Problem 1: Hệ thống quản lý sản phẩm sử dụng cấu trúc dữ liệu tuyến tính

Mô tả vấn đề:
Một hệ thống quản lý sản phẩm cần quản lý các sản phẩm với các mức độ ưu tiên của chúng. Mỗi sản phẩm có các thuộc tính sau:

- Mã sản phẩm (string)
- Mô tả sản phẩm (string)
- Ưu tiên (integer)

Yêu cầu:

1. Cấu trúc dữ liệu:

Thiết kế một cấu trúc dữ liệu tuyến tính thích hợp (ví dụ: ArrayList, LinkedList) để lưu trữ và quản lý các sản phẩm.

2. Thêm sản phẩm:

Viết hàm `void addProduct(String id, String description, int priority)` để thêm một sản phẩm mới vào hệ thống. Nếu mã sản phẩm đã tồn tại, cập nhật thông tin sản phẩm.

3. Xóa sản phẩm:

Viết hàm `void removeProduct(String id)` để xóa một sản phẩm khỏi hệ thống dựa trên mã sản phẩm.

4. Tìm kiếm sản phẩm:

Viết hàm `Product searchProduct(String id)` để tìm kiếm và trả về thông tin của một sản phẩm dựa trên mã sản phẩm.

5. Liệt kê sản phẩm:

Viết hàm `List<Product> listProductsByPriority()` để trả về danh sách tất cả các sản phẩm được sắp xếp theo mức độ ưu tiên giảm dần.

Problem 2: Quản lý dữ liệu xe ô tô

Mô tả vấn đề:
Ứng dụng này được thiết kế để quản lý một cơ sở dữ liệu các loại xe ô tô trong một showroom. Cơ sở dữ liệu bao gồm các loại xe, mỗi xe được xác định bằng ID, tên, hãng xe, số lượng bán ra, mức độ đánh giá của khách hàng và quốc gia sản xuất. Dữ liệu của các loại xe như sau:

Car ID	Name	Manufacturer	Units Sold	Customer Rating	Country
1	Model S	Tesla	5000	4.8	USA
2	Camry	Toyota	7000	4.5	Japan
3	3 Series	BMW	4500	4.6	Germany
4	Civic	Honda	8000	4.4	Japan

Car ID	Name	Manufacturer	Units Sold	Customer Rating	Country
5	A4	Audi	4200	4.7	Germany
6	F-150	Ford	9000	4.3	USA
7	Altima	Nissan	6500	4.2	Japan
8	Golf	Volkswagen	6000	4.5	Germany
9	Model 3	Tesla	7500	4.9	USA
10	Accord	Honda	7000	4.6	Japan

Yêu cầu:

1. Triển khai lớp Car:

- Triển khai một lớp `Car` với các thuộc tính: id, name, manufacturer, unitsSold, customerRating, và country. Bao gồm phương thức `compareTo` để hỗ trợ so sánh các loại xe dựa trên tên của chúng.

2. Cây nhị phân tìm kiếm (BST) để quản lý showroom:

- Triển khai lớp `BST` để quản lý các loại xe trong showroom. Bao gồm các phương thức để chèn và xóa xe.
- Triển khai phương thức trong `BST` để tìm kiếm xe với một tên cụ thể. Thảo luận về ưu và nhược điểm của việc sử dụng `BST` cho mục đích này.

3. Tái cấu trúc BST từ 2 kết quả duyệt cây:

- Triển khai phương thức trong `BST` để duyệt cây theo thứ tự in-order và trả về mảng các loại xe.
- Triển khai phương thức trong `BST` để duyệt cây theo thứ tự post-order và trả về mảng các loại xe.
- Triển khai phương thức khởi tạo `BST` từ hai mảng kết quả duyệt cây này.

4. Sắp xếp Quick Sort theo số lượng bán ra:

- Triển khai thuật toán Quick Sort để sắp xếp một mảng các loại xe theo số lượng bán ra.
- Giải thích tại sao Quick Sort được coi là một thuật toán sắp xếp hiệu quả và thảo luận về những ưu điểm của nó.

5. Hàm chính để minh họa:

- Khởi tạo mảng các loại xe với 10 xe từ bảng đã cho.
- Khởi tạo `BST` và chèn mảng các loại xe này vào `BST`.
- Minh họa thao tác xóa trên `BST`.
- Minh họa thao tác tìm kiếm trên `BST` với tên cụ thể.
- Sắp xếp các loại xe theo số lượng bán ra sử dụng Merge Sort và hiển thị xe có số lượng bán ra lớn nhất.
- Tạo `BST` trùng lặp từ hai mảng kết quả duyệt cây.