1. Khai báo cấu trúc **dothi** để lưu trữ đồ thị và các thông tin liên quan.
2. Hàm **nhapfile**: Đọc thông tin về số đỉnh và ma trận kề từ tệp và lưu vào cấu trúc **dothi**.
3. Hàm **nhap**: Nhập thông tin về số đỉnh và ma trận kề từ bàn phím và lưu vào cấu trúc **dothi**.
4. Hàm **xuatdd**: Xuất đường đi từ đỉnh **N** đến đỉnh **S** dựa trên mạng **vet**.
5. Hàm **xoax**: Xóa đỉnh **x** khỏi mảng **L** và cập nhật tại **Vt**.
6. Hàm **Dijkstra**: Thực hiện thuật toán Dijkstra để tìm đường đi ngắn nhất từ đỉnh **N** đến đỉnh **S**.
   * + Khởi tạo các mạng **L**, **Vt**, **W** và vet để lưu trữ thông tin về đỉnh và trong số.
     + Thiết lập trong số ban đầu của tất cả các đỉnh là **Max** (vô cùng).
     + Thiết lập trong số của đỉnh của **N** là 0.
     + Lặp lại cho đến khi tìm được đường đi đến đỉnh **S** hoặc đã duyệt qua tất cả các đỉnh:
       - Chọn đỉnh **x** có trong số nhỏ nhất từ các đỉnh còn lại.
       - Duyệt qua các đỉnh **L** còn lại và cập nhật trong số và đỉnh trước đó (nếu cần).
       - Xóa đỉnh **x** khỏi mảng **L** và cập nhật lại mảng **Vt**.
     + Nếu đã tìm được đường đi đến đỉnh S, xuất đường đi đó bằng cách gọi hàm xuatdd.
     + Nếu không có đường đi từ đỉnh **N** đến đỉnh **S**, thông báo không tồn tại đường đi.
7. Trong hàm **main**, tạo đối tượng **dothi** và thực hiện các thao tác sau:
   * + Nhập thông tin về đồ thị từ bàn phím.
     + Nhập hai đỉnh cần tìm đường đi ngắn nhất.
     + Gọi hàm **Dijkstra** để tìm và xuất đường đi ngắn nhất.
     + Yêu cầu người dùng nhập lựa chọn để tiếp tục hoặc thoát chương trình.