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

* **Cấu trúc dữ liệu:** **Danh sách liên kết (Linked List)**
* **Lý do chọn:**
  + Danh sách liên kết cho phép thêm phần tử vào cuối danh sách một cách linh hoạt mà không cần phải thay đổi kích thước của danh sách.
  + Lưu trữ theo thứ tự nhập vào nghĩa là các phần tử sẽ được thêm vào cuối danh sách, điều này rất phù hợp với danh sách liên kết.
  + Dễ dàng quản lý bộ nhớ động khi cần thêm hoặc xóa sinh viên.

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

* **Cấu trúc dữ liệu:** **Bảng băm (Hash Table)**
* **Lý do chọn:**
  + Bảng băm có khả năng tìm kiếm, chèn và xóa phần tử với thời gian trung bình là O(1), rất nhanh chóng và hiệu quả.
  + Khi có hàng triệu sản phẩm, việc tìm kiếm nhanh tên sản phẩm là rất quan trọng, và bảng băm sẽ giúp tối ưu hóa quá trình này.
  + Đảm bảo truy cập nhanh chóng thông qua giá trị băm của tên sản phẩm.

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

* **Cấu trúc dữ liệu:** **Hàng đợi ưu tiên (Priority Queue)**
* **Lý do chọn:**
  + Hàng đợi ưu tiên cho phép quản lý các công việc với mức độ ưu tiên khác nhau.
  + Cấu trúc này sử dụng **heap** (đặc biệt là heap nhị phân) để đảm bảo rằng công việc có ưu tiên cao luôn được xử lý trước.
  + Quản lý công việc theo thứ tự ưu tiên là yêu cầu rất phổ biến trong lập trình hệ thống, và hàng đợi ưu tiên hỗ trợ rất tốt cho yêu cầu này.

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

* **Cấu trúc dữ liệu:** **Cây (Tree)**
* **Lý do chọn:**
  + Hệ thống thư mục có cấu trúc phân cấp (parent-child relationship), cây là cấu trúc dữ liệu phù hợp nhất để mô phỏng mối quan hệ này.
  + Cây giúp tổ chức dữ liệu theo dạng phân cấp, mỗi thư mục có thể chứa các thư mục con hoặc các tập tin, và việc tìm kiếm, thêm, xóa thư mục sẽ dễ dàng hơn nhiều khi sử dụng cây.
  + Các loại cây nhị phân tìm kiếm hoặc cây B+ thường được sử dụng trong các hệ thống tập tin hoặc cơ sở dữ liệu.

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

* **Cấu trúc dữ liệu:** **Đồ thị (Graph)**
* **Lý do chọn:**
  + Mối quan hệ giữa các điểm trên bản đồ là một đồ thị, với các điểm (hoặc địa điểm) là các đỉnh và các con đường giữa chúng là các cạnh.
  + Các thuật toán tìm đường như Dijkstra hoặc Bellman-Ford sẽ giúp tìm ra đường đi ngắn nhất từ điểm này đến điểm khác trên bản đồ.
  + Đồ thị có thể là đồ thị có hướng hoặc không có hướng, và có thể sử dụng trọng số trên các cạnh để biểu diễn độ dài hoặc thời gian di chuyển.