **QC**: (Quote Currency) một mã (dạng chuỗi) thể hiện mã tiền tệ bán.
- **CODE**: một chuỗi bắt đầu bằng chữ `C` theo sau là một con số thể hiện yêu cầu cụ thể của lệnh.

- INST: một chuỗi thể hiện từ khoá tên lệnh.

Trong các lệnh có hai thông số thời gian $\langle \text{TIME_A} \rangle$ và $\langle \text{TIME_B} \rangle$ tùy chọn (biểu diễn $[\langle \text{TIME_A} \rangle [\langle \text{TIME_B} \rangle]]$), chương trình sẽ thực thi lệnh với các thông số khác và

- $\langle \text{TIME_A} \rangle \leq \langle \text{TIME} \rangle \leq \langle \text{TIME_B} \rangle$ nếu có cả $\langle \text{TIME_A} \rangle$ và $\langle \text{TIME_B} \rangle$ trong lệnh.
- $\langle \text{TIME} \rangle = \langle \text{TIME_A} \rangle$ nếu chỉ có $\langle \text{TIME_A} \rangle$ trong lệnh.
- mọi $\langle \text{TIME} \rangle$ nếu không có thông số $\langle \text{TIME_A} \rangle$ và $\langle \text{TIME_B} \rangle$.

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 nền 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

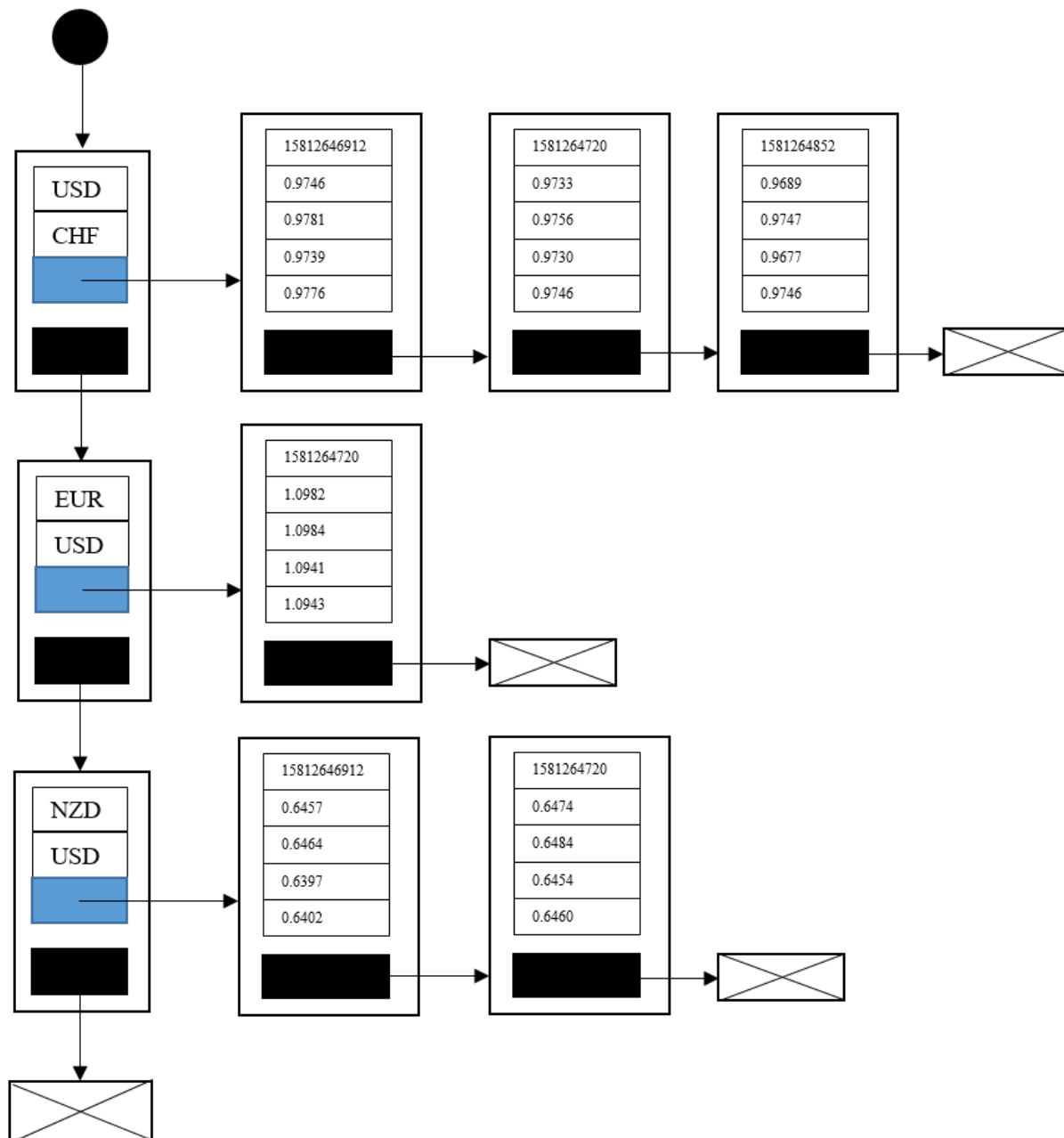
Yêu cầu	Độ phức tạp	Mô tả
INS $\langle \text{BC} \rangle$ $\langle \text{QC} \rangle$ $\langle \text{TIME} \rangle$ $\langle \text{OP} \rangle$ $\langle \text{HP} \rangle$ $\langle \text{LP} \rangle$ $\langle \text{CP} \rangle$	$O(N)$	Thêm dữ liệu nền vào cấu trúc dữ liệu và trả về số lượng nền sau khi thêm. Dữ liệu nền có thể thêm một cách ngẫu nhiên theo thời gian, nếu thời gian trùng nhau, dữ liệu nền đến sau không được thêm vào cấu trúc dữ liệu.
DEL $\langle \text{BC} \rangle$ $\langle \text{QC} \rangle$ $[\langle \text{TIME_A} \rangle$ $[\langle \text{TIME_B} \rangle]]$	$O(N)$	Xoá các dữ liệu nền có các giá trị $\langle \text{BC} \rangle$ và $\langle \text{QC} \rangle$ tương ứng với các thông số $\langle \text{BC} \rangle$, $\langle \text{QC} \rangle$ của lệnh theo thời gian được cung cấp bởi $\langle \text{TIME_A} \rangle$ và $\langle \text{TIME_B} \rangle$. Lệnh trả về số dữ liệu nền bị xoá.
UPD $\langle \text{BC} \rangle$ $\langle \text{QC} \rangle$ $\langle \text{TIME} \rangle$ $\langle \text{OP} \rangle$ $\langle \text{HP} \rangle$ $\langle \text{LP} \rangle$ $\langle \text{CP} \rangle$	$O(N)$	Thay đổi các giá trị $\langle \text{OP} \rangle$, $\langle \text{HP} \rangle$, $\langle \text{LP} \rangle$, $\langle \text{CP} \rangle$ của nền có các giá trị $\langle \text{BC} \rangle$, $\langle \text{QC} \rangle$, $\langle \text{TIME} \rangle$ tương ứng với thông số. Nếu không có dữ liệu nền nào khớp với các thông số $\langle \text{BC} \rangle$, $\langle \text{QC} \rangle$, $\langle \text{TIME} \rangle$, chương trình sẽ trả về 0, ngược lại, chương trình sẽ thay đổi các giá trị $\langle \text{OP} \rangle$, $\langle \text{HP} \rangle$, $\langle \text{LP} \rangle$, $\langle \text{CP} \rangle$ của nền tương ứng trong cấu trúc dữ liệu và trả về số nền được thay đổi.

ST <BC> <QC> [<TIME_A> [<TIME_B>]]	O(N)	Đếm và trả về số lượng nến của cặp giao dịch tiền tệ <BC> <QC> có đỉnh xoay tròn [1] trong đoạn thời gian từ TIME_A đến TIME_B.
MB <BC> <QC> [<TIME_A> [<TIME_B>]]	O(N)	Đếm và trả về số lượng nến Marubozu [1] của cặp giao dịch tiền tệ <BC> <QC> trong đoạn thời gian từ TIME_A đến TIME_B.
DJ <BC> <QC> [<CODE>] [<TIME_A> [<TIME_B>]]	O(N)	<p>Đếm và trả về số lượng nến Doji [1] của cặp giao dịch tiền tệ <BC> <QC> trong đoạn thời gian từ TIME_A đến TIME_B. Nếu có thông số <CODE> (là một chuỗi bắt đầu bằng chữ C), chương trình sẽ đếm cụ thể loại nến theo giá trị của <CODE> như sau:</p> <ul style="list-style-type: none"> • CODE = C1, đếm nến Doji bóng dài. • CODE = C2, đếm nến Doji chuồn chuồn. • CODE = C3, đếm nến Doji bia mộ. • CODE = C4, đếm nến Doji bốn giá. <p>Nếu không có thông số <CODE>, chương trình sẽ đếm tất cả các loại nến Doji.</p>
EG <BC> <QC> [<CODE>] [<TIME_A> [<TIME_B>]]	O(N)	<p>Đếm và trả về số lượng cụm nến Nhấn chìm [1] của cặp giao dịch tiền tệ <BC> <QC> trong đoạn thời gian từ TIME_A đến TIME_B. Xử lý thông số <CODE> tương tự như với lệnh DJ với mã <CODE> cho cụm nến Nhấn chìm như sau:</p> <ul style="list-style-type: none"> • CODE = C1, đếm cụm nến Nhấn chìm tăng. • CODE = C2, đếm cụm nến Nhấn chìm giảm. <p>Nếu không có thông số <CODE>, chương trình sẽ đếm tất cả các loại cụm Nến nhấn chìm.</p>

Ghi chú: Trong bài tập lớn này, ta giả định độ dài của thân nến hoặc bóng nến được xem là nhỏ nếu khoảng chênh lệch xem xét không quá 0.5 PIP, ngược lại được xem là lớn.

4 Gợi ý tổ chức cấu trúc dữ liệu

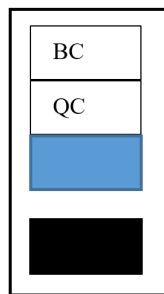
Sinh viên có thể chọn tổ chức dữ liệu theo gợi ý như sau để hiện thực bài tập lớn 1.



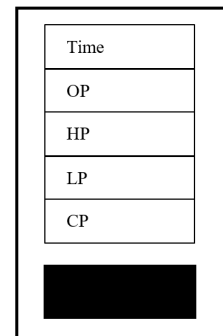
Hình 1: Biểu diễn dữ liệu biểu đồ nén của các cặp tiền tệ

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, trong đó có các thông tin:

- BC
- QC
- link tới một danh sách liên kết đơn có thứ tự theo thời gian của các nền ứng với BC và QC



(a) Thông tin của một cặp tiền tệ



(b) Thông tin của một nền

Hình 2: Các thông tin lưu trữ trong danh sách liên kết

5 Nộp bài

Sinh viên nộp 2 tập tin: **processData.h** và **processData.cpp** tại "Assignment 1 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

- Thay đổi về phần ghi chú dưới bảng giải thích các lệnh, khoảng chệch lệnh thay đổi từ 5 PIP thành 0.5 PIP.

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—————