Bài 1:

1. Oval:

-Tên gọi: Bắt đầu / Kết thúc

-Chức năng: Biểu thị điểm bắt đầu hoặc kết thúc của quy trình.

2. Hình chữ nhật

-Tên gọi: Xử lý / Hoạt động

-Chức năng: Dùng để thể hiện một bước xử lý, hoạt động hoặc lệnh trong quy trình.

3. Hình thoi

-Tên gọi: Quyết định / Điều kiện

-Chức năng: Biểu thị các điểm kiểm tra điều kiện hoặc quyết định trong quy trình. Nếu có điều kiện, sẽ có nhánh "Có" và "Không".

4. Mũi tên

-Tên gọi: Dòng chảy / Luồng

-Chức năng: Dùng để chỉ hướng dòng chảy của quy trình, từ bước này sang bước khác.

Bài 2:

1.Mảng (Array)

* Đặc điểm chính:

+ Lưu trữ dữ liệu có kích thước cố định.

+Truy cập phần tử nhanh (O(1)) theo chỉ số.

-Ứng dụng phổ biến:

+Lưu trữ dữ liệu liên tiếp, truy cập nhanh theo chỉ số.

+Các ứng dụng cần truy cập nhanh và đơn giản như tìm kiếm, sắp xếp.

2. Danh sách liên kết (Linked List)

-Đặc điểm chính:

+Kích thước động, dễ thay đổ

+Truy cập chậm (O(n)) nhưng hiệu quả khi chèn/xóa.

* Ứng dụng phổ biến :

+Các ứng dụng cần chèn/xóa phần tử linh hoạt (như quản lý danh sách).

+Quản lý dữ liệu thay đổi liên tục, như danh sách khách hàng.

3. Ngăn xếp (Stack)

-Đac điểm chính :

+Theo nguyên lý LIFO (Last In, First Out).

+Thêm và xóa phần tử ở đầu danh sách.

-Ứng dụng phổ biến :

+Quản lý ngữ cảnh (undo/redo), phân tích cú pháp, kiểm tra dấu ngoặc.

+Quản lý lịch sử, xử lý yêu cầu dạng chuỗi.

4. Hàng đợi (Queue)

-Đặc điểm chính:

+Theo nguyên lý FIFO (First In, First Out).

+Thêm vào cuối, lấy từ đầu.

-Ứng dụng phổ biến

+Xử lý các tác vụ theo thứ tự, như quản lý tác vụ hệ thống, hàng đợi CPU.

+Xử lý yêu cầu trong hệ thống mạng, hàng đợi máy in.

5. Cây (Tree)

-Đăc điểm chính:

+Cấu trúc phân cấp, mỗi nút có thể có nhiều con.

+Cung cấp tìm kiếm, chèn và xóa hiệu quả.

-Ứng dụng :

+Quản lý dữ liệu phân cấp, như hệ thống tệp, cơ sở dữ liệu.

+Quản lý phân cấp thông tin, như cấu trúc tổ chức.

6. Đồ thị (Graph)

-Đac điểm chính:

+Gồm các đỉnh và các cạnh nối các đỉnh lại với nhau.

+Có thể có chu trình hoặc không.

-Ứng dụng :

+Mô tả mối quan hệ phức tạp giữa các đối tượng.

+Mạng xã hội, tìm đường, các thuật toán tối ưu.

Bài3:

 **Lưu trữ danh sách sinh viên theo thứ tự nhập vào**  
**Cấu trúc dữ liệu phù hợp:** **Danh sách liên kết (Linked List)**  
**Lý do:** Danh sách liên kết cho phép lưu trữ các phần tử theo thứ tự nhập vào mà không cần xác định trước kích thước, đồng thời hỗ trợ việc thêm phần tử vào đầu hoặc cuối danh sách nhanh chóng. Tuy nhiên, nếu yêu cầu duy trì thứ tự chính xác và việc chèn/xóa nhiều, danh sách liên kết là lựa chọn hợp lý.

 **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:** Bảng băm cung cấp tốc độ tìm kiếm nhanh (O(1) trung bình) bằng cách sử dụng khóa để ánh xạ tới giá trị tương ứng. Việc tìm kiếm sản phẩm trong một danh sách lớn sẽ hiệu quả hơn rất nhiều so với các cấu trúc dữ liệu khác như mảng hay danh sách liên kết.

 **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:** Hàng đợi ưu tiên giúp quản lý các tác vụ theo thứ tự ưu tiên, đảm bảo các tác vụ quan trọng được xử lý trước. Cấu trúc dữ liệu này rất hiệu quả trong việc quản lý tác vụ và cho phép truy xuất phần tử có độ ưu tiên cao nhất nhanh chóng.

 **Lưu trữ dữ liệu hệ thống phân cấp như thư mục máy tính**  
**Cấu trúc dữ liệu phù hợp:** **Cây (Tree)**  
**Lý do:** Hệ thống thư mục máy tính là một ví dụ điển hình của cấu trúc phân cấp, nơi mỗi thư mục có thể chứa các thư mục con hoặc tệp. Cây là cấu trúc dữ liệu lý tưởng để lưu trữ các thông tin phân cấp như vậy, giúp dễ dàng thực hiện các thao tác như tìm kiếm, chèn, hoặc xóa.

 **Tìm đường đi ngắn nhất giữa 2 điểm trên bản đồ**  
**Cấu trúc dữ liệu phù hợp:** **Đồ thị (Graph)**  
**Lý do:** Bản đồ giao thông hoặc bất kỳ mạng lưới nào có thể được mô phỏng bằng đồ thị, trong đó các điểm (đỉnh) đại diện cho các vị trí và các cạnh nối giữa các điểm đại diện cho các con đường. Các thuật toán như Dijkstra hoặc A\* sử dụng đồ thị để tìm đường đi ngắn nhất giữa hai điểm.