

TRƯỜNG ĐẠI HỌC CẦN THƠ
KHOA CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG
BỘ MÔN CÔNG NGHỆ PHẦN MỀM



NIÊN LUẬN CƠ SỞ NGÀNH KTPM KHÓA 45

BÀI TOÁN TÌM CHU TRÌNH EULER

Sinh viên thực hiện:
Lê Thái Giao Tiên
MSSV: B1906586
Lớp: DI1996A5

Cán bộ hướng dẫn:
Ths. Phan Huy Cường

HỌC KỲ 2, 2021-2022

ĐÁNH GIÁ KẾT QUẢ THỰC HIỆN NIÊN LUẬN CƠ SỞ NGÀNH KTPM

(Học kỳ 2, Năm học 2021-2022)

GIÁO VIÊN HƯỚNG DẪN:

STT	HỌ VÀ TÊN	MSCB
1		

SINH VIÊN THỰC HIỆN:

STT	HỌ VÀ TÊN	MSSV	THƯỜNG (Tối đa 1,0 điểm)	ĐIỂM
1				

I. HÌNH THỨC (Tối đa 0,5 điểm)		
Bìa (tối đa 0,25 điểm)		
<ul style="list-style-type: none">Các tiêu đề: Trường ĐHCT, Khoa CNTTLoại niên luận: Cơ sở ngành KTPM, Tên đề tàiGiáo viên hướng dẫn: chức danh, họ tên.Thông tin về các sinh viên thực hiện: họ tên, mã số, lớpNăm thực hiện		
Bố cục (tối đa 0,25 điểm)		
<ul style="list-style-type: none">Nhận xét của giáo viên hướng dẫn và giáo viên chấmMục lục: cấu trúc chương, mục và tiểu mụcPhụ lục (nếu có)Tài liệu tham khảo		
II. NỘI DUNG (Tối đa 3,5 điểm)		
Tổng quan (tối đa 0,5 điểm)		
<ul style="list-style-type: none">Mô tả bài toán, mục tiêu cần đạt được (0,25 điểm)Hướng giải quyết và kế hoạch thực hiện (0,25 điểm)		
Lý thuyết (tối đa 0,5 điểm)		
<ul style="list-style-type: none">Các khái niệm sử dụng trong đề tàiKết quả vận dụng lý thuyết vào đề tài		
Ứng dụng (tối đa 2,0 điểm)		
<ul style="list-style-type: none">Phân tích yêu cầu bài toán, xây dựng các cấu trúc dữ liệu cần thiết (tối đa 0,5 điểm)Giải thuật (Lưu đồ-Ngôn ngữ giả) (1,0 điểm)Giới thiệu chương trình (0,5 điểm)		
Kết luận (tối đa 0,5 điểm)		
<ul style="list-style-type: none">Nhận xét kết quả đạt đượcHạn chếHướng phát triển		
III. CHƯƠNG TRÌNH DEMO (Tối đa 5,0 điểm)		
Giao diện thân thiện với người dùng (1,0 điểm)		
Hướng dẫn sử dụng (0,5 điểm)		
Kết quả thực hiện đúng với kết quả của phần ứng dụng (3,5 điểm)		

Ghi chú:

Nếu sinh viên demo chương trình và trả lời vấn đáp không đạt yêu cầu của giáo viên hướng dẫn thì sinh viên sẽ nhận điểm F cho học phần này.

Cần Thơ, ngày tháng năm 2022
GIÁO VIÊN CHẤM

MỤC LỤC

CHƯƠNG 1. TỔNG QUAN	1
I. MÔ TẢ BÀI TOÁN	1
II. MỤC TIÊU CẦN ĐẠT ĐƯỢC.....	1
III. HƯỚNG DẪN GIẢI QUYẾT.....	1
IV. KẾ HOẠCH THỰC HIỆN.....	2
CHƯƠNG 2. LÝ THUYẾT	2
I. CÁC KHÁI NIỆM ĐƯỢC SỬ DỤNG TRONG ĐỀ TÀI [1]	2
2.1.1. Khái niệm Đồ thị vô hướng	2
2.1.2. Khái niệm Đường đi	2
2.1.3. Duyệt đồ thị theo chiều sâu DFS (Depth – First Search)	3
2.1.4. Duyệt đồ thị theo chiều rộng BFS (Breadth – First Search).....	3
2.1.5. Khái niệm Đồ thị vô hướng liên thông và bộ phận liên thông	3
2.1.6. Khái niệm Chu trình	3
2.1.7. Khái niệm Đồ thị Euler.....	3
2.1.8. Khái niệm Chu trình Euler.....	3
II. KẾT QUẢ VẬN DỤNG LÝ THUYẾT VÀO ĐỀ TÀI	3
CHƯƠNG 3. KẾT QUẢ ỨNG DỤNG	3
I. PHÂN TÍCH YÊU CẦU BÀI TOÁN	3
II. XÂY DỰNG CÁC CẤU TRÚC DỮ LIỆU CẦN THIẾT	4
III. THIẾT KẾ GIẢI THUẬT	4
3.3.1. Menu điều khiển vẽ đồ thị	4
3.3.2. Bảng vẽ đồ thị.....	9
3.3.3. Số miền liên thông	12
3.3.4. Chu trình Euler.....	16
3.3.5. Bảng thông báo	20
IV. GIỚI THIỆU CHƯƠNG TRÌNH	20
3.4.1. Menu điều khiển vẽ đồ thị	20
3.4.2. Bảng vẽ đồ thị.....	21
3.4.3. Giải thuật của đồ thị.....	21
3.4.4. Bảng thông báo	21
3.4.5. Hướng dẫn sử dụng.....	21

CHƯƠNG 4. KẾT LUẬN – ĐÁNH GIÁ	22
I. KẾT QUẢ ĐẠT ĐƯỢC.....	22
II. THU HOẠCH.....	22
4.2.1. Về mặt chuyên môn	22
4.2.2. Về mặt kinh nghiệm.....	22
III. HẠN CHẾ - NGUYÊN NHÂN.....	22
IV. HƯỚNG PHÁT TRIỂN	22
CHƯƠNG 5. PHẦN PHỤ LỤC	22
CHƯƠNG 6. TÀI LIỆU THAM KHẢO.....	22

DANH MỤC HÌNH

Hình 1-1 Lưu đồ nhấn nút "Hướng dẫn"	4
Hình 1-2 Lưu đồ nhấn nút "Cập nhật đồ thị"	4
Hình 1-3 Lưu đồ nút “Vẽ đỉnh”	5
Hình 1-4 Lưu đồ nút "Vẽ cạnh"	5
Hình 1-5 Lưu đồ nút "Di chuyển đỉnh"	6
Hình 1-6 Lưu đồ nút "Mở đồ thị"	7
Hình 1-7 Lưu đồ nút "Lưu đồ thị"	8
Hình 1-8 Lưu đồ nút "Tạo đồ thị mới"	8
Hình 2-1 Lưu đồ xử lý Vẽ đồ thị	9
Hình 2-2 Lưu đồ xử lý Hiện thị đồ thị.....	9
Hình 2-3 Lưu đồ thao tác "Xóa đỉnh"	10
Hình 2-4 Lưu đồ thao tác "Xóa cạnh"	11
Hình 3-1 Lưu đồ nút “Số miền liên thông”	12
Hình 3-2 Lưu đồ giải thuật Duyệt đồ thị theo chiều sâu DFS của hàm dfs()	13
Hình 3-3 Lưu đồ giải thuật Duyệt đồ thị theo chiều rộng BFS của hàm bfs()	14
Hình 3-4 Lưu đồ giải thuật Kiểm tra số miền liên thông của hàm findConnectedComponent()	15
Hình 4-1 Lưu đồ nút "Chu trình Euler"	16
Hình 4-2 Lưu đồ giải thuật của hàm Tính bậc degree()	17
Hình 4-3 Lưu đồ giải thuật của hàm Kiểm tra liên thông isConnected()	18
Hình 4-4 Lưu đồ giải thuật của hàm Tìm chu trình Euler findEulerCircuit()	19
Hình 5-1 Lưu đồ xử lý Hiện thị bảng thông báo	20
Hình 5-1 Giao diện phần mềm "Euler Circuit"	20

CHƯƠNG 1. TỔNG QUAN

I. MÔ TẢ BÀI TOÁN

Bắt nguồn từ bài toán bảy cây cầu ở Königsberg nước Đức. Có một con sông chảy qua thành phố làm cho trung tâm của thành phố thành một hòn đảo. Sau khi vượt qua hòn đảo con sông bị chia làm hai nhánh. Bảy cây cầu được xây dựng để mọi người có thể đi từ phần đất này sang phần đất khác.

Bài toán đặt ra là: Có thể nào, xuất phát từ một phần đất nào đó, đi qua tất cả các cây cầu đúng một lần và quay về nơi xuất phát không?

Nhà toán học Euler đã trả lời trọn vẹn bài toán này.

Để mọi người có thể hiểu rõ và nhìn thấy một cách trực quan trên đồ thị có chứa cách nào đi qua tất cả các cạnh/cung đúng một lần và quay trở về nơi xuất phát (còn gọi là tìm chu trình Euler) hay đơn giản kiểm chứng xem có tồn tại đường đi nào để mọi người đi qua hết bảy cây cầu ở Königsberg nước Đức không? Đó là lý do phần mềm “Euler Circuit” ra đời.

Phần mềm sử dụng các hình ảnh trực quan để tìm ra chu trình Euler một cách dễ hiểu nhất. Do đó, phần mềm “Euler Circuit” giúp đỡ phần nào cho việc học Lý thuyết đồ thị của sinh viên, hay hỗ trợ phát triển những tựa game có ứng dụng của chu trình Euler.

II. MỤC TIÊU CẦN ĐẠT ĐƯỢC

Nắm kiến thức cấu trúc dữ liệu và giải thuật, lý thuyết đồ thị về đồ thị vô hướng, đường đi, các cách duyệt theo chiều sâu DFS, duyệt theo chiều rộng BFS, miền liên thông của đồ thị và chu trình Euler.

Tạo bất kì đồ thị vô hướng và biểu diễn chu trình Euler thông qua giao diện đồ họa có sử dụng AWT - các (API) cho lập trình GUI trong Java.

Biết cách thức tự tìm hiểu một vấn đề và xây dựng những thứ riêng biệt cho phần mềm của mình từ logo, các icon đến đối tượng của phần mềm.

Đối với giao diện đồ họa cần dễ sử dụng, có các chức năng cơ bản cho phép dùng chuột vẽ, thêm, sửa, xóa, di chuyển các đỉnh và cung, lưu và mở đồ thị. Cung cấp các chức năng kiểm tra số miền liên thông của đồ thị và tìm chu trình Euler.

III. HƯỚNG DẪN GIẢI QUYẾT

Phần mềm được viết bằng ngôn ngữ Java, gồm một form chính chứa:

- Panel điều khiển việc vẽ đồ thị vô hướng: tạo đồ thị mới, vẽ đỉnh, vẽ cung, di chuyển đỉnh, cập nhật lại đồ thị sau khi chỉnh sửa, lưu và mở đồ thị.
- Panel vẽ và hiển thị đồ thị.
- Panel chứa các chức năng kiểm tra số miền liên thông và tìm chu trình Euler.
- Panel hiển thị kết quả kiểm tra số miền liên thông và chu trình Euler.

IV. KẾ HOẠCH THỰC HIỆN

Tuần 1 – 4:

- Xem lại giáo trình lý thuyết đồ thị về đồ thị vô hướng, đường đi, các cách duyệt DFS, BFS, đồ thị liên thông và đường đi Euler, chu trình Euler.
- Tham khảo các chương trình có trên Internet và Youtube.

Tuần 5 – 7:

- Tiếp tục tham khảo từ các web. Chọn ngôn ngữ Java và công cụ lập trình Netbeans để tìm hiểu và thực hiện chương trình.
- Tìm hiểu các lớp Java để vẽ giao diện.
- Mô tả phần tổng quan của bài báo cáo.
- Tạo form chương trình Euler Circuit.

Tuần 8 – 11:

- Vẽ lưu đồ giải thuật.
- Tạo chức năng cho chương trình Euler Circuit.

Tuần 12 – 15:

- Tiếp tục xây dựng giải thuật và giao diện cho chương trình Euler Circuit.

Tuần 16 – 17:

- Viết hoàn chỉnh bài báo cáo.

CHƯƠNG 2. LÝ THUYẾT

I. CÁC KHÁI NIỆM ĐƯỢC SỬ DỤNG TRONG ĐỀ TÀI [1]

2.1.1. Khái niệm Đồ thị vô hướng

Đồ thị vô hướng G bao gồm một cặp (E, V) , ký hiệu $G = (E, V)$, trong đó:

- E là một tập hợp mà các phần tử của nó được gọi là các đỉnh của đồ thị.
- V là một tập hợp mà các phần tử của nó là các cặp đỉnh không tính đến thứ tự $v = (i, j) = (j, i)$, gọi là cạnh nối giữa đỉnh i và đỉnh j của đồ thị.

Bằng hình vẽ, các đỉnh của đồ thị vô hướng được trình bày bằng các điểm, các cạnh được trình bày bằng đoạn thẳng nối hai đỉnh. Khi tồn tại một cạnh nối giữa hai đỉnh thì có thể xem như tồn tại hai cung trái chiều nhau giữa hai đỉnh đó.

2.1.2. Khái niệm Đường đi

Đường đi (vô hướng) trên một đồ thị vô hướng là một dãy các cạnh kề nhau.

Một đường đi được diễn tả bằng một bộ các đỉnh nằm trên đường đi. Đỉnh mà tại đó đường đi bắt đầu gọi là đỉnh đầu, đỉnh mà tại đó đường đi kết thúc gọi là đỉnh cuối. Số cạnh nằm trên đường đi gọi là độ dài của đường đi.

2.1.3. Duyệt đồ thị theo chiều sâu DFS (Depth – First Search)

Khi trong thuật toán duyệt các đỉnh đồ thị, nếu danh sách L được tổ chức theo stack (kiểu ngăn xếp: vào trước ra sau) thì người ta có phương pháp duyệt theo chiều sâu. Trong phương pháp này mỗi lần duyệt một đỉnh ta duyệt đến tận cùng mỗi nhánh rồi mới chuyển sang duyệt nhánh khác.

2.1.4. Duyệt đồ thị theo chiều rộng BFS (Breadth – First Search)

Khi trong thuật toán duyệt các đỉnh đồ thị, nếu danh sách L được tổ chức theo queue (kiểu hàng đợi: vào trước ra trước) thì người ta có phương pháp duyệt theo chiều rộng. Trong phương pháp này việc duyệt có tính chất lan rộng. Một đỉnh được duyệt xong thì duyệt ngay các đỉnh kề với nó.

2.1.5. Khái niệm Đồ thị vô hướng liên thông và bộ phận liên thông

Đồ thị vô hướng G được gọi là liên thông nếu với mỗi cặp đỉnh i, j bất kỳ thì luôn tìm được đường đi nối i và j . Đường đi nối i và j cũng là đường đi nối j và i .

a- Đồ thị G là liên thông khi và chỉ khi G chỉ có một bộ phận liên thông.

b- Mỗi bộ phận liên thông là một đồ thị liên thông.

c- Mỗi đỉnh cô lập là một bộ phận liên thông.

2.1.6. Khái niệm Chu trình

Một đường đi mà đỉnh đầu và đỉnh cuối của nó trùng nhau được gọi là chu trình.

2.1.7. Khái niệm Đồ thị Euler

Đồ thị G có chứa chu trình Euler được gọi là đồ thị Euler.

2.1.8. Khái niệm Chu trình Euler

Một chu trình trên đồ thị hữu hạn liên thông G đi qua tất cả các cạnh/cung của G và mỗi cạnh/cung đúng một lần được gọi là chu trình Euler.

II. KẾT QUẢ VẬN DỤNG LÝ THUYẾT VÀO ĐỀ TÀI

Sử dụng mảng động (Arraylist trong Java) để lưu trữ các đối tượng đỉnh và cung của đồ thị.

Áp dụng các giải thuật duyệt, tìm số miền liên thông của đồ thị vô hướng và tìm chu trình Euler để đưa vào giao diện đồ họa trực quan.

Sử dụng các lớp có sẵn trong Java và bất sự kiện để tạo nên giao diện giúp người dùng dễ dàng vẽ ra đồ thị và quan sát các giải thuật.

CHƯƠNG 3. KẾT QUẢ ỨNG DỤNG

I. PHÂN TÍCH YÊU CẦU BÀI TOÁN

Phần mềm “Euler Circuit” yêu cầu xây dựng giao diện đồ họa cho phép người dùng nhập đồ thị bằng cách dùng chuột để nhập đỉnh, cạnh,... và lưu đồ thị thành file sau khi vẽ, cũng như mở file đồ thị đã lưu.

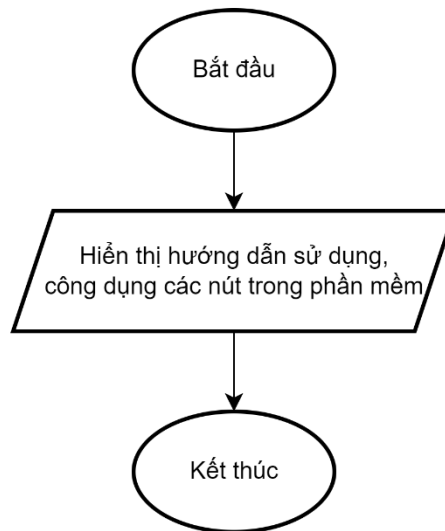
Sau khi vẽ đồ thị, chương trình cung cấp cho người dùng chức năng kiểm tra số miền liên thông của đồ thị và tìm chu trình Euler.

II. XÂY DỰNG CÁC CẤU TRÚC DỮ LIỆU CẦN THIẾT

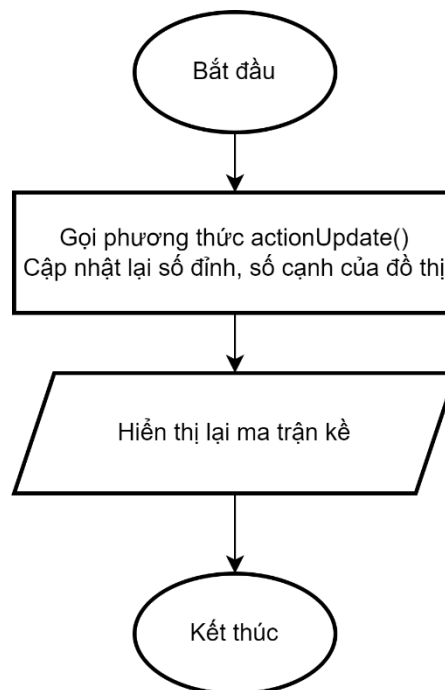
Tạo cấu trúc dữ liệu danh sách đặc trưng – Arraylist để lưu trữ đỉnh, cung của đồ thị.

III. THIẾT KẾ GIẢI THUẬT

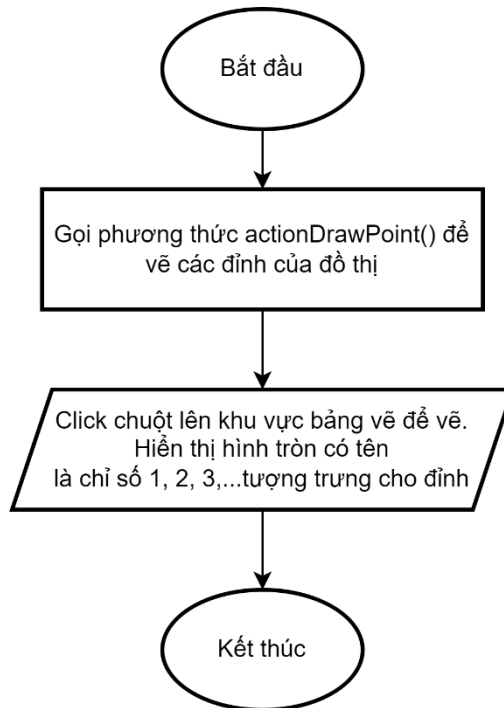
3.3.1. Menu điều khiển vẽ đồ thị



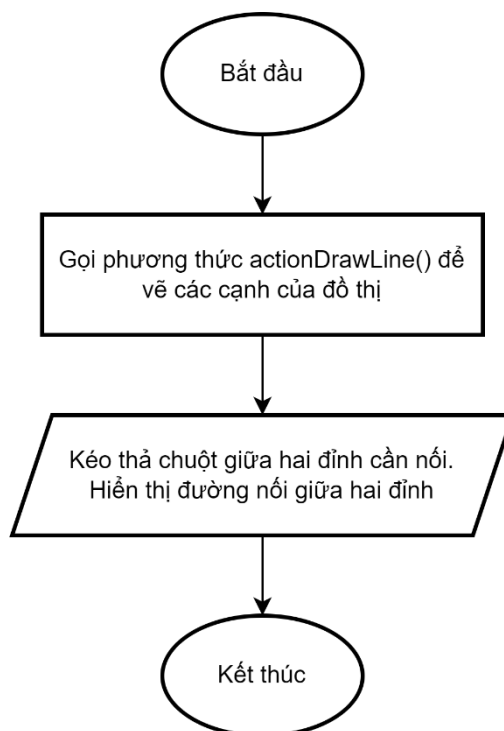
Hình 1-1 Lưu đồ nhấn nút "Hướng dẫn"



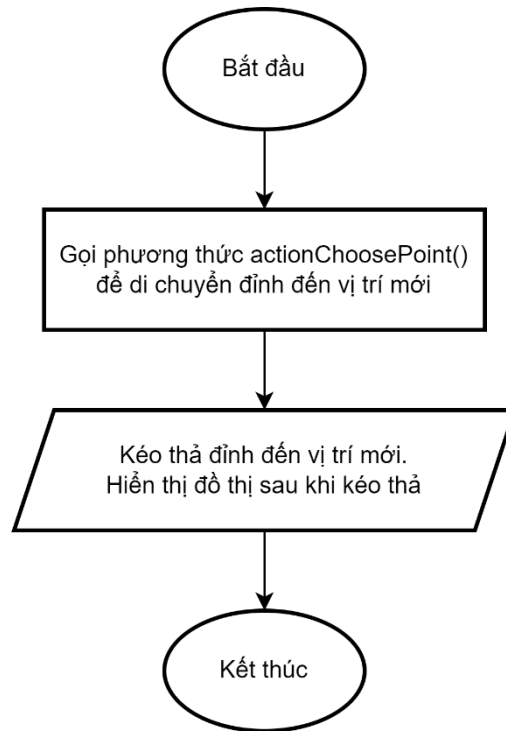
Hình 1-2 Lưu đồ nhấn nút "Cập nhật đồ thị"



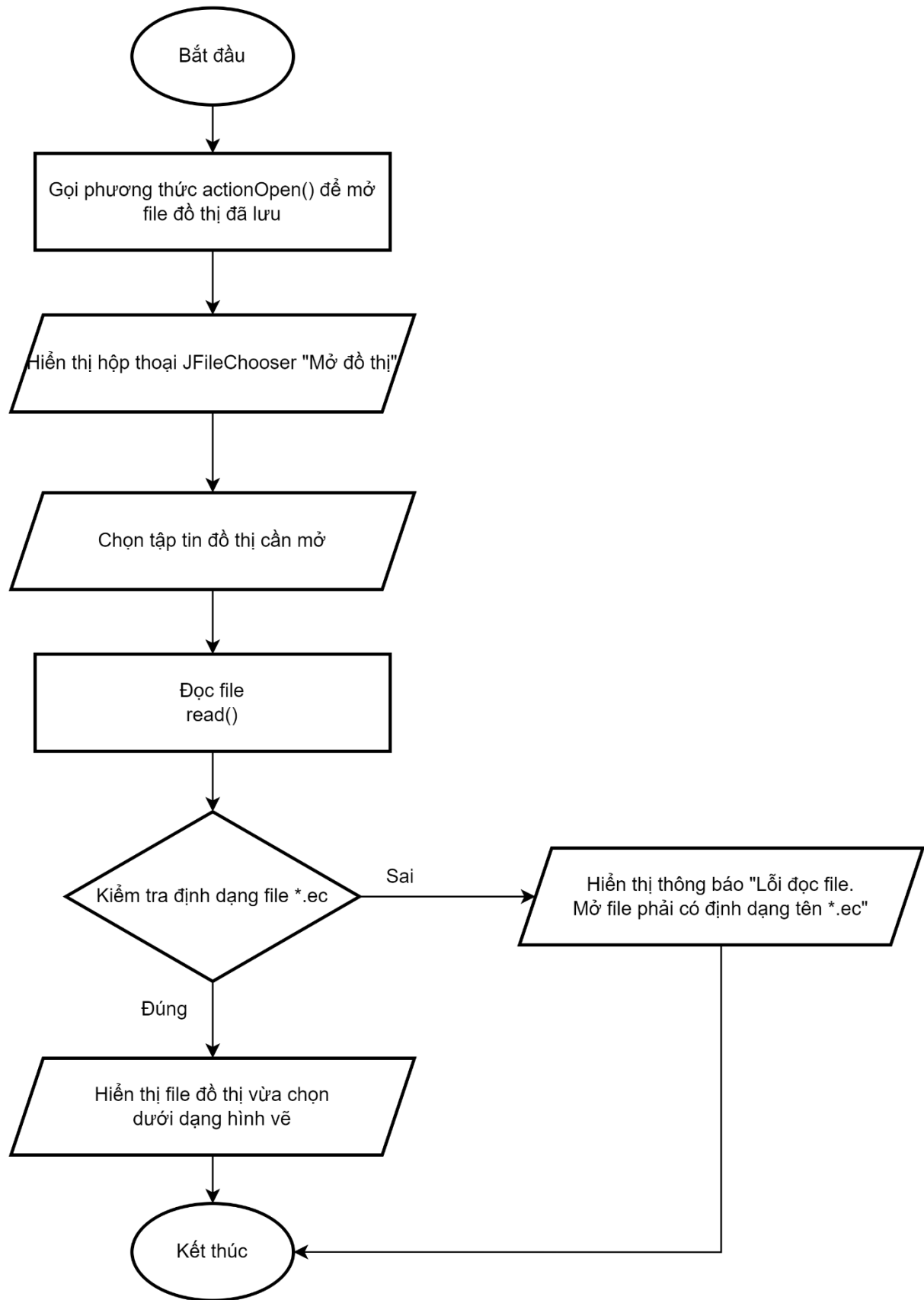
Hình 1-3 Lưu đồ nút “Vẽ đỉnh”



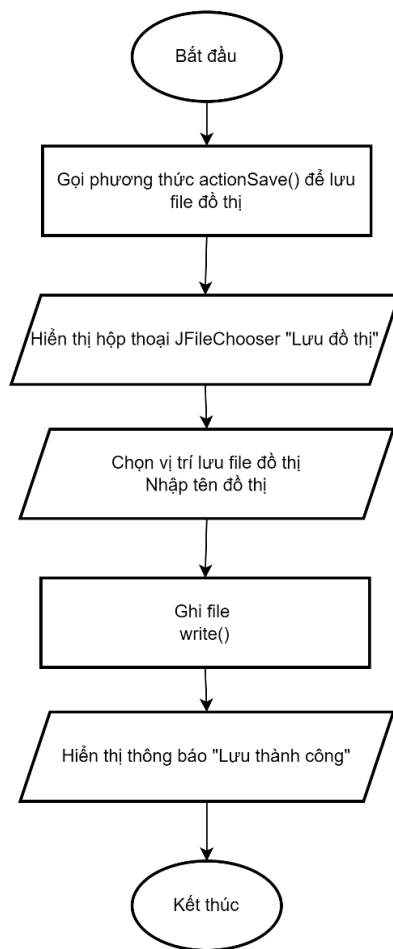
Hình 1-4 Lưu đồ nút “Vẽ cạnh”



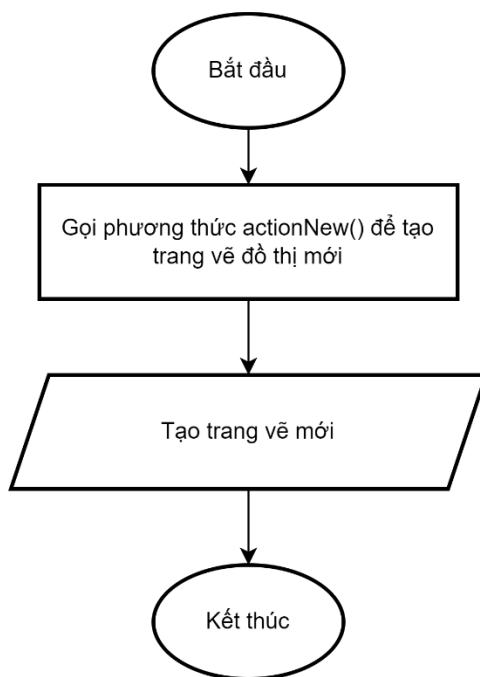
Hình 1-5 Lưu đồ nút "Di chuyển đỉnh"



Hình 1-6 Lưu đồ nút "Mở đồ thị"

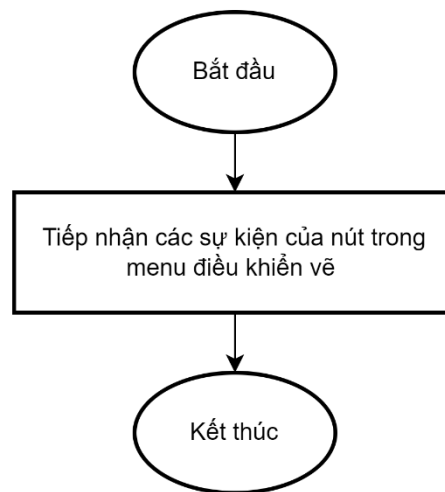


Hình 1-7 Lưu đồ nút "Lưu đồ thị"

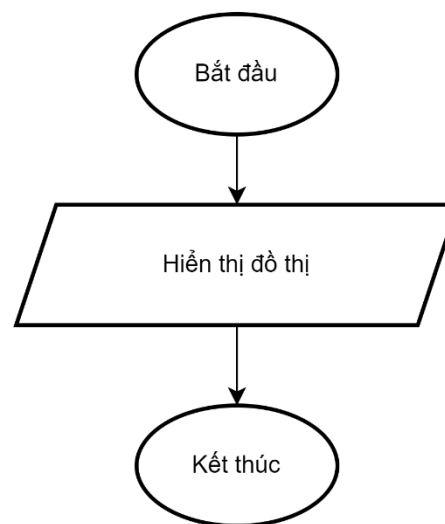


Hình 1-8 Lưu đồ nút "Tạo đồ thị mới"

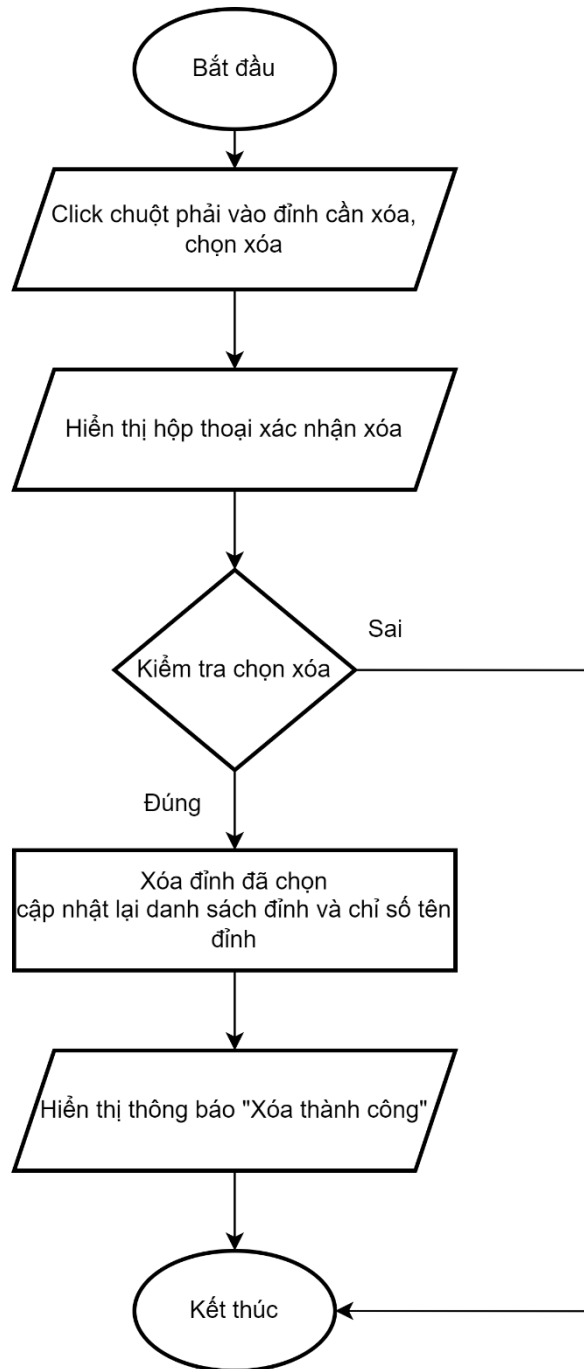
3.3.2. Bảng vẽ đồ thị



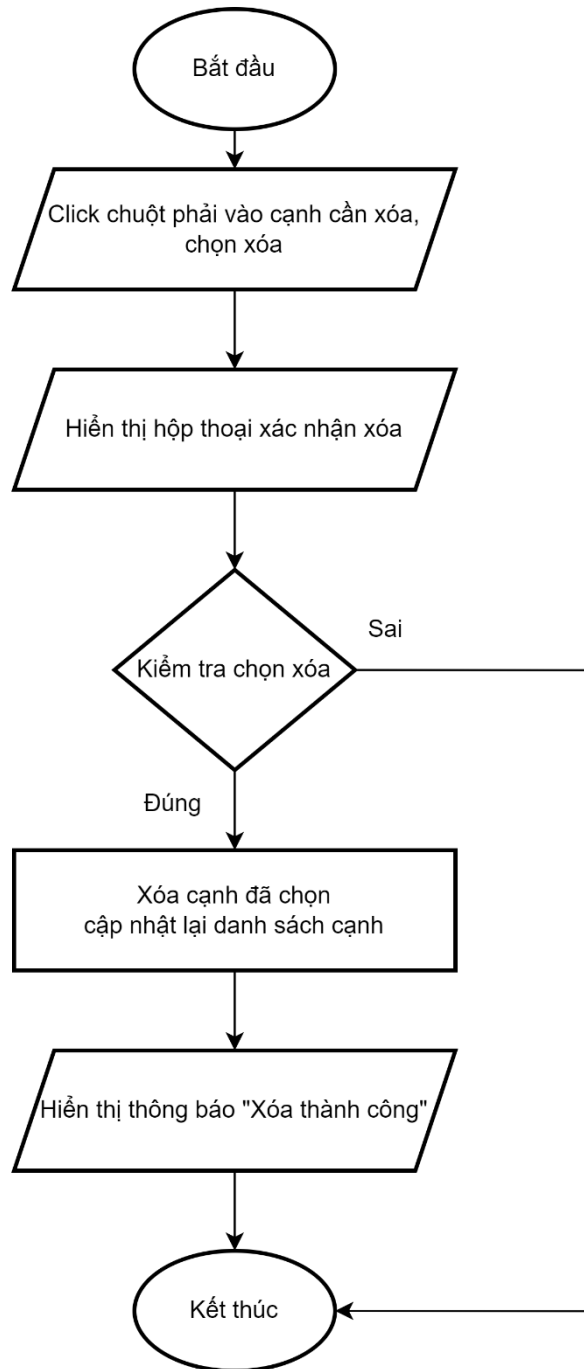
Hình 2-1 Lưu đồ xử lý Vẽ đồ thị



Hình 2-2 Lưu đồ xử lý Hiển thị đồ thị

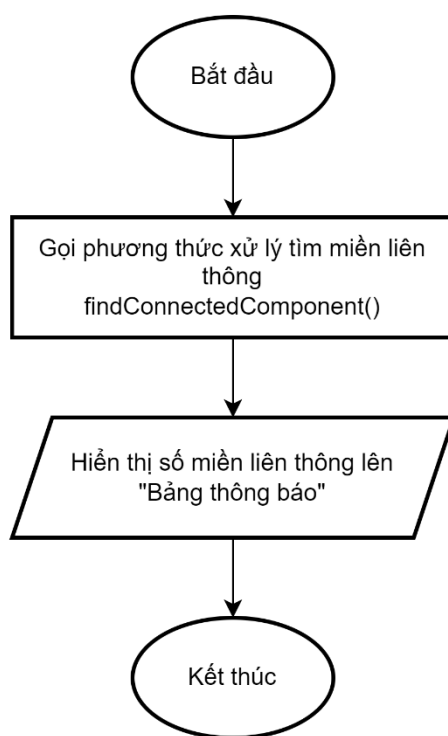


Hình 2-3 Lưu đồ thao tác "Xóa đỉnh"

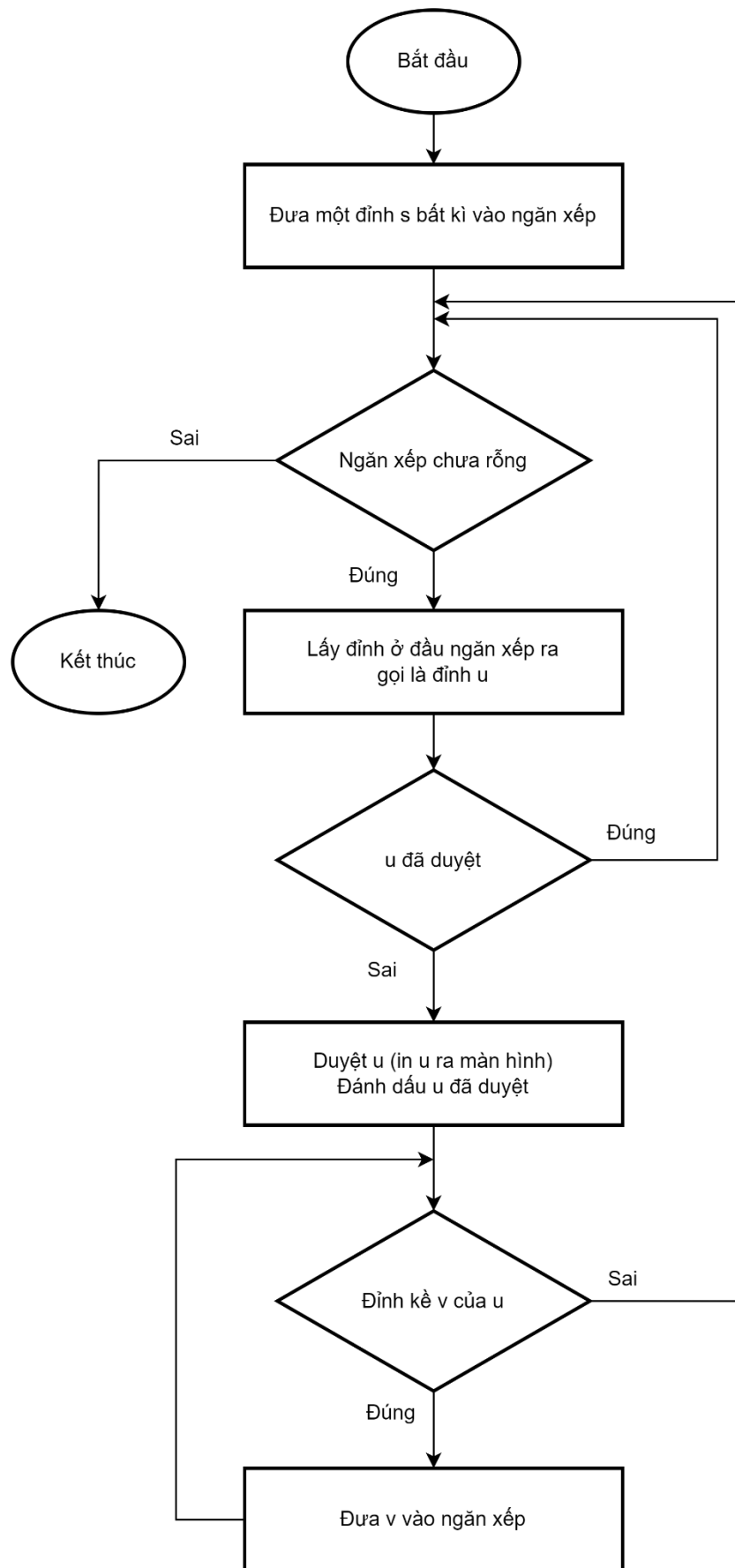


Hình 2-4 Lưu đồ thao tác "Xóa cạnh"

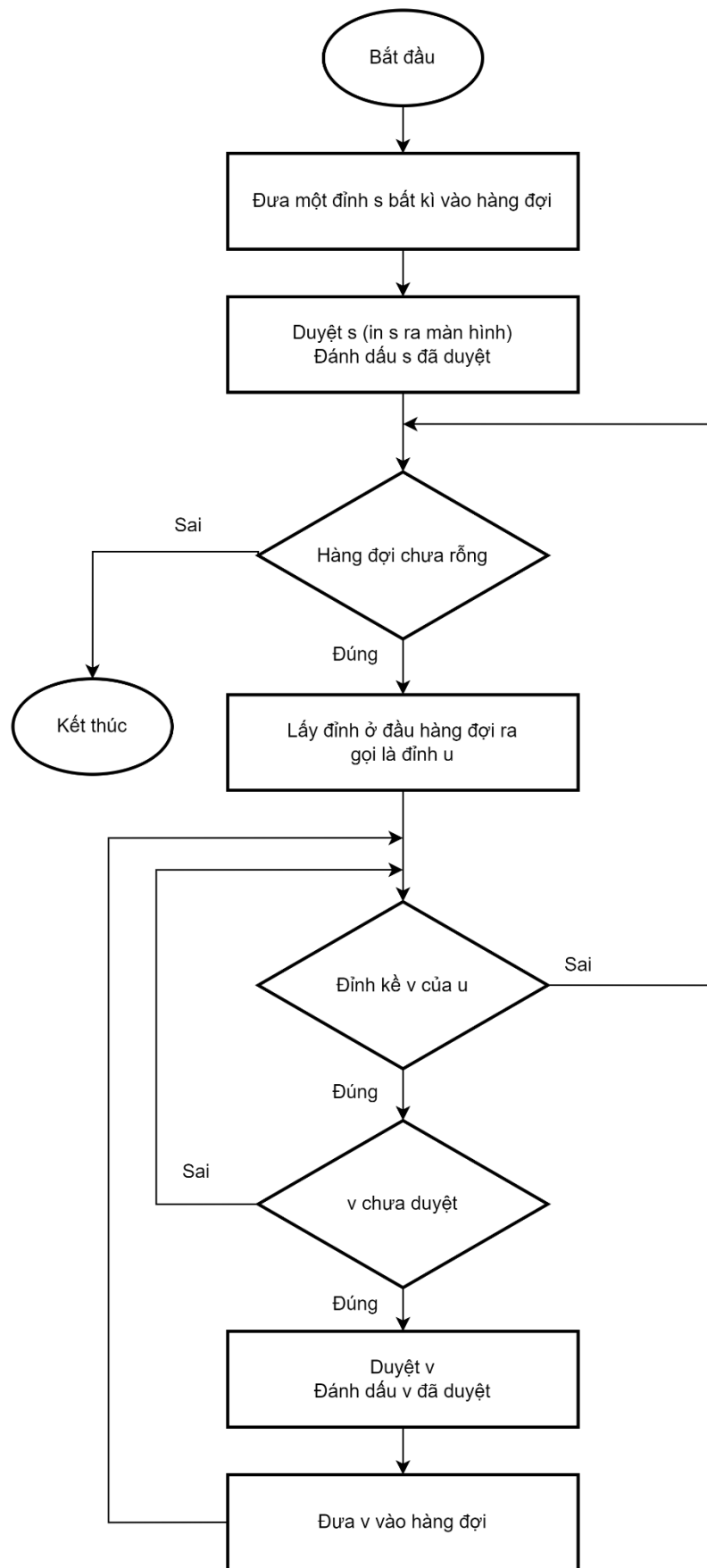
3.3.3. Số miền liên thông



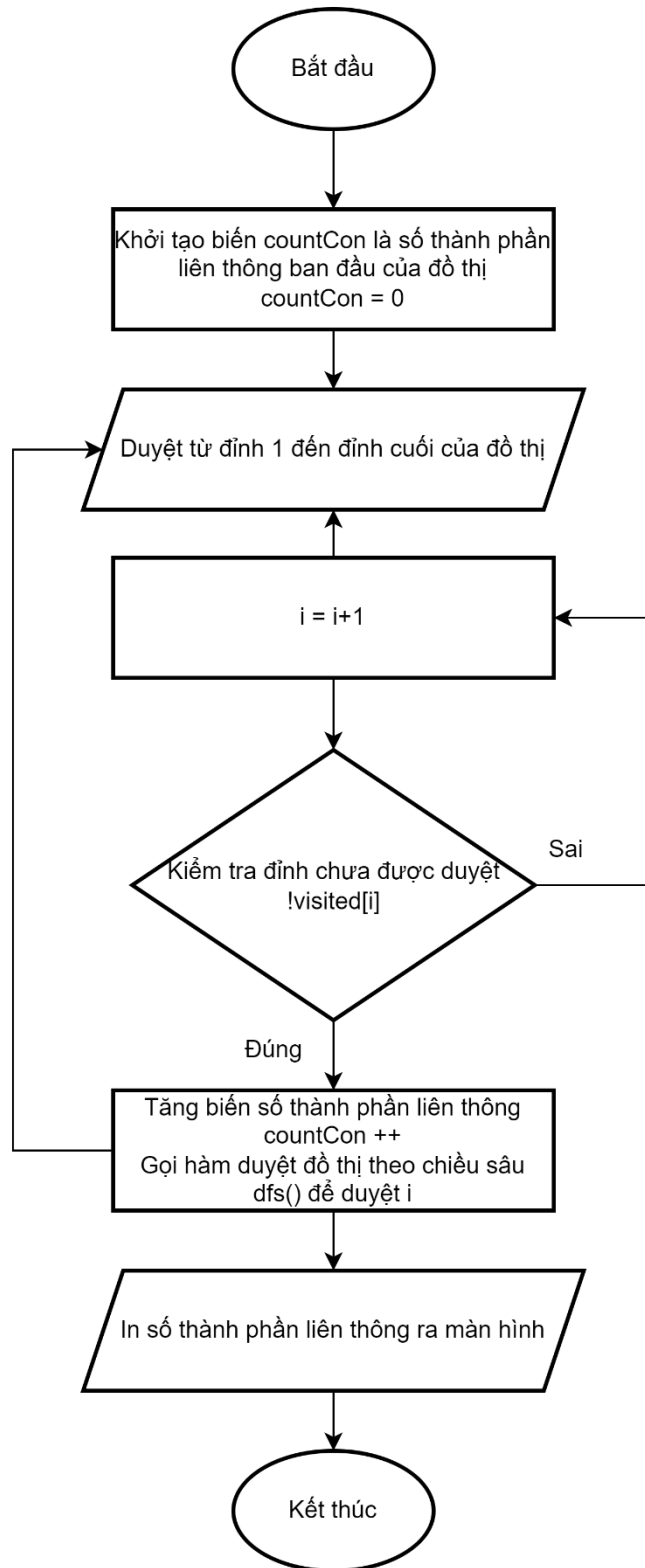
Hình 3-1 Lưu đồ nút “Số miền liên thông”



Hình 3-2 Lưu đồ giải thuật Duyệt đồ thị theo chiều sâu DFS của hàm dfs()

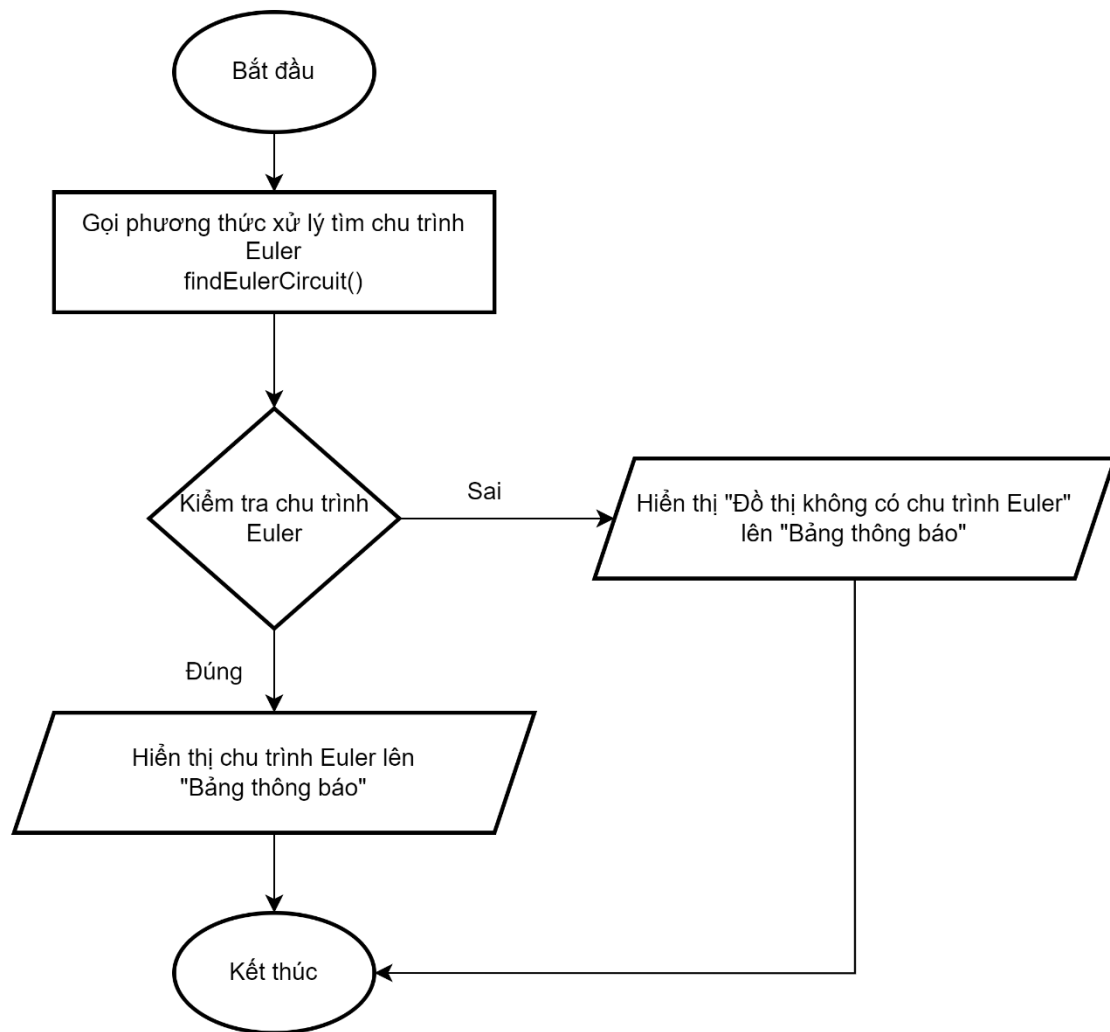


Hình 3-3 Lưu đồ giải thuật Duyệt đồ thị theo chiều rộng BFS của hàm bfs()

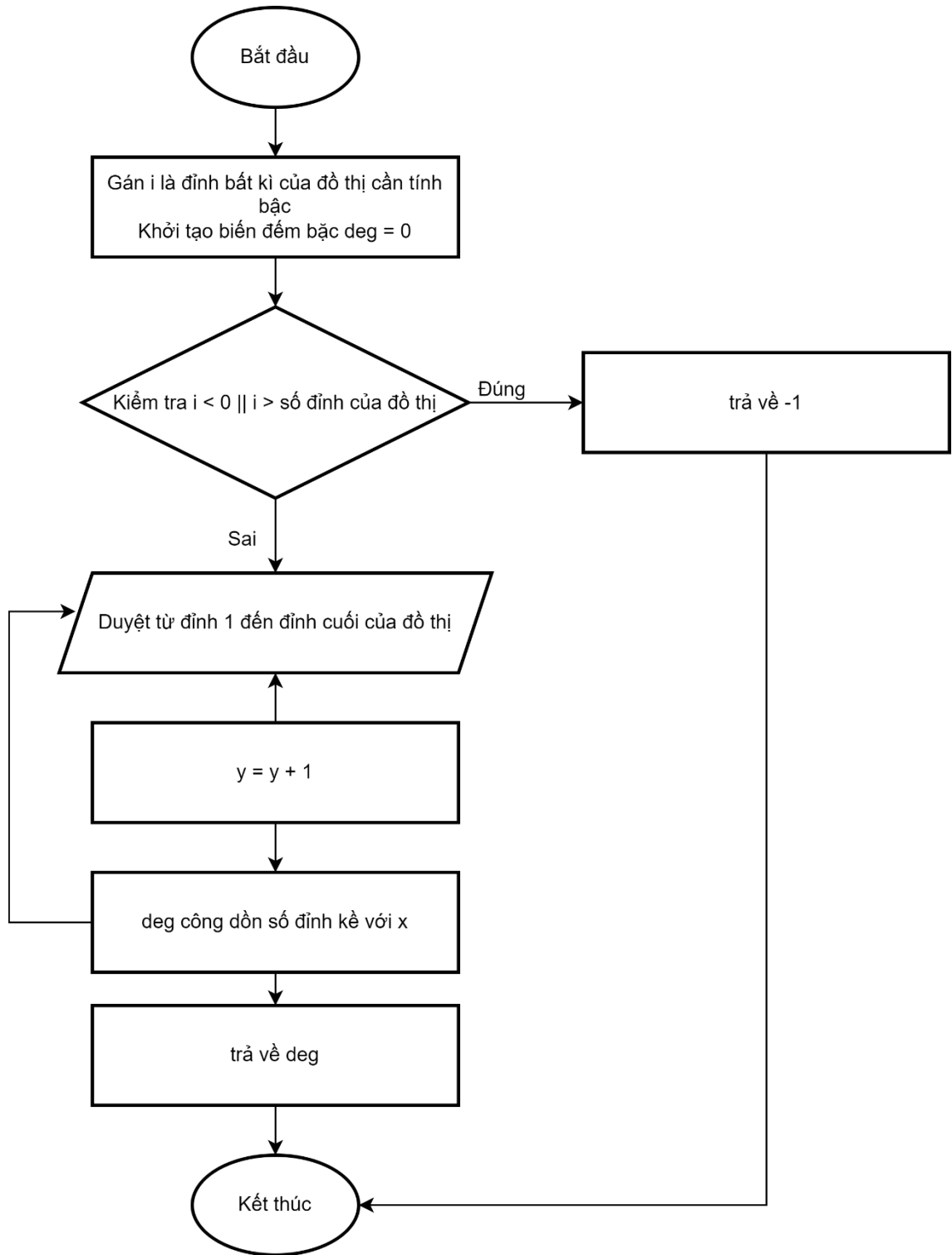


Hình 3-4 Lưu đồ giải thuật Kiểm tra số miền liên thông của hàm `findConnectedComponent()`

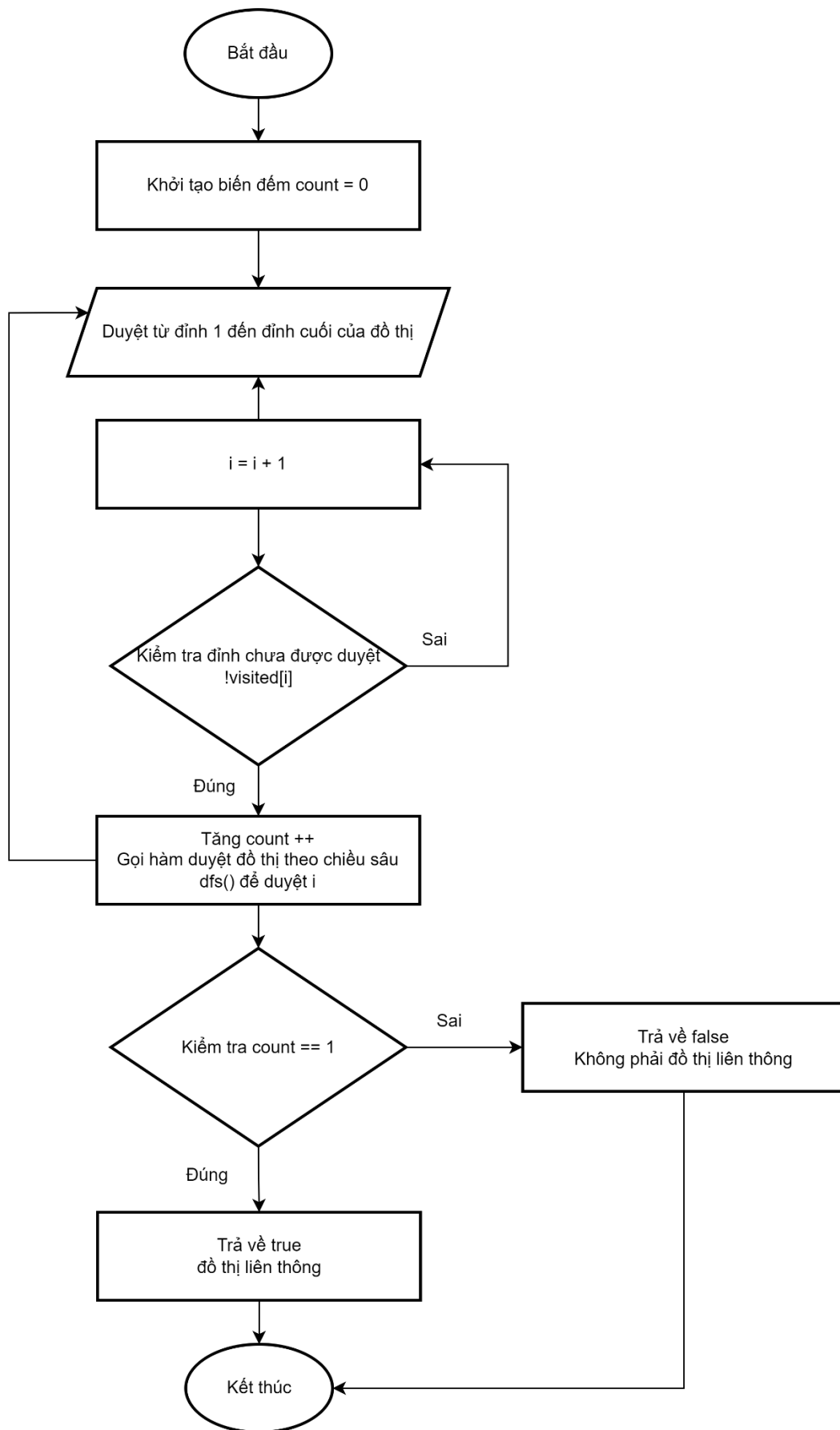
3.3.4. Chu trình Euler



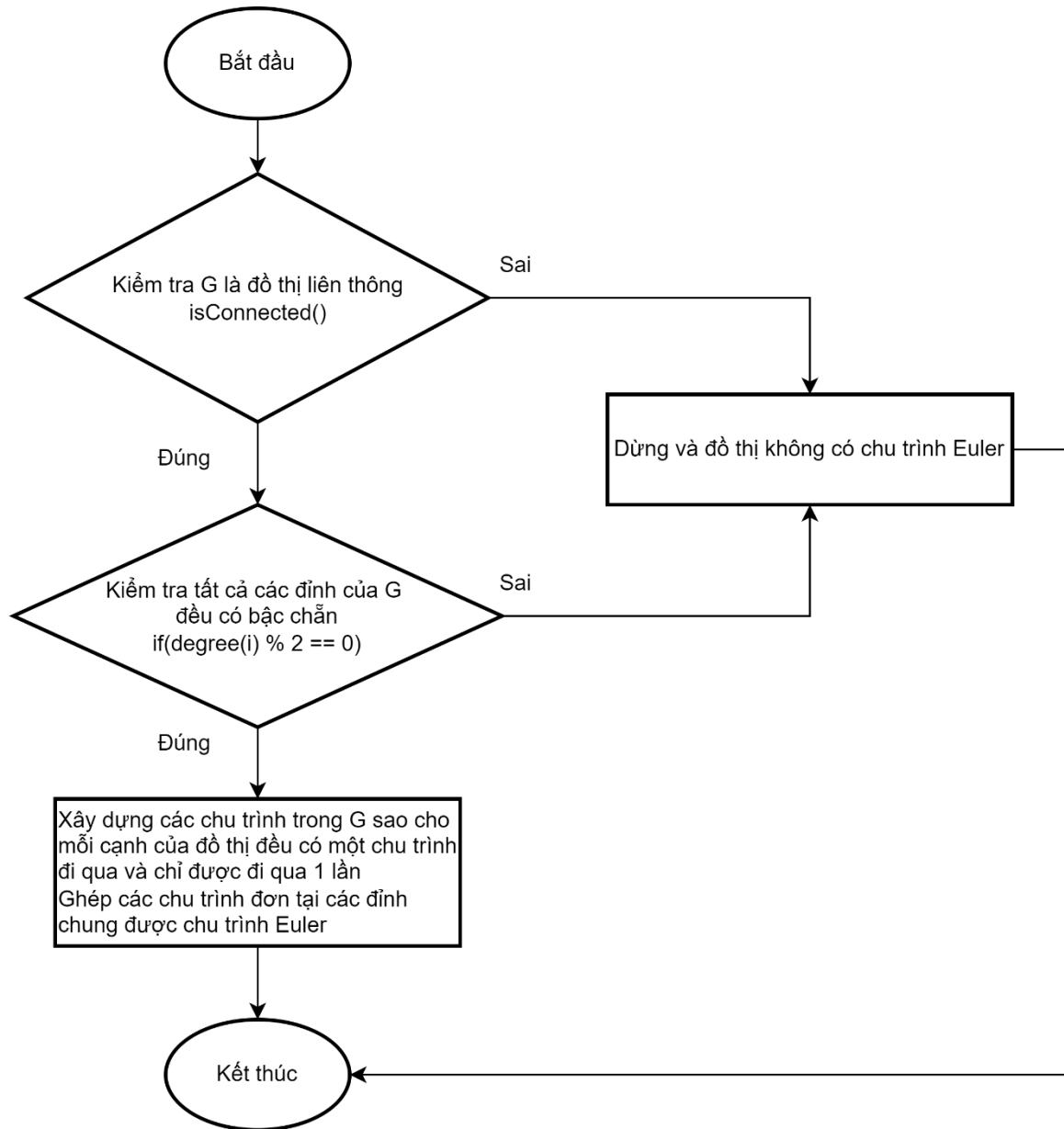
Hình 4-1 Lưu đồ nút "Chu trình Euler"



Hình 4-2 Lưu đồ giải thuật của hàm Tính bậc degree()

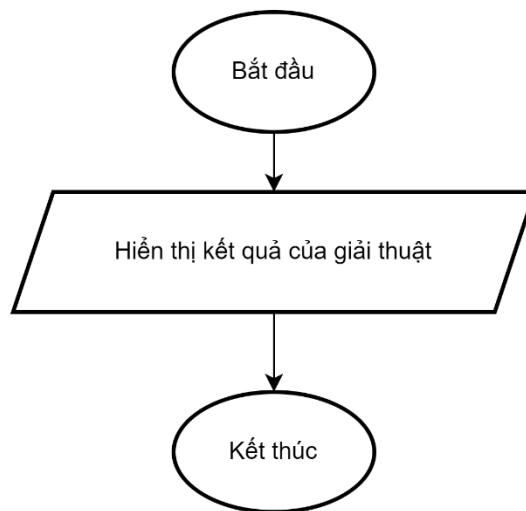


Hình 4-3 Lưu đồ giải thuật của hàm Kiểm tra liên thông isConnected()



Hình 4-4 Lưu đồ giải thuật của hàm Tìm chu trình Euler *findEulerCircuit()*

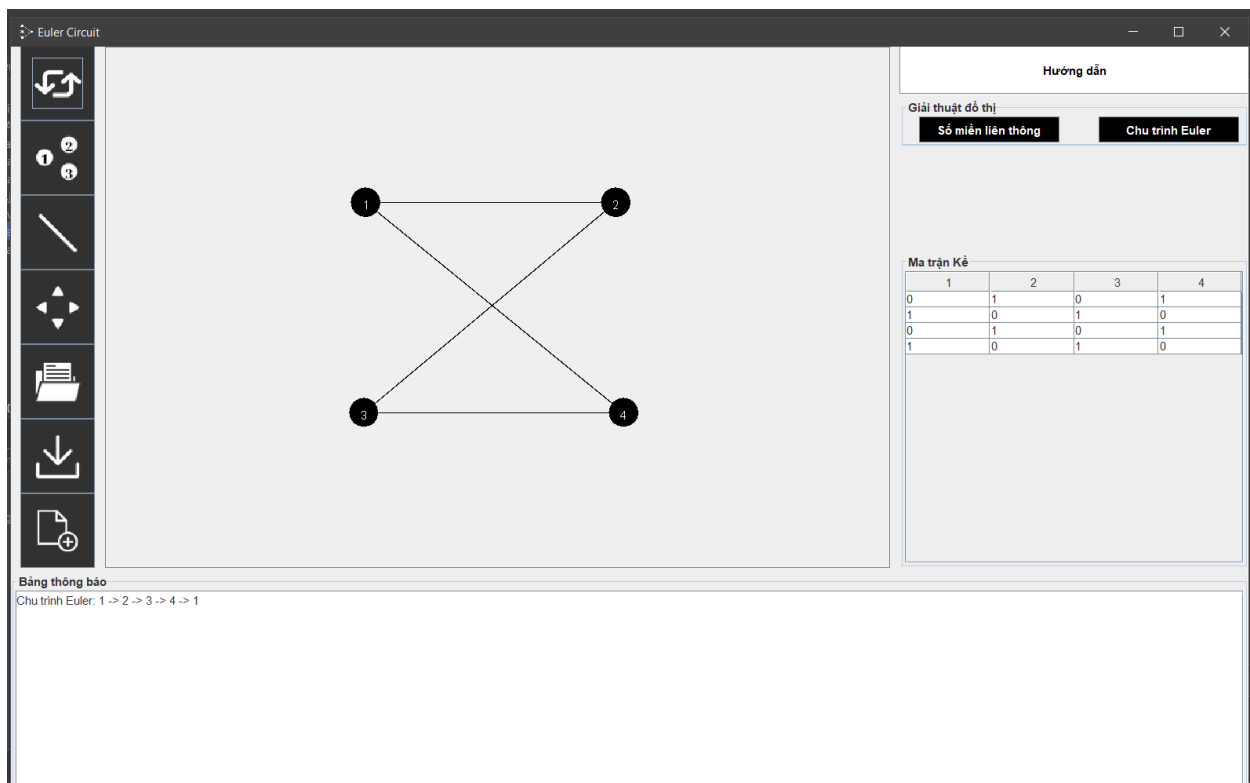
3.3.5. Bảng thông báo



Hình 5-1 Lưu đồ xử lý Hiện thị bảng thông báo

IV. GIỚI THIỆU CHƯƠNG TRÌNH

Giới thiệu về các giao diện của chương trình Euler



Hình 5-1 Giao diện phần mềm "Euler Circuit"

3.4.1. Menu điều khiển vẽ đồ thị

Giao diện bao gồm các nút:

- Nút "Cập nhật đồ thị": Cập nhật lại đồ thị khi có thay đổi về đỉnh, cung của đồ thị.
- Nút "Vẽ đỉnh": Vẽ các đỉnh của đồ thị.

- Nút “Vẽ cạnh”: Vẽ các cạnh nối giữa các đỉnh của đồ thị.
- Nút “Di chuyển đỉnh”: Di chuyển các đỉnh đến vị trí mới để đồ thị trông dễ nhìn.
- Nút “Mở đồ thị”: Mở file đồ thị đã lưu.
- Nút “Lưu đồ thị”: Lưu file đồ thị đã vẽ.
- Nút “Tạo đồ thị mới”: Tạo trang vẽ đồ thị mới.

3.4.2. Bảng vẽ đồ thị

Sau khi chọn nút trên menu điều khiển vẽ đồ thị và thực hiện vẽ vào bảng vẽ đồ thị. Bảng sẽ cho phép thao tác click, kéo thả chuột và hiển thị đồ thị khi vẽ.

3.4.3. Giải thuật của đồ thị

- Nút “Số miền liên thông”: Cho biết số miền liên thông của đồ thị.
- Nút “Chu trình Euler”: Tìm chu trình Euler trong đồ thị.
- Hiển thị ma trận kề trong bảng.

3.4.4. Bảng thông báo

Hiển thị số miền liên thông hoặc chu trình Euler tùy theo lựa chọn của người dùng.

3.4.5. Hướng dẫn sử dụng

- Nút “Hướng dẫn”: Hiển thị hướng dẫn sử dụng, công dụng các nút trong phần mềm
- Hướng dẫn sử dụng cơ bản: Sau khi mở phần mềm
 - + Nhấn nút “Vẽ đỉnh” click chuột vào bảng vẽ đồ thị để vẽ đỉnh. Tên đỉnh bắt đầu từ 1 và tăng dần.
 - + Nhấn nút “Vẽ cạnh” kéo thả chuột giữa hai đỉnh cần đường đi trong đồ thị.
 - + Nhấn nút “Di chuyển đỉnh”, chọn đỉnh và kéo thả đến vị trí mong muốn.
 - + Xóa đỉnh: Click chuột phải vào đỉnh cần xóa, chọn xóa và nhấn xác nhận để xóa đỉnh.
 - + Xóa cạnh: Click chuột phải vào cạnh cần xóa, chọn xóa và nhấn xác nhận để xóa cạnh.
 - + Nhấn nút “Cập nhật đồ thị” để cập nhật lại những thao tác vừa mới thay đổi.
 - + Nhấn nút “Tạo đồ thị mới” nếu muốn tạo lại bảng trống để vẽ.
 - + Nhấn nút “Lưu đồ thị” và chọn nơi lưu trữ để lưu đồ thị vừa vẽ.
 - + Nhấn nút “Mở đồ thị” và chọn file đồ thị cần mở.

CHƯƠNG 4. KẾT LUẬN – ĐÁNH GIÁ

I. KẾT QUẢ ĐẠT ĐƯỢC

Tạo được phần mềm có giao diện đồ họa cho phép vẽ đồ thị vô hướng bằng cách nhấp chuột và lưu đồ thị.

II. THU HOẠCH

4.2.1. Về mặt chuyên môn

Ôn tập lại các lý thuyết về đồ thị vô hướng, miền liên thông, BFS, DFS, và tìm hiểu về Euler. Vận dụng được các lớp trong Java để tạo ra phần mềm vẽ đồ thị trực quan.

4.2.2. Về mặt kinh nghiệm

Biết cách bố trí các nút, panel điều khiển vào vị trí phù hợp và cách bắt sự kiện trong Java, cũng như lựa chọn màu sắc hài hòa cho phần mềm.

Biết cách tạo một tài liệu báo cáo theo mẫu.

III. HẠN CHẾ - NGUYÊN NHÂN

Chưa vận dụng giải thuật tìm chu trình Euler và kiểm tra số miền liên thông của đồ thị lên giao diện đồ họa vì chỉ nắm các giải thuật về mặt lý thuyết. Chưa vận dụng tốt thời gian một phần do trì hoãn, một phần do sắp xếp thời gian học không hợp lý, các môn khác tiêu tốn nhiều thời gian hơn so với dự kiến.

IV. HƯỚNG PHÁT TRIỂN

Thêm chức năng vẽ đồ thị có hướng và các thuật toán liên quan.

Thêm chức năng mô phỏng duyệt theo chiều sâu và chiều rộng lên giao diện đồ họa.

Thêm chức năng mở đồ thị mẫu (đồ thị demo).

Thêm chức năng hỏi lưu đồ thị nếu thoát chương trình.

Thêm các phím điều hướng, hoàn tác cho phần mềm.

Làm cho giao diện đẹp hơn.

CHƯƠNG 5. PHẦN PHỤ LỤC

CHƯƠNG 6. TÀI LIỆU THAM KHẢO

[1] **Bộ môn Khoa học máy tính.** *Giáo trình Lý thuyết đồ thị.* Khoa Công nghệ Thông tin và Truyền thông ĐHCT.

[2] **PGS.TS. Phạm Nguyên Khang.** *Bài giảng Lý thuyết đồ thị.* Cần Thơ, 2018.

[3] **Geeksforgeeks.** *Eulerian path and circuit for undirected graph.* Geeksforgeeks, 2022.

<https://www.geeksforgeeks.org/eulerian-path-and-circuit/>

[4] **Manish Bhojasia.** *Java Program to Implement Euler Circuit Problem.* Sanfoundry.

<https://www.sanfoundry.com/java-program-implement-euler-circuit-problem/>