**Một số yêu cầu cần cải thiện:**

Trong quá trình thiết kế, phát triển và thử nghiệm chương trình "Quản lý sinh viên bằng các cấu trúc lưu trữ", nhóm đã tiến hành đánh giá hiệu quả hoạt động của các chức năng, đồng thời ghi nhận và phân tích một số hạn chế cũng như bất hợp lý trong quá trình vận hành. Từ đó, nhóm đề xuất một số điều chỉnh và cải tiến nhằm nâng cao độ chính xác, tốc độ xử lý, cũng như tối ưu trải nghiệm người dùng. Các thay đổi được chia thành bốn nhóm chính như sau:

1. **Khắc phục lỗi hiển thị sai thời gian giữa Binary Search và Brute Force Search (do in ngược thời gian của nhau).**
2. **Thay đổi định dạng mã số sinh viên thành (NxxABCDxxx).**
3. **Hiển thị thêm chi tiết thuật toán thời gian tìm kiếm của từng phần khi hiển thị danh sách.**
4. **Tối ưu hóa thao tác tìm sinh viên có điểm cao nhất/thấp nhất khi danh sách đã sắp xếp.**

**=> Những yêu cầu cần cải thiện, nhóm đã hoàn thành theo đúng yêu cầu của giáo viên và tối ưu chương trình hiệu quả nhất.**

**Chi tiết**

**1. Khắc phục lỗi hiển thị sai thời gian giữa Binary Search và Brute Force Search:**

**Mô tả vấn đề:** Thuật toán Binary Search hiển thị thời gian của thuật toán Brute Force Search và thuật toán Brute Force hiển thị thời gian của thuật toán Binary Search. Tuy vấn đề rất nhỏ nhưng điều này khiến thời gian Brute Force Search lớn hơn Binary Search làm cho việc đánh giá hiệu năng của hai phương pháp trở nên sai lệch và có thể gây hiểu lầm cho người dùng hoặc giảng viên khi kiểm tra kết quả.

**Giải pháp cải thiện:** Hoán đổi lại thời gian hiển thị của 2 thuật toán.

**2. Thay đổi định dạng mã số sinh viên (Student ID):**

**Mô tả vấn đề:** Định dạng cũ của mã số sinh viên được sử dụng trong chương trình là dạng 'N23DCCNxxx', trong đó 23 là năm nhập học, DCCN là mã chuyên ngành (Công nghệ thông tin), và xxx là số thứ tự sinh viên. Tuy nhiên, định dạng này chưa đủ linh hoạt để áp dụng cho các ngành học khác trong thực tế, như kế toán, quản trị kinh doanh, tài chính, v.v.

**Giải pháp cải thiện:** Chương trình đã được nâng cấp để hỗ trợ định dạng mới 'NxxABCDxxx', trong đó:

* Nxx: năm nhập học (ví dụ: N25 là năm 2025),
* ABCD: mã chuyên ngành (ví dụ: DCKT cho ngành Kế toán, DCKQ cho ngành Quản trị),
* xxx: số thứ tự sinh viên.

**Ví dụ cụ thể:**

* N25DCKT168 – sinh viên thứ 168 ngành Kế toán, nhập học năm 2025.
* N24DCBA045 – sinh viên thứ 45 ngành Ngân hàng, nhập học năm 2024.

Cấu trúc này giúp mở rộng tính ứng dụng của phần mềm cho các hệ thống quản lý sinh viên đa ngành và phù hợp hơn với thực tế tại các cơ sở đào tạo hiện nay.

**3. Hiển thị thêm chi tiết thuật toán thời gian tìm kiếm của từng phần khi danh sách:**

**Mô tả vấn đề:** Trong phiên bản trước, khi người dùng thực hiện thao tác tìm kiếm một sinh viên, kết quả chỉ đơn thuần là hiển thị thông tin sinh viên tìm được, mà không kèm theo thời gian thực hiện thao tác. Điều này khiến người dùng không có cơ sở để đánh giá hiệu suất xử lý của từng thuật toán được áp dụng, đặc biệt là trong bối cảnh cần so sánh các phương pháp tìm kiếm khác nhau.

**Giải pháp cải thiện:** Nhóm đã bổ sung thêm chức năng hiển thị thời gian chi tiết sau mỗi thao tác tìm kiếm, được tính bằng mili giây (ms) hoặc micro giây (μs) tùy theo độ chính xác của hệ thống đo. Khi tìm kiếm thành công, người dùng sẽ thấy thời gian được ghi chú cụ thể bên dưới thông tin sinh viên khi hiển thị danh sách.

**4. Tối ưu hóa thao tác tìm sinh viên có điểm cao nhất/thấp nhất khi danh sách đã sắp xếp:**

**Mô tả vấn đề:** Trong các chức năng liên quan đến xử lý điểm số, nhóm nhận thấy rằng khi tìm kiếm sinh viên có điểm cao nhất hoặc thấp nhất, chương trình vẫn đang duyệt toàn bộ danh sách sinh viên, kể cả trong trường hợp danh sách đã được sắp xếp theo điểm.

**Giải pháp cải thiện:** Chương trình đã tối ưu thuật toán để nhận biết khi danh sách điểm đã được sắp xếp. Trong các trường hợp đó:

* Điểm thấp nhất sẽ được truy xuất trực tiếp tại phần tử đầu tiên của danh sách,
* Điểm cao nhất sẽ được truy xuất trực tiếp tại phần tử cuối cùng của danh sách.

Nhờ đó, thời gian xử lý giảm đáng kể, đặc biệt khi làm việc với tập dữ liệu lớn. Điều này cũng góp phần minh họa rõ ràng cho vai trò của cấu trúc dữ liệu và thuật toán trong việc tối ưu hóa hiệu năng chương trình.