

KHOA CÔNG NGHỆ THÔNG TIN &TRUYỀN THÔNG

**BỘ MÔN CÔNG NGHỆ PHẦN MỀM**

### NIÊN LUẬN CƠ SỞ NGÀNH KTPM

# Giải thuật tìm chu trình Euler

Sinh viên thực hiện: Cán bộ hướng dẫn :

**Quách Vũ Linh Th.S Phan Huy Cường**

MSSV:B1505888

## HỌC KỲ 1, 2018-2019

# NHẬN XÉT ĐÁNH GIÁ CỦA GIÁO VIÊN HƯỚNG DẪN

*Cần Thơ, ngày.....tháng….năm 2018*

GVHD: Th.S Phan Huy Cường

# MỤC LỤC

[NIÊN LUẬN CƠ SỞ NGÀNH KTPM 0](#_Toc530753386)

[Giải thuật tìm chu trình Euler 0](#_Toc530753387)

[HỌC KỲ 1, 2018-2019 0](#_Toc530753388)

[NHẬN XÉT ĐÁNH GIÁ CỦA GIÁO VIÊN HƯỚNG DẪN 1](#_Toc530753389)

[MỤC LỤC 2](#_Toc530753390)

[CHƯƠNG 1. TỔNG QUAN 3](#_Toc530753391)

[1.1. Mô tả khái quát : 3](#_Toc530753392)

[1.2. Mục tiêu cần đạt được: 5](#_Toc530753393)

[1.3. Hướng giải quyết và thời gian thực hiện: 5](#_Toc530753394)

[CHƯƠNG 2. LÝ THUYẾT 6](#_Toc530753395)

[2.1. Các khái niệm sử dụng trong đề tài: 7](#_Toc530753396)

[2.2 Kết quả vận dụng lý thuyết vào đề tài: 7](#_Toc530753397)

[CHƯƠNG 3. KẾT QUẢ ỨNG DỤNG 8](#_Toc530753398)

[3.1. Phân tích yêu cầu bài toán và xây dựng cấu trúc dữ liệu cần thiết: 8](#_Toc530753399)

[3.2. Xây dựng giải thuật cho các thuật toán: 8](#_Toc530753400)

[3.3. Giới thiệu chương trình: 13](#_Toc530753401)

[3.4. Mô tả và cách sử dụng ứng dụng cụ thể: 14](#_Toc530753402)

[CHƯƠNG 4. KẾT LUẬN-ĐÁNH GIÁ 15](#_Toc530753403)

[4.1. Kết luận: 15](#_Toc530753404)

[4.2. Đánh giá: 16](#_Toc530753405)

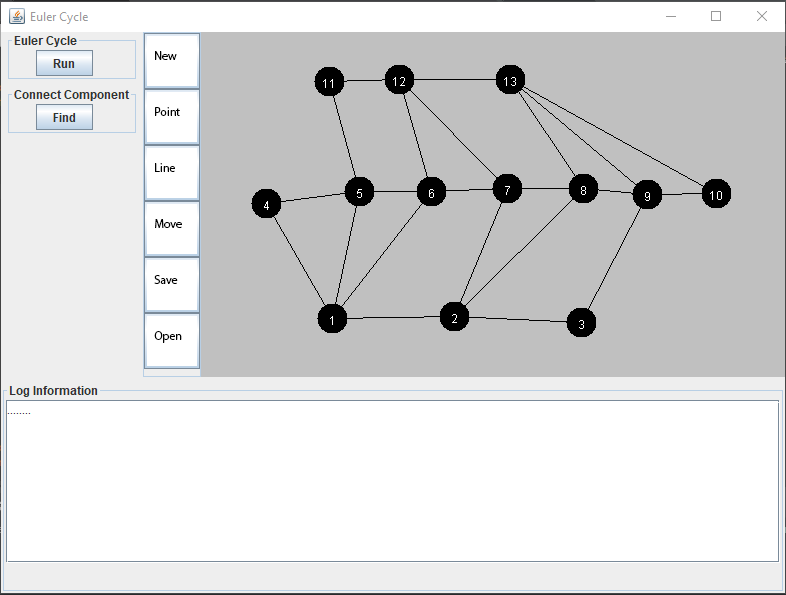
[Tài liệu tham khảo 16](#_Toc530753406)

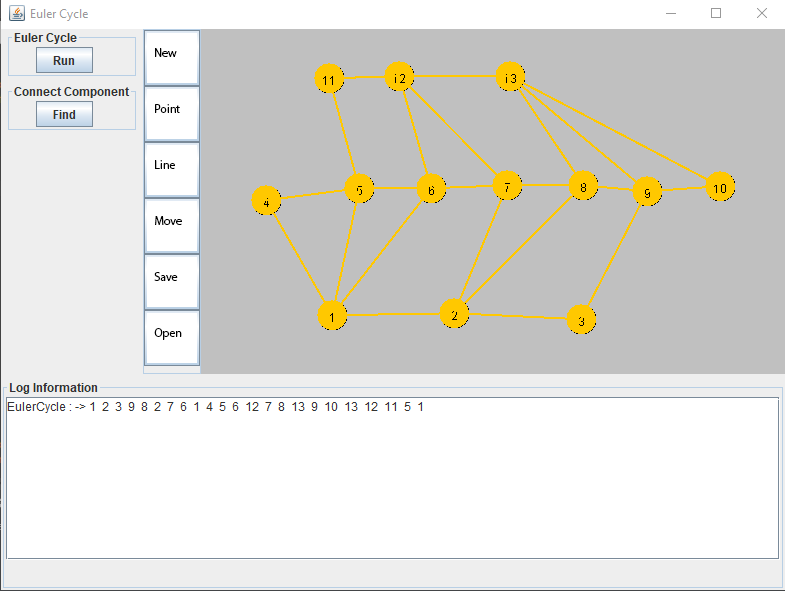
# CHƯƠNG 1. TỔNG QUAN

### 1.1. Mô tả khái quát :

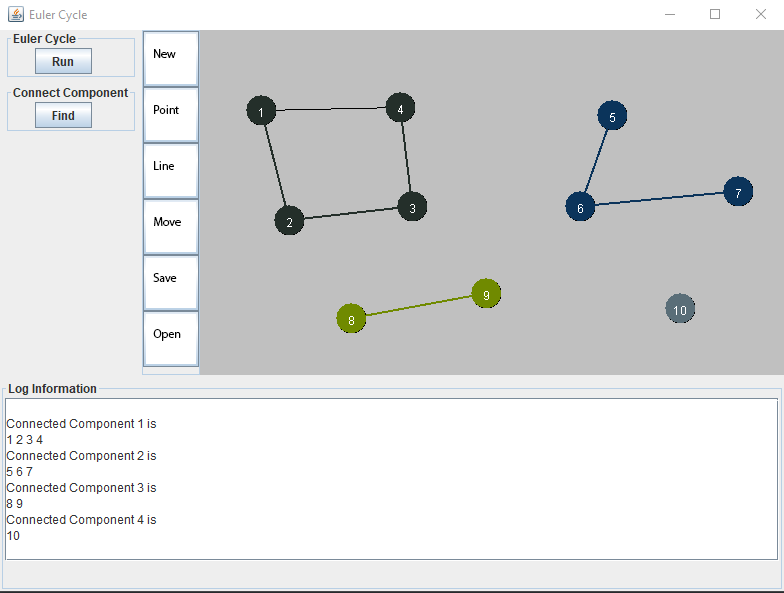
* Tên ứng dụng: giải thuật tìm chu trình Euler.
* Ngôn ngữ sử dụng: JAVA.
* Yêu cầu về hệ thống và các dịch vụ cần thiết cho ứng dụng: hệ điều hành windows 7 trở lên và các gói thư viện động hỗ trợ JAVA.
* Chu trình Euler (bài toán bảy cây cầu Euler),ý tưởng bài toán xuất phát từ nơi chốn cụ thể Konigsberg(Đức) có 2 vùng bị ngăn cách bởi một dòng sông và có 2 đảo ở giữa sông.Có 7 chiếc cầu nối những vùng này với nhau.
* Tác nhân lên hệ thống:
  + - Vì ứng dụng chỉ tương tác với một người dùng tại một thời điểm nên tác nhân chính là người dùng.
    - Chức năng chính của ứng dụng (bao gồm dữ liệu nhập-xuất đồ thị,kiểm tra số miền liên thông,chu trình và đường đi Euler của đồ thị).
* Phần mềm được viết : hỗ trợ cho giao diện đồ họa.
* Mô tả tổng quan về giao diện:
  + - Ở giao diện đầu tiên(ban đầu) của ứng dụng sẽ hiện thị các chức năng cơ bản sau:
      * Tạo mới đồ thị.
      * Tạo điểm mới cho đồ thị.
      * Tạo cạnh cho đồ thị.
      * Di chuyển điểm của đồ thị.
      * Lưu đồ thị.
      * Mở đồ thị đã lưu trước đó.
      * Sau khi người dùng tạo mới một đồ thị có thể click “Run” trong tiêu đề Euler Cycle để tìm xem đồ thị có chung trình hay đường đi Euler hay không, người dùng có thể click “Find” trong tiêu đề Connect Component để kiểm tra số miền liên thông.
* Cách sử dụng ứng dụng:

Ta định nghĩa các thành phần cơ bản của ứng dụng như Hình 1.

Hình 1.Giao diện giúp người dùng tạo mới một đồ thị.



Hình 2. Kết quả tìm chu trình Euler ở hình 1.

Hình 3. Kết quả tìm các miền liên thông.

Bắt đầu từ việc người dùng tạo mới một đồ thị thông qua các chức năng cơ bản trên giao diện. Mục tiêu của bài toán là người dùng sẽ dùng chuột để nhập đỉnh , cạnh để tạo thành một đồ thị mới (xem hình 1) ứng dụng sẽ cập nhật lại giao diện đồ thị. Sau đó người dùng có thể tìm chu trình và đường đi Euler, số miền liên thông của đồ thị.

Ứng dụng sẽ thực hiện các thuật toán để trả về kết quả phù hợp từ những yêu cầu của người dùng (xem hình 2 và 3) .

### 1.2. Mục tiêu cần đạt được:

* Xây dựng “ứng dụng tìm chu trình Euler” nhằm đáp ứng nhu cầu học tập cũng như tìm hiểu và tăng sự ham thích trong học tập.
* Tìm hiểu và triển khai thuật toán Fleury,tìm kiếm theo chiều rộng(BFS) và cách thuật toán kiểm tra để giải quyết bài toán.
* Tìm hiểu và phát triển ứng dụng bằng ngôn ngữ JAVA.
* Hoàn thành ứng dụng tìm chu trình Euler với giao diện đồ họa.

### 1.3. Hướng giải quyết và thời gian thực hiện:

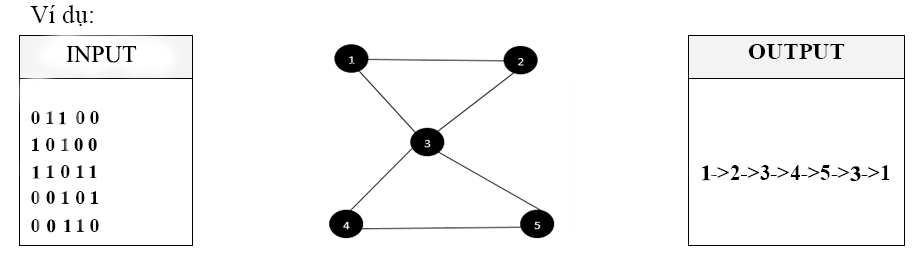
* Về lý thuyết:
  + Nắm vững định nghĩa đồ thị Euler để có thể tìm chu trình Euler trên đồ thị vô hướng.
  + Thực hiện các công việc tìm kiếm thông qua internet để tìm hiểu được thuật toán Fleury để áp dụng giải quyết bài toán tìm chu chình Euler.
  + Hoàn thành các đặc tả của ứng dụng từ đó lựa chọn các mô hình kiến trúc, xây dựng mã nguồn trở nên dễ dàng và phù hợp.Cuối cùng, hoàn thiện ứng dụng với các giao diện đồ họa.
* Về phân tích:
* Thống kê và khảo sát sự hiệu quả của thuật toán, phân tích thời gian thực hiện các giải thuật.
* Vẽ sơ đồ khối mô tả các giải thuật.
* Về phần mềm:
* Dùng phần mềm Eclipse Java phiên bản 4.8 để hỗ trợ lập trình, kiểm tra lỗi.( Tìm kiếm các giáo trình hoặc các bài giảng trên internet để học tập ngôn ngữ JAVA).
* Về kỹ thuật:
* Giao diện dễ sử dụng.
* Chương trình dễ mở rộng và phát triển.
* Sử dụng ngôn ngữ lập trình JAVA để lập trình phần mềm.
* Thời gian thực hiện:
* Đề tài được thực hiện trong 15 tuần:  
  • Tuần 1: Mô tả bài toán.  
  • Tuần 2: Xây dựng lưu đồ thuật toán chi tiết ứng dụng vào phần mềm.  
  • Tuần 3: Xây dựng lưu đồ thuật toán chi tiết để ứng dụng vào phần mềm.  
  • Tuần 4 đến 5: Xây dựng các chức năng phần mềm.  
  • Tuần 6 đến 8: Thiết kế chức năng.  
  • Tuần 9 đến 11: Thiết kế giao diện.  
  • Tuần 11đến 12: Kiểm thử sản phẩm.  
  • Tuần 13 đến 15: Viết báo cáo.

# CHƯƠNG 2. LÝ THUYẾT

Chu trình đơn trong đồ thị G đi qua mỗi cạnh của đồ thị đúng một lần được gọi là **chu trình Euler**. Đường đi đơn trong G đi qua mỗi cạnh của nó đúng một lần được gọi là **đường đi Euler**. Đồ thị được gọi là **đồ thị Euler** nếu nó có **chu trình Euler**. Đồ thị có **đường đi Euler** được gọi là nửa Euler.

Đối với bài toán tìm chu trình Euler có định lý sau : Một đồ thị vô hướng liên thông G=(V,E) là đồ thị Euler khi và chỉ khi mọi đỉnh của G đều là bậc chẵn.

Sau khi người dùng đã hoàn tất việc tạo mới một đồ thị thông qua chức năng tạo điểm và cạnh, ứng dụng sẽ nhận vào một ma trận kề.Người dùng click “**Run**” kết quả ứng dụng sẽ kiểm tra mọi đỉnh của đồ thị có là bậc chẵn và đồ thị có liên thông hay không bằng thuật toán BFS sau đó dùng thuật toán Fleury để kiểm tra đồ thị có chu trình hoặc đường đi Euler hay không.



Hình 4: Mô phỏng dữ liệu đầu vào và đầu ra của bài toán tìm chu trình Euler.

### 2.1. Các khái niệm sử dụng trong đề tài:

* **Tìm kiếm theo chiều rộng:**

Tìm kiếm theo chiều rộng (**BFS-Breadth First Search**) là một thuật toán tìm kiếm trong đồ thị trong đó việc tìm kiếm chỉ bao gồm 2 thao tác: (a) cho trước một đỉnh của đồ thị; (b) thêm các đỉnh kề với đỉnh vừa cho vào danh sách có thể hướng tới tiếp theo.Thuật toán BFS bắt đầu từ đỉnh gốc và lần lượt nhìn các đỉnh kề với đỉnh gốc. Sau đó, với mỗi đỉnh trong số đó, thuật toán lại lần lượt nhìn trước các đỉnh kề với nó mà chưa được quan sát trước đó và lặp lại.

**Thuật toán Fleury:**

Ta có thể vạch được 1 chu trình Euler trong đồ thị liên thông (G) có bậc mọi đỉnh là chẵn theo thuật toán Fleury sau:

Xuất phát từ 1 đỉnh bất kỳ của đồ thị (G) và tuân theo 2 quy tắc sau:

1. Mỗi khi đi qua một cạnh nào đó thì xóa nó đi, sau đó xóa đỉnh cô lập ( nếu có ).
2. Không bao giờ đi qua cầu trừ khi không còn cách đi nào khác .

### 2.2 Kết quả vận dụng lý thuyết vào đề tài:

* Thuật toán tìm kiếm theo chiều rộng được áp dụng vào đề tài để xác định số miền liên thông từ đó có thể xác định đồ thị có liên thông hay không.
* Thuật toán Fleury được áp dụng vào đề tài để xác định đồ thị có chu trình Euler hay không .

# CHƯƠNG 3. KẾT QUẢ ỨNG DỤNG

### 3.1. Phân tích yêu cầu bài toán và xây dựng cấu trúc dữ liệu cần thiết:

Chúng ta sẽ thiết kế chương trình tìm chu trình Euler để đáp ứng những yêu cầu sau:

* Thiết kế giao diện đồ họa cho người dùng nhập đồ thị:  
  Nhập điểm: điểm của đồ thị được tạo bằng sự kiện click chuột.

Nhập cạnh: cạnh của đồ thị được tạo bằng sự kiện kéo thả chuột.

cạnh của đồ thị được tạo bằng cách liên kiết hai điểm của đồ thị lại với nhau. Hệ thống sẽ xác định các sự kiện click,kéo,thả để thực hiện được yêu cầu.

* Tạo các điểm với vị trí ngẫu nhiên sao cho vị trí thuộc vào khu vực hiển thị trạng thái đồ thị.
* Di chuyển điểm : thay đổi vị trí của điểm bằng sự kiện click , kéo , thả chuột.
* Lưu đồ thị : lưu trạng thái đồ thị.
* Open đồ thị : mở đồ thị đã sao lưu trước đó.
* Kiểm tra số miền liên thông.
* Tạo các cạnh bằng cách click chọn điểm bất đầu và kéo thả đến ví trí kết thúc.Hệ thống sẽ xác định các vị trí bất đầu và kết thúc có hợp lí (vị trí chuột phải đúng vị trí điểm) hay không, nếu không thì thao tác tạo cạnh không thành công.
* Giảm thời gian cũng như độ phức tạp trong quá trình khởi tạo đồ thị.
* Tìm cấu trúc lưu trữ các thông số trạng thái của mỗi đỉnh,cạnh trong quá trình khởi tạo đồ thị sẽ đóng vai trò rất quan trọng trong việc đưa đến phương pháp xây dựng thuật toán để giải quyết bài toán,giúp tăng tính tương tác cao giữa người dùng và ứng dụng.
* Để đáp ứng các yêu cầu trên với từng trường hợp chúng ta sẽ có những giải pháp phù hợp như sau:
  + Với mục tiêu thực hiện giải thuật và trả về kết quả chính xác nhất.Việc nhận dữ liệu từ người dùng và cấu trúc lưu trữ là rất quan trọng.Xây dựng các Class đảm nhiệm các thành phần cụ thể chẳng hạn như : Class myDraw sẽ hiển thị trạng thái của đồ thị, Class myPoint sẽ xác định cấu trúc điểm của đồ thị , Class myLine sẽ xác định cấu trúc cạnh của đồ thị , Class myData sẽ lưu trữ các cạnh,điểm của đồ thị.Quan trọng nhất là Class myEuler nơi thực hiện các giải thuật với dữ liệu nhận từ các Class khác.Việc chia tách để dễ dàng trong việc quản lí cũng như lập trình.
* Kiểu dữ liệu của bài toán được sử dụng là một ma trận có kích thích bằng với số đỉnh đồ thị. Và mỗi phần tử trên ma trận này là Point.  
  Việc biểu diễn ma trận này trong các ngôn ngữ lập trình cụ thể như JAVA bằng  
  cách sử khởi tạo một mảng động kiểu **ArrayList<myPoint>** có kích thước bằng với tổng số điểm của đồ thị. Việc truy cập từng phần tử trên ma trận này được gọi thông qua vị trí của từng điểm trong mảng động nói trên.

### 3.2. Xây dựng giải thuật cho các thuật toán:

Thuật toán tiềm kiếm theo chiều rộng (BFS):

Để thực hiện việc xác định số miền liên thông cũng như đồ thị có liên thông hay không chúng ta lựa chọn phương pháp tiềm kiếm theo chiều rộng.

Cách hoạt động của thuật toán:Thuật toán sử dụng một cấu trúc dữ liệu hàng đợi để lưu trữ thông tin trung gian thu được trong quá trình tìm kiếm.Đầu tiên từ đỉnh v nào đó chưa thăm,thăm v,cất tất cả các đỉnh u (chưa thăm) kề với v vào hàng đợi.Lấy từ hàng đợi một đỉnh u,thăm u,cất tất cả các đỉnh t (chưa thăm) kề với u vào hàng đợi…

Thuật toán lập lại việc thăm cho tới khi hàng đợi rỗng.

Nếu tại một đỉnh x nào đó , không còn đỉnh nào kề với x là chưa thăm thì quay trở lại tiếp tục tìm đỉnh kề chưa thăm của y (y là đỉnh trước khi đến x).

Nếu đồ thị không liên thông (số thành phần liên thông lớn hơn 1) chúng ta có thể tách chúng thành những đồ thị con liên thông.Điều này có nghĩa là trong phép duyệt đồ thị, số thành phần liên thông của nó bằng số lần gọi tới thủ tục BFS.Vì thế khi áp dụng thủ tục BFS có sửa đổi.

Thủ tục BFS có thể được sửa lại như sau:

Function BFS (int u)

khởi tạo hàng đợi rỗng

nạp u vào hàng đợi

**while** hàng đợi chưa rỗng **then**

lấy x từ hàng đợi

**for** tất cả các cạnh kề với x

**if** cạnh kề với x chưa xét **then**

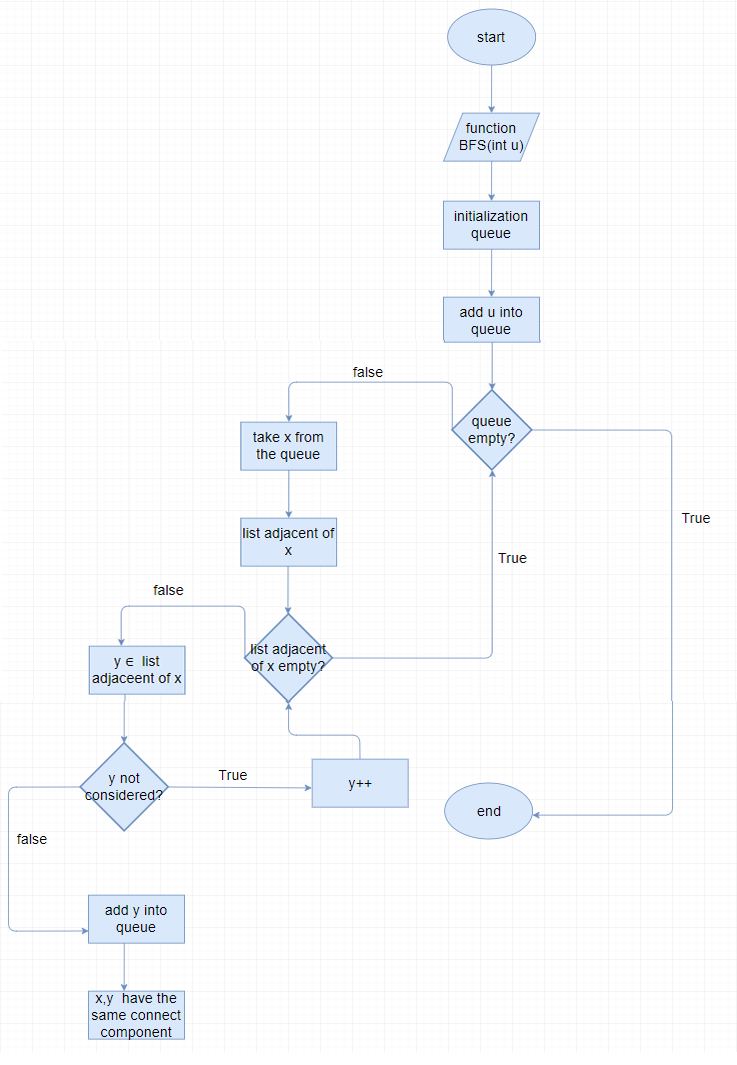
nạp cạnh kề với x vào hàng đợi

x và cạnh kề của nó có cùng thành phần liên thông

**end if**

end function

Đây là mã giả của thuật toán trên.



Hình 5 : Đây là lưu đồ của thuật toán BFS trên .

Thuật toán Fleury :

Ta có thể vạch được một chu trình Euler trong đồ thị liên thông (G) có bậc của mọi đỉnh là chẵn theo thuật toán Fleury.

Cách hoạt động của thuật toán : xuất phát từ một đỉnh u nào đó của G ta đi theo các cạnh của nó một cách tùy ý chỉ cần tuân thủ 2 qui tắc sau : xóa bỏ cạnh đã đi qua đồng thời xóa bỏ đỉnh cô lập tạo thành,ở mỗi bước ta chỉ đi qua cầu khi không còn cách nào khác.

Cầu là một cạnh nếu loại bỏ thì đồ thị sẽ không còn liên thông.

Đây là mã giả của thuật toán trên :

**Function** Try()

Khởi tạo stack rỗng

u = 1 /\* đỉnh nào đó của đồ thị \*/

nạp u vào stack

**while** stack chưa rỗng **then**

x = lay phan tu dau cua stack

**if** danh sách đỉnh kề của x chưa rỗng **then**

y = đỉnh đầu trong danh sách kề

nạp y vào stack

xóa cạnh x,y

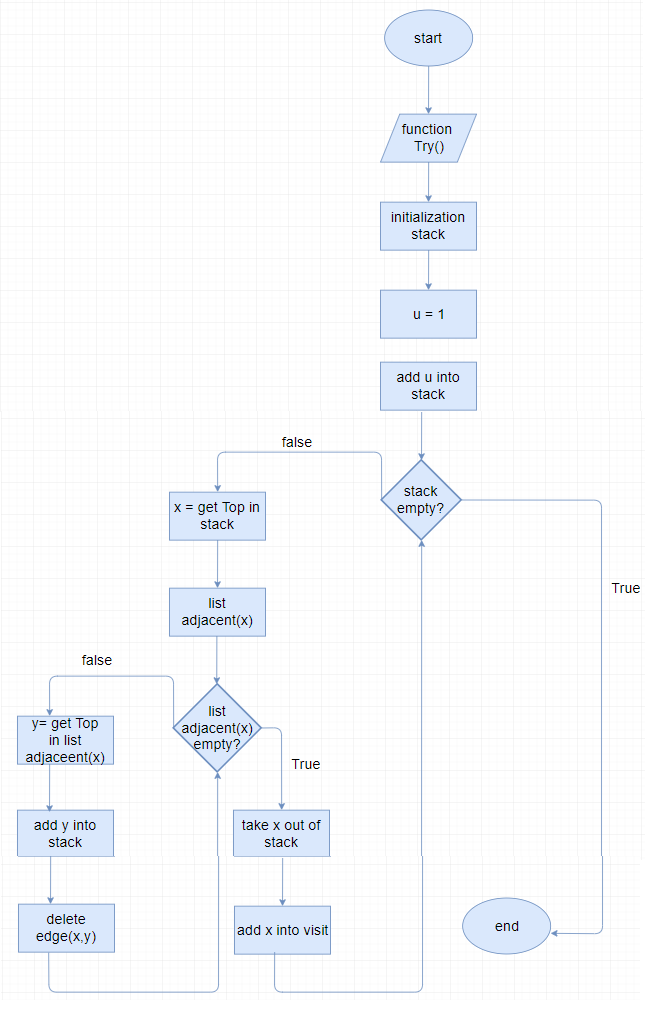
**else**

lấy x khỏi stack

nạp x vào mảng đã thăm

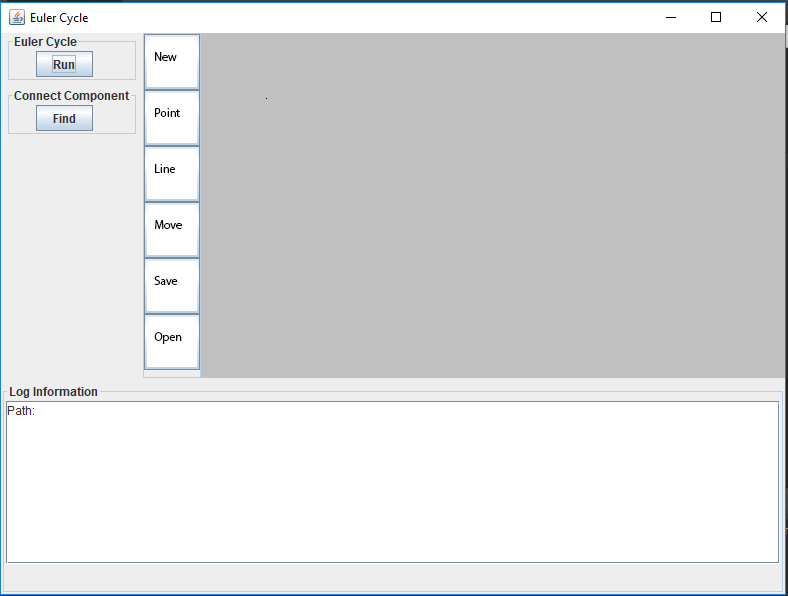
**end if**

**end function**



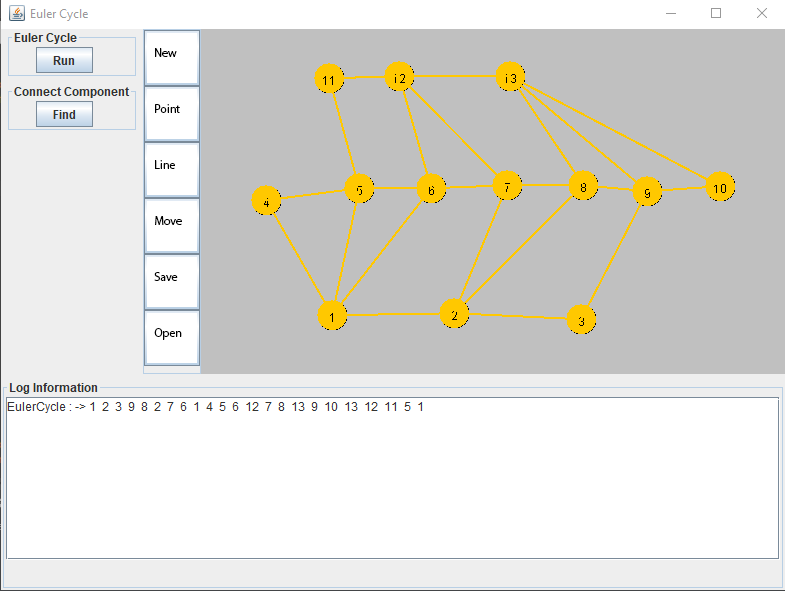
Hình 6 : Đây là lưu đồ của thuật toán Fleury.

### 3.3. Giới thiệu chương trình:



Hình 7. Giao diện chính khi gọi chương trình.

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên | Mô tả | Chức năng |
| 1 | New | Nút tạo mới đồ thị | Bất đầu tạo một đồ thị và reset lại toàn giao diện về trạng thái ban đầu . |
| 2 | Point | Nút tạo điểm đồ thị | Bất đầu tạo điểm mới cho đồ thị hoặc có thể di chuyển vị trí điểm đã tạo. |
| 3 | Line | Nút tạo đường thẳng đồ thị | Bất đầu tạo đường thẳng từ các điểm của đồ thị. |
| 4 | Move | Nút di chuyển điểm đồ thị | Di chuyển điểm đồ thị. |
| 5 | Save | Nút lưu đồ thị | Lưu lại trạng thái của đồ thị hiện tại. |
| 6 | Open | Nút mở đồ thị | Mở một đồ thị đã lưu trước đó. |
| 7 | Run | Nút kiểm tra đồ thị Euler | Kiểm tra đồ thị có chu trình hoặc đường đi Euler hay không. |
| 8 | Find | Nút kiểm tra miền liên thông đồ thị | Kiểm tra số miền liên thông của đồ thị. |
| 9 | Log | Khung hiển thị thông tin kết quả | Hiển thị các kết quả đạt được. |
| 10 | MyDraw | Panel hiển thị đồ thị  ( Khung xám góc phải) | Hiển thị trạng thái đồ thị ( danh sách các điểm , đường thẳng). |



Hình 8. Đây là giao diện sau khi tìm chu trình Euler.

### 3.4. Mô tả và cách sử dụng ứng dụng cụ thể:

Khi chạy trương chình chính: Người dùng sẽ được tương tác với giao diện hệ thống đã lập trình

1.Bất đầu từ việc tạo mới một đồ thị người dùng chọn “new”.Hệ thống sẽ reset lại tất cả các khung hiển thị cũng như các dữ liệu đầu vào trong hệ thống.

Đồ thị là sự hình thành của điểm và cạnh vì thế để tạo điểm cho đồ thị người dùng chọn “Point” sau đó tại khung ghi nhận dữ liệu đầu vào (gốc phải giao diện),người dùng có thể click vào bất kì vị trí nào, một điểm mới sẽ được tạo ra tại vị trí click.

Tiếp theo để tạo cạnh cho đồ thị người dùng chọn “Line” sau đó click vào đúng vị trí điểm bất đầu và kéo đến điểm kết thúc khi đó quá trình kết thúc và cạnh mới được tạo thành, nếu vị trí click và thả chuột không đúng vị trí điểm thì thao tác sẽ không thành công.

Để di chuyển các điểm trong đồ thị người dùng chọn “Move” sau đó click vào đúng vị trí điểm để bất đầu thay đổi vị trí điểm.

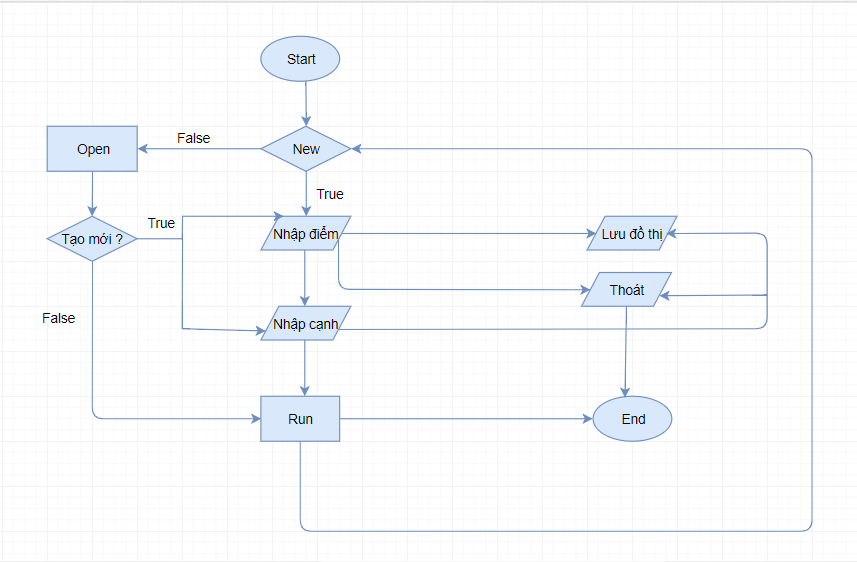
Lưu ý : việc tạo mới đồ thị cũng có thể bất đầu từ việc mở một thị đã lưu trước đó và có thể tạo điểm, đường thẳng mới để tạo thành một trạng thái đồ thị mới.

Các thao tác tạo điểm , đường thẳng , di chuyển điểm có thể sử dụng cả chuột phải và chuột trái .

2. Thực hiện các yêu cầu: sau khi hoàn thành việc khởi tạo đồ thị người dùng có thể kiểm tra đồ thị có chu trình hoặc đường đi Euler hay không bằng cách click “Run” trong thành phần Euler Cycle (gốc trên bên trái giao diện) , người dùng cũng có thể kiểm tra số miền liên thông của đồ thị bằng cách click “Find” trong thành phần Connect Component hệ thống sẽ dùng thuật toán để kiểm tra và trả về kết quả phù hợp.

3.Lưu đồ thị: sau khi hoàn thành việc khởi tạo đồ thị người dùng có thể lưu lại trạng thái hiện thời của đồ thị vào một thư mục trên máy tính bằng cách chọn “Save” và chọn địa chỉ lưu .

4.Mở đồ thị : để sử dụng lại các trạng thái đồ thị trước đó người dùng chọn “Open” và chọn địa chỉ đã lưu đồ thị .Hệ thống sẽ nhận lại dữ liệu đầu vào và hiện thị trạng thái đồ thị,người dùng có thể tiếp tục các thao tác tiếp theo.



Hình 9. Đây là lưu đồ của ứng dụng.

# CHƯƠNG 4. KẾT LUẬN-ĐÁNH GIÁ

### 4.1. Kết luận:

Sau thời gian thực hiện đề tài,em đã đạt được một số kết quả nhất định:  
Về phía bản thân:  
• Nắm được các kiến thức về thuật toán Fleury , tiềm kiếm theo chiều sâu ( BFS) và các khái niệm về đồ thị.

• Mở rộng sự hiểu biết về ngôn ngữ JAVA.

• Có thêm kinh nghiệm trong việc viết báo cáo.  
• Khả năng tìm kiếm và giải quyết vấn đề.  
Về phía phần mềm đã thực hiện được:  
• Giao diện đồ họa cho phép người dùng tướng tác tốt với ứng dụng (hiển thị các chức năng cơ bản : tạo mới đồ thị , tạo điểm, tạo cạnh, lưu đồ thị…).  
• Có thực hiện vẽ lại đồ thị làm nổi bật kết quả thực hiện cho người dùng dễ dàng quan sát.

• Nắm vững cấu trúc giao diện đồ họa trong ngôn ngữ JAVA.

### 4.2. Đánh giá:

Ưu điểm của phần mềm:  
• Phần mềm mô phỏng tốt và hoàn thiện về mặt giao diện.  
• Phần mềm cho kết quả chính xác theo các thuật toán đã cài đặt.  
• Giao diện thân thiện, cung cấp hoàn chỉnh các chức năng cần thiết cho việc tương tác với ứng dụng.

• Phần mềm có thể phóng to thu nhỏ được.

Khuyết điểm của phần mềm:

• Chưa thể mô đun hóa toàn bộ ứng dụng.

• Dòng lệnh nhiều rời rạc khiến cho việc phát triển mở rộng ứng dụng gặp hạn chế và khó khăn về sau.

Hướng phát triển:

• Mô đun hoàn toàn bộ phần mềm.

• Phát triển ứng dụng web theo kiểu mẫu này giúp cho nhiều có sử dụng trực tuyến.

# Tài liệu tham khảo

[1]Wikipedia:

<https://vi.wikipedia.org/wiki/%C4%90%E1%BB%93_th%E1%BB%8B_Euler>

https://vi.wikipedia.org/wiki/Th%C3%A0nh\_ph%E1%BA%A7n\_li%C3%AAn\_th%C3%B4ng

[2]Tài liệu về ngôn ngữ JAVA:

https://www.tutorialspoint.com/java/java\_documentation.htm

[3]Các thuật toán ứng dụng trong phần mềm:

<https://www.geeksforgeeks.org/backtracking-algorithms/>

[4]Tài liệu về lý thuyết đồ thị của thầy Ngô Hữu Phúc:

<http://fit.lqdtu.edu.vn/files/FileMonHoc/Ly%20Thuyet%20do%20thi.pdf>