1. Lưu trữ danh sách sinh viên theo thứ tự nhập vào

* Cấu trúc dữ liệu phù hợp: Danh sách liên kết (LinkedList)
* Lý do chọn: Danh sách liên kết phù hợp với trường hợp này vì nó cho phép thêm phần tử vào cuối danh sách một cách dễ dàng và hiệu quả. Các phần tử trong danh sách liên kết có thể được thêm vào mà không cần phải di chuyển các phần tử khác, giúp tiết kiệm bộ nhớ và thời gian khi thêm sinh viên mới.

1. Tìm kiếm nhanh tên một sản phẩm trong danh sách hàng triệu sản phẩm

* Cấu trúc dữ liệu phù hợp: Bảng băm (Hash Table)
* Lý do chọn: Bảng băm cho phép tìm kiếm cực kỳ nhanh chóng, với độ phức tạp trung bình là O(1). Khi lưu trữ các sản phẩm, có thể sử dụng tên sản phẩm hoặc mã sản phẩm làm khóa, giúp tìm kiếm sản phẩm trong danh sách hàng triệu sản phẩm chỉ mất thời gian hằng số. Điều này là rất quan trọng khi cần tìm kiếm nhanh trong một tập dữ liệu lớn.

1. Quản lý danh sách các task công việc cần xử lý theo thứ tự ưu tiên

* Cấu trúc dữ liệu phù hợp: Hàng đợi ưu tiên (Priority Queue)
* Lý do chọn: Hàng đợi ưu tiên là cấu trúc dữ liệu lý tưởng cho các tác vụ cần xử lý theo thứ tự ưu tiên. Mỗi task công việc được gán một mức độ ưu tiên, và hàng đợi sẽ tự động xử lý các task với mức độ ưu tiên cao nhất trước. Cấu trúc này thường được triển khai bằng heap, giúp việc truy xuất và cập nhật các task có ưu tiên cao thực hiện nhanh chóng.

1. Lưu trữ dữ liệu hệ thống phân cấp như thư mục máy tính

* Cấu trúc dữ liệu phù hợp: Cây (Tree)
* Lý do chọn: Hệ thống thư mục trong máy tính có cấu trúc phân cấp, nơi mỗi thư mục có thể chứa các thư mục con. Cấu trúc cây rất thích hợp để mô tả hệ thống phân cấp này, trong đó mỗi thư mục là một nút trong cây, và các thư mục con là các nút con. Cây giúp quản lý và duyệt qua các thư mục dễ dàng, hỗ trợ các thao tác như tìm kiếm, thêm, xóa thư mục rất hiệu quả.

1. Tìm đường đi ngắn nhất giữa 2 điểm trên bản đồ

* Cấu trúc dữ liệu phù hợp: Đồ thị (Graph)
* Lý do chọn: Để mô phỏng các điểm trên bản đồ và các con đường nối chúng, đồ thị là cấu trúc dữ liệu phù hợp. Trong đồ thị, các điểm trên bản đồ là các đỉnh, và các con đường nối chúng là các cạnh. Để tìm đường đi ngắn nhất, các thuật toán như Dijkstra hoặc *A Search*\* có thể được áp dụng. Đồ thị cho phép mô phỏng và xử lý các kết nối phức tạp giữa các điểm trên bản đồ một cách hiệu quả.