BÁO CÁO ĐỒ ÁN MÔN HỌC
Lớp: IT003.CTTN.P21

SINH VIÊN THỰC HIỆN

Mã sinh viên: 24521883 **Họ và tên: Lê Quang Trung**

TÊN ĐỀ TÀI: CHAPTERLY

CÁC NỘI DUNG CẦN BÁO CÁO:

1. Giới thiệu đồ án:

Mục tiêu: Xây dựng ứng dụng mô phỏng đồ thị bằng Python, hỗ trợ các chức năng:

- Tạo & chỉnh sửa đồ thị: Thêm/xóa đỉnh, kéo thả đỉnh, nối cạnh.
- Tính toán trên đồ thị: Kiểm tra liên thông (BFS), tìm đường đi ngắn nhất (BFS).
- Lưu trữ: Save/Load đồ thị dưới dạng JSON.

Công nghệ & Thư viện:

- Python 3
- Tkinter
- JSON
- Math
- Collections.deque

GitHub: https://github.com/trungngayxua/chapterly

2. Quá trình thực hiện:

Tuần 1: Khởi tạo dự án và thiết kế cơ bản

- Thiết lập môi trường phát triển:
  - Cài đặt Python 3.8+
  - Thiết lập PyCharm IDE
- Thiết kế kiến trúc ứng dụng:
  - Phân tích yêu cầu bài toán
  - Thiết kế class GraphEditor

    - Lập kế hoạch triển khai các module chính
- Triển khai giao diện cơ bản:
    - Tạo canvas vẽ đồ thị
    - Xây dựng control panel với các nút chức năng
    - Thiết lập các chế độ làm việc (select, add node, add edge, delete)

Tuần 2: Triển khai chức năng cốt lõi

- Hoàn thiện thao tác với đỉnh:
    - Thêm/xóa đỉnh
    - Kéo thả đỉnh (drag & drop)
    - Fix lỗi di chuyển đỉnh kèm cạnh
- Triển khai thao tác với cạnh:
    - Thêm/xóa cạnh
    - Kiểm tra trùng lặp cạnh
    - Tự động cập nhật khi đỉnh di chuyển
- Cải tiến giao diện:
    - Thay đổi cursor theo chế độ
    - Highlight đỉnh/cạnh khi tương tác
    - Hiển thị trạng thái chế độ hiện tại

Tuần 3: Triển khai thuật toán đồ thị

- Triển khai thuật toán kiểm tra liên thông:
    - Sử dụng BFS duyệt đồ thị
    - Đếm số thành phần liên thông
    - Hiển thị kết quả trực quan
- Triển khai tìm đường đi ngắn nhất:
    - Giao diện chọn đỉnh đầu/cuối
    - Triển khai BFS tìm đường đi
    - Highlight đường đi tìm được
- Xử lý ngoại lệ:
    - Kiểm tra điều kiện biên
    - Thông báo lỗi chi tiết
    - Phục hồi trạng thái khi có lỗi

Tuần 4: Hoàn thiện và tối ưu

- Triển khai lưu/tải đồ thị:
    - Định dạng lưu trữ JSON
    - Serialize/deserialize đồ thị

- Validate dữ liệu khi tải
- Tối ưu hiệu năng:
  - Cải thiện tốc độ vẽ đồ thị lớn
  - Tối ưu thuật toán tìm đường
  - Giảm thiểu render thừa
- Hoàn thiện tài liệu:
  - Viết docstring đầy đủ
  - Bổ sung comments code
  - Chuẩn bị báo cáo
- Kiểm thử tổng thể:
  - Test các trường hợp biên
  - Kiểm tra tính tương thích
  - Fix các lỗi phát sinh

3. Kết quả đạt được:

- Đã hoàn thành tất cả yêu cầu: tạo đỉnh, kéo thả, kiểm tra liên thông, tìm đường ngắn nhất, save/load đồ thị
- Chức năng hoạt động ổn định, giao diện trực quan
- Code có cấu trúc rõ ràng, đầy đủ docstring

So với yêu cầu:

- Đáp ứng 100% yêu cầu ban đầu
- Một số tính năng vượt trội: highlight đường đi, kiểm tra dữ liệu khi load file

4. Tài liệu tham khảo

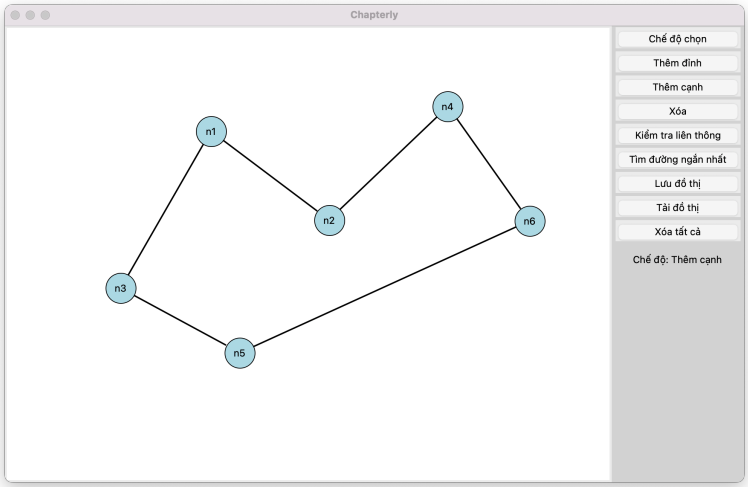
a) Documentation :

- Python Documentation *(Ngôn ngữ chính)*
- Tkinter GUI Documentation *(Thư viện xây dựng giao diện)*
- JSON Module *(Lưu trữ đồ thị)*

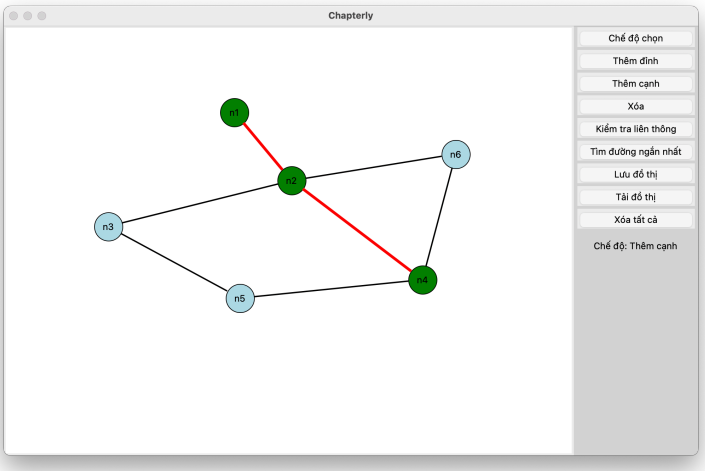
b) Source Code :

- Tkinter Canvas Tutorial (Real Python) *(Kéo thả đỉnh)*
- Python Graph Implementation (GitHub Gist) *(Cấu trúc đồ thị cơ bản)*

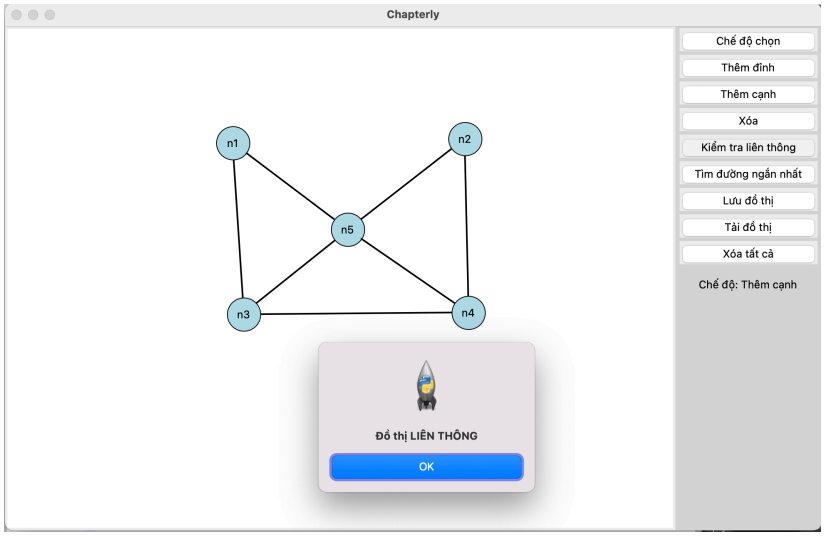
5. Phụ lục 1: Giới thiệu (demo) kết quả



Đồ thị có 6 đỉnh



Highligh đường đi ngắn nhất từ 1 -> 4



Kiểm tra đồ thị liên thông

6. Phụ lục 2: docstring

Lớp GraphEditor

```
"""
Lớp chính để tạo và quản lý ứng dụng đồ thị tương tác sử dụng
Tkinter.

Ứng dụng cho phép người dùng:
- Thêm, xóa, kéo thả đỉnh trên canvas.
- Thêm, xóa cạnh giữa các đỉnh.
```

```
- Kiểm tra tính liên thông của đồ thị.
- Tìm đường đi ngắn nhất giữa hai đỉnh.
- Lưu và tải đồ thị từ file JSON.
"""
```

Phương thức __init__

```
"""
Khởi tạo ứng dụng đồ thị với cửa sổ chính Tkinter.

Tham số:
    root (tk.Tk): Cửa sổ gốc của ứng dụng.
"""
```

Phương thức setup_ui

```
"""
Thiết lập giao diện người dùng với các nút điều khiển và nhãn
trạng thái.

Các nút bao gồm:
- Chế độ chọn, thêm đỉnh, thêm cạnh, xóa
- Kiểm tra liên thông
- Tìm đường ngắn nhất
- Lưu, tải và xóa đồ thị
"""
```

Phương thức set_select_mode

```
"""Chuyển sang chế độ chọn đỉnh để có thể kéo thả."""
```

Phương thức set_add_node_mode

```
"""Chuyển sang chế độ thêm đỉnh mới."""
```

Phương thức set_add_edge_mode

```
"""Chuyển sang chế độ thêm cạnh giữa hai đỉnh."""
```

Phương thức set_delete_mode

```
"""Chuyển sang chế độ xóa đỉnh hoặc cạnh."""
```

Phương thức handle_click

```
"""
Xử lý sự kiện click chuột trên canvas.

Tùy theo chế độ hiện tại, xử lý thêm đỉnh, thêm cạnh,
chọn đỉnh để kéo hoặc xóa đối tượng tại vị trí click.

Tham số:
    event (tk.Event): Đối tượng sự kiện chuột.
"""
```

Phương thức handle_drag

```
"""
Xử lý kéo thả chuột để di chuyển đỉnh khi ở chế độ chọn.

Tham số:
    event (tk.Event): Đối tượng sự kiện kéo chuột.
"""
```

Phương thức handle_release

```
"""
Xử lý thả chuột kết thúc thao tác kéo đỉnh.

Tham số:
    event (tk.Event): Đối tượng sự kiện thả chuột.
```

```
"""
```

Phương thức create_node

```
"""
Tạo đỉnh mới tại vị trí (x, y).

Tham số:
    x (int): Tọa độ x trên canvas.
    y (int): Tọa độ y trên canvas.
"""
```

Phương thức create_edge

```
"""
Tạo cạnh nối giữa hai đỉnh node1 và node2.

Nếu cạnh đã tồn tại hoặc node1 == node2 thì không tạo.

Tham số:
    node1 (str): ID đỉnh đầu.
    node2 (str): ID đỉnh cuối.
"""
```

Phương thức draw_edge

```
"""
Vẽ cạnh nối giữa hai đỉnh lên canvas.

Tính toán điểm bắt đầu và kết thúc trên mép hình tròn đại diện
đỉnh.

Tham số:
    node1 (str): ID đỉnh đầu.
    node2 (str): ID đỉnh cuối.
"""
```

Phương thức update_edges_for_node

```
"""
Cập nhật lại vị trí các cạnh có nối với đỉnh node_id khi đỉnh thay
đổi vị trí.

Tham số:
    node_id (str): ID đỉnh cần cập nhật cạnh.
"""
```

Phương thức delete_at_position

```
"""
Xóa đỉnh hoặc cạnh tại vị trí (x, y) nếu có.

Ưu tiên xóa cạnh nếu click gần cạnh, nếu không thì xóa đỉnh.

Tham số:
    x (int): Tọa độ x.
    y (int): Tọa độ y.
"""
```

Phương thức delete_node

```
"""
Xóa đỉnh và tất cả các cạnh liên quan tới đỉnh đó.

Tham số:
    node_id (str): ID đỉnh cần xóa.
"""
```

Phương thức delete_edge

```
"""
Xóa cạnh nối giữa hai đỉnh node1 và node2.

Tham số:
    node1 (str): ID đỉnh đầu.
    node2 (str): ID đỉnh cuối.
```

```
"""
```

Phương thức get_node_at_position

```
"""
Tìm đỉnh tại vị trí (x, y).

Trả về ID đỉnh nếu có, hoặc None nếu không có đỉnh nào gần.

Tham số:
    x (int): Tọa độ x.
    y (int): Tọa độ y.

Trả về:
    str hoặc None: ID đỉnh tìm được hoặc None.
"""
```

Phương thức get_edge_at_position

```
"""
Tìm cạnh gần vị trí (x, y) nhất trong khoảng cách threshold.

Tham số:
    x (int): Tọa độ x.
    y (int): Tọa độ y.
    threshold (int): Khoảng cách tối đa để nhận diện cạnh.

Trả về:
    tuple hoặc None: Cặp (node1, node2) của cạnh hoặc None nếu
không tìm được.
"""
```

Phương thức point_to_line_distance

```
"""
Tính khoảng cách từ điểm (x, y) đến đoạn thẳng nối (x1, y1) - (x2,
y2).

Tham số:
```

```
    x, y (int): Tọa độ điểm.
    x1, y1, x2, y2 (int): Tọa độ hai đầu đoạn thẳng.

Trả về:
    float: Khoảng cách từ điểm đến đoạn thẳng.
"""
```

Phương thức highlight_node

```
"""
Làm nổi bật đỉnh node_id trên canvas (thay đổi màu sắc).

Tham số:
    node_id (str): ID đỉnh cần làm nổi bật.
"""
```

Phương thức reset_edge_mode

```
"""
Reset trạng thái chọn đỉnh khi đang ở chế độ thêm cạnh.
"""
```

Phương thức check_connectivity

```
"""
Kiểm tra đồ thị có liên thông hay không.

Sử dụng thuật toán BFS để duyệt các đỉnh.

Hiển thị hộp thoại thông báo kết quả.
"""
```

Phương thức find_shortest_path

```
"""
Mở cửa sổ phụ để chọn 2 đỉnh rồi tìm đường đi ngắn nhất giữa
chúng.
```

```
Sử dụng BFS tìm đường.

Nếu tìm được, sẽ làm nổi bật đường đi và hiển thị hộp thoại kết
quả.
"""
```

Phương thức bfs_shortest_path

```
"""
Tìm đường đi ngắn nhất giữa hai đỉnh bằng thuật toán BFS.

Tham số:
    start (str): ID đỉnh bắt đầu.
    end (str): ID đỉnh kết thúc.

Trả về:
    list hoặc None: Danh sách ID đỉnh theo đường đi ngắn nhất hoặc
None nếu không tìm được.
"""
```

Phương thức highlight_path

```
"""Làm nổi bật đường đi trên đồ thị."""
```

Phương thức save_graph

```
"""Lưu đồ thị vào file JSON."""
```

Phương thức load_graph

```
"""Tải đồ thị từ file JSON."""
```

Phương thức clear_all

```
    """Xóa toàn bộ đồ thị."""
```