

CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT II BÀI TẬP LỚN 1

HỆ THỐNG QUẢN LÝ SẢN XUẤT KINH DOANH

v 3.0

1. Giới thiệu

Một công ty chuyên sản xuất và cung ứng các mặt hàng ra thị trường. Công ty cần phải quản lý lượng sản xuất được, đơn đặt hàng từ khách hàng cũng như các tính toán thống kê cần thiết. Giả định rằng, chỉ có mã các sản phẩm là được lưu vào trong hệ thống. Ở mỗi ca trực, công ty cần ghi nhận lại toàn bộ quá trình sản xuất, kinh doanh.

2. Yêu cầu

Trong bài tập lớn này, sinh viên sẽ được cung cấp một file chứa dữ liệu nhập, bao gồm thông tin về các sự kiện đến hệ thống. Một sự kiện có thể là một lệnh nhập kho về một sản phẩm vừa mới được sản xuất ra, hoặc xuất kho cho một đơn đặt hàng về một sản phẩm của công ty, hoặc là một yêu cầu tính toán thống kê ... Sau khi kết thúc công việc, chương trình sẽ xuất ra màn hình hoặc lưu vào file hiện trạng dữ liệu của hệ thống. Dữ liệu lưu trữ và xuất là cây nhị phân. Chi tiết cụ thể công việc sinh viên phải làm sẽ mô tả trong phần 4.

3. Dữ liệu nhập

Dữ liệu nhập của chương trình được chứa trong file mang tên events.txt. File này sẽ chứa các thông tin về các sự kiện gặp phải trong ca trực. Lưu ý là các số liệu tồn nếu có sẽ được đưa vào thông qua các sự kiện nhập kho. Mỗi sự kiện sẽ được mô tả bằng một giá trị số, gọi là mã sự kiện. Ý nghĩa tương ứng của từng sự kiện được mô tả trong Bảng 1. Số sự kiện là không cố định, có thể thay đổi tùy theo test case, và tối đa là 2^{100} sự kiện. Một sự kiện có thể xảy ra nhiều lần. Các sự kiện có thể trình bày thành nhiều dòng.

Bảng 1 – Các sự kiện xảy ra trong ca trực

Mã sự kiện	Ý nghĩa
0	Ca trực chấm dứt đột ngột
1XXXY	Nhập kho Y sản phẩm XXX
2XXXY	Một số lượng Y của sản phẩm XXX đã được khách hàng đặt hàng
3	Yêu cầu cấu trúc lại sổ kho theo AVL theo cách đơn giản
4	Yêu cầu cấu trúc lại sổ kho theo AVL theo cách phức tạp
5XXXY	Yêu cầu nhập kho đặc biệt
6Z	Yêu cầu loại bỏ các sản phẩm dư thừa

Ví dụ 1: Với dữ liệu nhập là

11113 21112

thì có các sự kiện sau:

Sự kiện 1: nhập kho 3 sản phẩm 111.

Sự kiện 2: có 2 sản phẩm 111 được đặt hàng.

4. Hiện thực

Sinh viên được cung cấp 3 lớp ProductRecord, WarehouseBook và WarehouseNode.

Lớp ProductRecord lưu trữ thông tin về sản phẩm gồm mã sản phẩm (productID) và số lượng có trong kho (quantity). Giá trị của productID nằm trong khoảng [0-999], giá trị của quantity nằm trong khoảng [0-99]. Lớp ProductRecord đã có đầy đủ các thuộc tính và phương thức, sinh viên không được thêm hoặc thay đổi lớp này.

```
public class ProductRecord {  
    private int productID;  
    private int quantity;  
    ...  
}
```

Lớp WarehouseBook là sổ kho sử dụng cấu trúc cây nhị phân để lưu trữ toàn bộ dữ liệu về sản phẩm có trong kho, được định nghĩa như sau

```
public class WarehouseBook  
{  
    private WarehouseNode root;  
    private int size;  
    ...  
}
```

- WarehouseNode root: gốc của cây nhị phân;
- int size: kích thước cây nhị phân (số sản phẩm có trong kho).

Mỗi nút trong cây nhị phân là đối tượng của lớp WarehouseNode gồm

- ProductRecord record: thông tin về sản phẩm
- WarehouseNode left, right: cây con bên trái và cây con bên phải
- int balance: thông tin về mức cân bằng của nút (chiều cao cây con bên phải trừ chiều cao cây con bên trái; lưu ý cây có 1 nút có chiều cao là 1) và chỉ dùng trong trường hợp cần cân bằng lại cây.

```
protected static class WarehouseNode {  
    private ProductRecord record;  
    private WarehouseNode left, right;  
    private int balance;  
    ...  
}
```

Sinh viên sẽ hiện thực các phương sau trong lớp WarehouseBook:

- **public** WarehouseBook(File file): phương thức khởi tạo sổ kho từ thông tin được lưu trữ trong tập tin file. Định dạng lưu trữ được mô tả trong phần 5.
- **public void** save(File file): lưu sổ kho ra tập tin file theo định dạng được mô tả trong phần 5.
- **public void** process(File file): xử lý các sự kiện được lưu trong tập tin file.
- **public void** process(List<String> events): xử lý các sự kiện được truyền vào bằng danh sách các mã sự kiện.
- **public** String toString(): trả về chuỗi ký tự biểu diễn thông tin của sổ kho theo định dạng trong phần 5.

5. Xây dựng cây nhị phân và các định dạng

5.1) Sổ kho được biểu diễn dưới dạng chuỗi ký tự theo quy tắc sau:

- Thông tin về số lượng của mỗi sản phẩm có trong kho được biểu diễn bởi chuỗi ký tự có 5 chữ số bằng cách ghép các chữ số của mã sản phẩm (3 chữ số) và số lượng của sản phẩm (2 chữ số).
- Sổ kho được biểu diễn dưới dạng chuỗi ký tự để lưu vào tập tin hoặc xuất ra màn hình bằng cách sử dụng ngoặc đơn theo định dạng sau

(<thông tin về sản phẩm nút cha> (<thông tin các sản phẩm cây con bên trái> <thông tin các sản phẩm cây con bên phải>))

Ví dụ 2: Nếu có hai sản phẩm trong sổ kho, sản phẩm ở vị trí nút gốc là 3 sản phẩm 111, sản phẩm thứ hai đứng ở vị trí nút con bên phải của ghi chú đầu, là 2 sản phẩm 222; thì NsI của cây là

(11103 (N 22202))

N viết tắt cho NULL, tức nút gốc không có nút con bên trái

5.2) Tập tin lưu trữ các sự kiện cần xử lý được lưu trữ dưới dạng văn bản (text) với các sự kiện được cách nhau bởi dấu cách (khoảng trắng).

5.3) Cây nhị phân sử dụng để biểu diễn sổ kho được xây dựng theo quy tắc sau:

S1: Sổ kho là một cây nhị phân tìm kiếm (BST) với mã sản phẩm là **khóa tìm kiếm**. Nếu sau một sự kiện nào đó, cây nhị phân không còn tồn tại nút nào thì các sự kiện còn lại sẽ không được xử lý.

S2: Nếu gặp sự kiện 0, các sự kiện tiếp theo sau đó sẽ không được xử lý.

S3: Nếu gặp sự kiện 1XXXY, chương trình sẽ tìm trong sổ kho sản phẩm XXX. Nếu tìm thấy sẽ cộng dồn số lượng Y vào số lượng hiện tại của nút tìm thấy. Nếu không tìm thấy, một nút mới với các số liệu tương ứng sẽ được thêm vào (cây) sổ kho theo nguyên tắc của cây BST.

Ví dụ 3: Với chuỗi sự kiện là

17234 19343 12246 17236

Sau 3 sự kiện nhập kho đầu tiên, sổ kho là

(72304 (22406 93403))

Khi gặp sự kiện thứ tư, số lượng tồn của sản phẩm 723 sẽ được cập nhật thành 10. Cây nhị phân kết quả sẽ là

(72310 (22406 93403)).

S4: Khi gặp sự kiện 2XXXY, sản phẩm ABC nào “gần” trùng với XXX nhất sẽ được lựa chọn để giao hàng. Khi đó, số lượng tồn sản phẩm này trong sổ kho sẽ được trừ đi một lượng Y. Nếu sau khi trừ đi, số lượng tồn nhỏ hơn hoặc bằng 0, sản phẩm này sẽ bị loại khỏi sổ kho. Nguyên tắc loại bỏ 1 nút trên cây BST là **thay thế bằng nút nhỏ nhất của cây con bên phải**.

Nút có sản phẩm ABC được xem là gần với XXX nhất trên toàn cây nhị phân nếu |ABC-XXX| là

nhỏ nhất so với các nút khác. Nếu có hai nút có cùng giá trị gần nhất như vậy, sẽ ưu tiên cho nút có mã sản phẩm nhỏ hơn.

Ví dụ 4: Với chuỗi sự kiện là

17234 17243 17259 27242

Khi gặp lệnh đặt hàng, giá trị của sổ kho là

(72304 (N 72403 (N 72509)))

do đó sản phẩm giao hàng có mã là 724 (chính là sản phẩm đặt hàng). Khi đó sổ kho kết quả là

(72304 (N 72401 (N 72509)))

Ví dụ 5: Với chuỗi sự kiện là

17234 17253 27233

Sản phẩm giao hàng sẽ có mã là 723. Khi đó sổ kho kết quả là

(72301 (N 72503))

Ví dụ 6: Với chuỗi sự kiện là

17234 17253 17211 27234

Trước khi gặp lệnh đặt hàng, sổ kho là

(72304 (72101 72503))

Sản phẩm giao hàng sẽ có mã là 723. Do sản phẩm 723 chỉ có 4 đơn vị mà lệnh đặt hàng cũng 4 đơn vị nên sau khi giao hàng xong, sản phẩm này sẽ bị loại khỏi sổ kho. Khi đó, nút cực trái của cây con phải (là 72503) sẽ được đem thay cho nút gốc. Sổ kho kết quả là

(72503 (72101 N))

S5: Khi gặp sự kiện 3, sổ kho sẽ được cấu trúc lại thành một AVL-BST theo cách sau:

- Duyệt sổ kho theo LNR vào một danh sách tạm
- Xóa các phần tử trong sổ kho hiện hành
- Xây dựng lại sổ kho AVL-BST với đầu vào là danh sách tạm nói trên
 - Lấy phần tử “giữa” của danh sách làm nút gốc. Phần tử giữa của 1 đoạn danh sách có chỉ số từ low đến high là $(low+high) \div 2$ (chia nguyên).
 - Xây dựng AVL-BST cho danh sách từ đầu đến trước phần tử giữa (đệ qui), gắn vào nhánh trái
 - Xây dựng AVL-BST cho danh sách từ sau phần tử giữa đến cuối (đệ qui), gắn vào nhánh phải

Sau khi tạo thành AVL-BST, sổ kho vẫn được vận hành như một BST bình thường (**không cần cân bằng lại khi mất cân bằng**).

Ví dụ 7: Với chuỗi sự kiện là

17234 17253 17291 17324 17419 3

Trước khi gặp sự kiện 3, sổ kho là

(72304 (N 72503 (N 72901 (N 73204 (N 74109)))))

Khi gặp sự kiện 3, danh sách được duyệt thành [72304, 72503, 72901, 73204, 74109]. Khi đó cây AVL-BST được xây dựng bằng cách đưa 72901 thành nút gốc, nhánh trái là một AVL-BST xây dựng từ danh sách [72304, 72503], nhánh phải là một AVL-BST xây dựng từ danh sách [73204, 74109]. Sổ kho kết quả là

(72901 (72304 (N 72503) 73204 (N 74109)))

S6: Khi gặp sự kiện 4, sổ kho sẽ được cấu trúc lại thành một AVL-BST theo cách sau:

- Duyệt sổ kho theo NRL vào một danh sách tạm
- Xóa các phần tử khỏi sổ kho hiện hành
- Xây dựng lại sổ kho AVL-BST bằng cách lần lượt thêm 1 phần tử từ danh sách tạm đã nói vào trong AVL-BST và cân bằng nếu cần.

Sau khi tạo thành AVL-BST, sổ kho vẫn được vận hành như một BST bình thường (**không cần cân**

bằng lại khi mất cân bằng).

Ví dụ 8: Với chuỗi sự kiện là

17234 17253 17291 17324 17419 4

Trước khi gặp sự kiện 4, sổ kho là

(72304 (N 72503 (N 72901 (N 73204 (N 74109))))

Khi gặp sự kiện 4, danh sách được duyệt thành [72304, 72503, 72901, 73204, 74109]. Khi đó cây AVL-BST được xây dựng bằng cách lần lượt đưa 72304, 72503, ... vào một AVL-BST rỗng ban đầu:

(72304)

(72304 (N 72503))

(72304 (N 72503 (N 72901))) = right rotate => (72503 (72304 72901))

(72503 (72304 72901 (N 73204)))

(72503 (72304 72901 (N 73204 (N 74109)))) = right rotate => (72503 (72304 73204 (72901 74109)))

Sổ kho kết quả là

(72503 (72304 73204 (72901 74109)))

S7: Khi gặp sự kiện 5XXXXY, sổ kho sẽ được xây dựng lại theo cách sau:

- Duyệt sổ kho theo RLN vào một danh sách tạm
- Xóa các phần tử khỏi sổ kho hiện hành
- Gọi ZZ là lượng tồn sản phẩm XXX có trong danh sách tạm RLN vừa được tạo ra. Loại phần tử XXXZZ ra khỏi danh sách đã nói. Nếu không có sản phẩm XXX trong danh sách, thì ZZ=0.
- Một phần tử mới XXXWW được thêm vào nút gốc của sổ kho. WW=ZZ+Y.
- Lần lượt thêm vào sổ kho các sản phẩm còn lại trong danh sách RLN, từ phần tử đầu tiên đến cuối. Chú ý: cách thức thêm vào như là thêm vào cây BST bình thường.

Ví dụ 9: Với chuỗi sự kiện là

17234 17253 17291 17324 17419 57298

Trước khi gặp sự kiện 57298, sổ kho là

(72304 (N 72503 (N 72901 (N 73204 (N 74109))))

Khi gặp sự kiện 57298, danh sách được duyệt thành [74109, 73204, 72901, 72503, 72304]. Tồn kho của sản phẩm 729 đang là 1 => Tồn kho mới sẽ là 1+8 = 9. Khi đó sổ kho mới được xây dựng :

(7299)

(7299 (N 74109))

(7299 (N 74109 (73204 N)))

(7299 (72503 74109 (73204 N)))

(7299 (72503 (72304 N) 74109 (73204 N)))

S8: Khi gặp sự kiện 6Z, tất cả các sản phẩm ở độ sâu lớn hơn hoặc bằng Z sẽ bị xóa sổ kho.

Độ sâu (depth) của một nút sẽ bằng khoảng cách từ nút đó đến nút gốc cộng thêm 1. Như vậy, độ sâu của nút gốc sẽ là 1.

Ví dụ 9: Với chuỗi sự kiện là

17234 17253 17291 17324 17419 57298 63

Tương tự như ví dụ 10, trước khi gặp sự kiện 63, sổ kho là

(7299 (72503 (72304 N) 74109 (73204 N)))

Khi đó, tất cả các phần tử ở độ sâu lớn hơn hoặc bằng 3 đều bị xóa bỏ. Sổ kho còn lại là

(7299 (72503 74109))

6. Cách dịch và thực thi chương trình

Sinh viên được cung cấp file assignment1.zip gồm các file sau:

- assignment1.pdf - file mô tả Bài tập lớn

- src\vn\edu\tdt\it\dsa\ProductRecord.java - lớp thông tin sản phẩm trong kho
- src\vn\edu\tdt\it\dsa\WarehouseBook.java - lớp sổ kho
- test\warehouse.txt - ví dụ file lưu trữ sổ kho
- test\events.txt - ví dụ file sự kiện

Lưu ý:

- sinh viên viết mã cho các phương thức cần thực hiện được mô tả trong phần 4
- được phép thêm các phương thức riêng hoặc thuộc tính riêng (private method, private field) trong lớp WarehouseBook
- không được phép thay đổi khuôn mẫu (prototype) các phương thức đã được định nghĩa;
- không được phép chỉnh sửa lớp ProductRecord;
- không được phép thay đổi tên package;
- file bài làm phải biên dịch được bằng Java phiên bản 1.8 và không sử dụng bất kỳ thư viện ngoài nào;
- comment phải viết bằng tiếng Anh hoặc tiếng Việt không dấu.

Lệnh biên dịch (trên Linux)

```
javac -d ./bin/ -sourcepath ./src/ ./src/vn/edu/tdt/it/dsa/WarehouseBook.java
```

Lệnh thực thi (trên Linux)

```
java -classpath ./bin/ vn.edu.tdt.it.dsa.WarehouseBook
```

7. Nộp bài

Khi nộp bài, sinh viên sử dụng account đã được cấp phát trên hệ thống Sakai để nộp bài qua mạng. Sinh viên chỉ nộp đúng một file **WarehouseBook.java**. Tất cả các file nộp khác sẽ bị tự động xóa khi chấm bài. File được nộp phải là file chương trình gốc, sinh viên không được nén file khi nộp bài.

Thời hạn để nộp bài sẽ được thông báo trên Sakai. Sinh viên phải dùng account trên hệ thống Sakai để nộp bài. **KHÔNG** nhận bài được gửi qua mail hoặc bất kỳ hình thức nào khác. Bài nộp trễ sẽ **KHÔNG** được nhận.

8. 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. Các bài làm của các sinh viên ở các học kỳ trước cũng sẽ được dùng để kiểm tra gian lận.
- Sinh viên không hiểu mã nguồn do chính mình viết, trừ những phần mã được cung cấp sẵn. 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.

Trong trường hợp bị kết luận là gian lận, sinh viên sẽ bị điểm 0 cho toàn bộ môn học.

KHÔNG CHẤP NHẬN BẤT KỲ GIẢI THÍCH NÀO

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.