

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN



ĐỒ ÁN MÔN HỌC

LÝ THUYẾT ĐỒ THỊ

NHÓM: P14

GVHD: Đặng Trần Minh Hậu

GVHD: Nguyễn Ngọc Thảo

TP Hồ Chí Minh, tháng 4 năm 2024

MỤC LỤC

MỤC LỤC	1
Thành viên nhóm	2
Bảng phân công công việc.....	3
Chương 1 Tổng quan lý thuyết và hướng dẫn sử dụng	4
1.1 Tổng quan lý thuyết	4
1.1.1 Tổng quan	4
1.1.2 Mục tiêu:	4
1.2 Hướng dẫn sử dụng	4
1.2.1 Giới thiệu chức năng và phân bố dự án.....	4
1.2.2 Hướng dẫn thực hiện dự án	4
Chương 2 Tính toán lý thuyết	6
2.1 Yêu cầu 1 Nhận diện một số dạng đồ thị đặc biệt.....	6
2.1.1 Kiểm tra đầu vào	6
2.1.2 Thuật toán.....	6
2.2 Yêu cầu 2: Xác định thành phần liên thông mạnh	6
2.2.1 Xác định tính liên thông của đồ thị.....	7
2.2.2 Xác định thành phần liên thông mạnh của đồ thị có hướng.....	7
2.3 Yêu cầu 3: Tìm cây khung lớn nhất	7
2.3.1 Tìm cây khung lớn nhất bằng thuật toán Prim:	7
2.3.2 Tìm cây khung lớn nhất bằng thuật toán Kruskal	7
2.4 Yêu cầu 4: Tìm đường đi ngắn nhất.....	8
2.4.1 Trình bày lý thuyết:	8
2.5 Yêu cầu 5: Tìm chu trình hoặc đường đi Euler.....	9
2.5.1 Kiểm tra đầu vào	9
2.5.2 Kiểm tra tính chất Euler của đồ thị	9
2.5.3 Tìm chu trình/đường đi Euler	9
Chương 3 Kết quả	11
3.1 Yêu cầu 1 Nhận diện một số dạng đồ thị đặc biệt.....	11
3.2 Yêu cầu 2: Xác định thành phần liên thông mạnh	14
3.3 Yêu cầu 3: Tìm cây khung lớn nhất	18
3.4 Yêu cầu 4: Tìm đường đi ngắn nhất.....	20
3.5 Yêu cầu 5: Tìm chu trình hoặc đường đi Euler.....	23
Chương 4 Tự đánh giá.....	26

Thành viên nhóm

Nhóm P14:

Nguyễn Quốc Bình	23880008
Nguyễn Tấn Lộc	23880039
Nguyễn Trung Nguyên	23880052
Khổng Minh Thu	23880080

Bảng phân công công việc

STT	Phân công	Người phụ trách
1	- Mô tả dữ liệu đầu vào - Yêu cầu 1: Nhận diện một số dạng đồ thị đặc biệt	Trung Nguyên
2	- Yêu cầu 2: Xác định thành phần liên thông mạnh - Yêu cầu 3: Tìm cây khung lớn nhất	Minh Thu
3	- Yêu cầu 4: Tìm đường đi ngắn nhất	Tấn Lộc
4	- Yêu cầu 5: Tìm chu trình hoặc đường đi Euler	Quốc Bình
5	- Kiểm tra logic và chức năng	Trung Nguyên+ Tấn Lộc
6	- Tổng hợp code	Quốc Bình
7	- Tổng hợp word + pdf + excel	Quốc Bình

Chương 1 Tổng quan lý thuyết và hướng dẫn sử dụng

1.1 Tổng quan lý thuyết

1.1.1 Tổng quan

- Phần mềm Visual Studio.
- Ngôn ngữ C# (.NET).
- Sử dụng giao diện console để thực hiện và thể hiện các yêu cầu.
- Các file đồ thị được lưu trong folder Requirement1 đến Requirement5 và được lưu trữ với định dạng txt.

1.1.2 Mục tiêu:

Thực hiện các yêu cầu:

- Yêu cầu 1: Nhận diện một số dạng đồ thị đặc biệt
- Yêu cầu 2: Xác định thành phần liên thông mạnh
- Yêu cầu 3: Tìm cây khung lớn nhất
- Yêu cầu 4: Tìm đường đi ngắn nhất
- Yêu cầu 5: Tìm chu trình hoặc đường đi Euler

1.2 Hướng dẫn sử dụng

1.2.1 Giới thiệu chức năng và phân bổ dự án

Bảng 1: Bảng phân bổ các class và chức năng

Class	Chức năng
Program.cs	Main class, code tổ chức các function giao tiếp với user.
Requirement1.cs - Requirement5.cs	Tổ chức, thực hiện giải thuật tương ứng với các yêu cầu từ 1 - 5
Validation.cs	Chứa các function kiểm tra đồ thị
AdjacencyList.cs	Khởi tạo và tổ chức các function của Danh sách kề
AdjacencyMatrix.cs	Khởi tạo và tổ chức các function của Ma trận kề

- **Danh sách kề**: Là một danh sách liên kết chứa các danh sách liên kết con, mỗi danh sách liên kết con sẽ có chỉ số (trong danh sách cha) thể hiện đỉnh đầu và chứa các Edge, với Edge là kiểu dữ liệu tự định nghĩa bao gồm đỉnh cuối (Destination) và trọng số (Weight).
- **Ma trận kề**: Là một ma trận 2 chiều biểu diễn đồ thị với các chỉ số (i, j) lần lượt thể hiện đỉnh đầu và đỉnh cuối của một cạnh và giá trị $W(i, j)$ là trọng số của cạnh tương ứng.

1.2.2 Hướng dẫn thực hiện dự án

Bước 1: Mở dự án với file Project.sln

Bước 2: Build and Run.

Bước 3: Với giao diện console

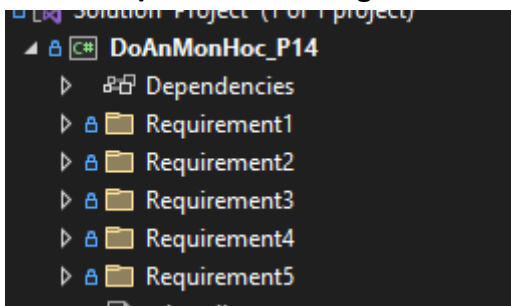
Nhập 1 – 5 để lựa chọn yêu cầu 1 đến yêu cầu 5 tương ứng

Bước 4: Sau khi lựa chọn yêu cầu thì danh sách các đồ thị có sẵn trong folder và lựa chọn đồ thị cần kiểm tra tương ứng.

```

ĐỒ ÁN MÔN HỌC NHÓM P14
-----
Vui lòng chọn yêu cầu
Nhập 1 - Yêu cầu 1: Nhận diện một số dạng đồ thị đặc biệt
Nhập 2 - Yêu cầu 2: Xác định thành phần liên thông mạnh
Nhập 3 - Yêu cầu 3: Tìm cây khung lớn nhất
Nhập 4 - Yêu cầu 4: Tìm đường đi ngắn nhất
Nhập 5 - Yêu cầu 5: Tìm chu trình hoặc đường đi Euler
1
Danh sách các input file cho yêu cầu này là:
1. BarbellGraph.txt
2. DirectedGraph.txt
3. HasLoopGraph.txt
4. MultiGraph.txt
5. WindmillGraph.txt
Chọn input file bằng cách nhập chỉ mục tương ứng. Ví dụ 1 cho file đầu tiên trong danh sách
|
    
```

*** Có thể tạo thêm file trong các folder tương ứng để kiểm tra**



Chương 2 Tính toán lý thuyết

2.1 Yêu cầu 1 Nhận diện một số dạng đồ thị đặc biệt

2.1.1 Kiểm tra đầu vào

Kiểm tra lần lượt đồ thị có thỏa các yêu cầu đầu vào của yêu cầu hay không:

- Đồ thị không có cạnh bội: Dùng danh sách kề và kiểm tra xem từng đỉnh cuối trong list con có bị lặp lại hay không.
- Đồ thị vô hướng: Dùng ma trận kề kiểm tra xem ma trận kề có đối xứng hay không.
- Đồ thị không có cạnh khuyên: Dùng ma trận kề, kiểm tra đường chéo của ma trận kề có bằng 0 hay không.

2.1.2 Thuật toán

Sau khi đáp ứng đầy đủ yêu cầu đầu vào tiến hành thực hiện yêu cầu bài toán:

Kiểm tra đồ thị windmill: Tìm đỉnh chung duy nhất với bậc bằng tổng số cạnh trừ 1, kiểm tra xem tất cả các đỉnh còn lại có bậc đúng bằng $k - 1 = 2$ hay không (với $k = 3$ theo đề bài). Tìm số bản sao n của đồ thị bằng cách lấy tổng số cạnh của đồ thị chia cho tổng số cạnh của một đồ thị con đầy đủ

Kiểm tra đồ thị barbell: Chia đồ thị thành 2 nửa bên trái và bên phải kiểm tra xem số đỉnh có thể chia làm đôi hay không. Tìm ra 2 đỉnh cầu nối của đồ thị trái và đồ thị phải và kiểm tra 2 đỉnh đó có nối với nhau hay không. Kiểm tra từng đồ thị con bên trái và bên phải có phải đồ thị đầy đủ hay không. Kết luận bậc của đồ thị barbell chính là số đỉnh của mỗi đồ thị con.

Kiểm tra đồ thị k -partite: Cho lần lượt các đỉnh vào từng partite sao cho mỗi đỉnh trong từng partite đều không kề với nhau. Sau khi có danh sách các partite thì sẽ lặp qua một lần nữa để kiểm tra xem mỗi đỉnh trong từng partite có kề với bất kỳ đỉnh nào trong tất cả các partite còn lại hay không. Đưa ra kết luận với kết quả là danh sách các partite tìm được.

2.2 Yêu cầu 2: Xác định thành phần liên thông mạnh

Một đồ thị vô hướng được gọi là liên thông nếu có đường đi giữa mọi cặp đỉnh phân biệt của đồ thị.

Các thành phần liên thông là các đồ thị con liên thông của một đồ thị không liên thông.

Đồ thị có hướng được gọi là liên thông mạnh nếu có đường đi từ a tới b và từ b tới a với mọi cặp cạnh đỉnh a và b của đồ thị.

Đồ thị có hướng được gọi là liên thông một phần nếu với mọi cặp đỉnh a, b bất kỳ, có ít nhất một đỉnh đến được đỉnh còn lại.

Đồ thị có hướng được gọi là liên thông yếu nếu có đường đi giữa 2 đỉnh bất kỳ của đồ thị vô hướng nền.

2.2.1 Xác định tính liên thông của đồ thị

Đồ thị có hướng liên thông mạnh, liên thông yếu, liên thông một phần

Duyệt đồ thị theo chiều rộng (BFS) cho tất cả các đỉnh của đồ thị có hướng, với V là số đỉnh được duyệt qua tại một đỉnh.

Nếu V của tất cả các đỉnh đều bằng số đỉnh của đồ thị thì đồ thị có hướng liên thông mạnh.

Nếu có ít nhất một V của một đỉnh bằng với số đỉnh của đồ thị thì đồ thị có hướng liên thông một phần

Nếu không có V của đỉnh nào bằng với số đỉnh của đồ thị thì đồ thị có hướng liên thông yếu.

2.2.2 Xác định thành phần liên thông mạnh của đồ thị có hướng

Bước 1: gọi thuật toán duyệt đồ thị trên đồ thị ban đầu và lưu thứ tự các đỉnh duyệt vào stack.

Bước 2: Xây dựng đồ thị lật ngược của đồ thị ban đầu

Bước 3: Lần lượt lấy các đỉnh trong stack ở bước 1 và gọi thuật toán duyệt đồ thị để tìm thành phần liên thông mạnh.

2.3 Yêu cầu 3: Tìm cây khung lớn nhất

Nếu đồ thị bộ phận T của đồ thị vô hướng liên thông $G = (V, E)$ là một cây thì khi đó cây T được gọi là cây khung của đồ thị G .

Cây khung lớn nhất là cây khung có độ dài lớn nhất trong số tất cả các cây khung của đồ thị G .

2.3.1 Tìm cây khung lớn nhất bằng thuật toán Prim:

Cho $G = (V, E)$ là một đồ thị liên thông có trọng số gồm n đỉnh.

Bước 1. Chọn tùy ý một đỉnh bất kỳ $u \in V$ và khởi tạo: $Y = \{u\}$ và $E_T = \Phi$.

Bước 2. Trong số những cạnh $e = \{u, v\}$, trong đó $u \in Y$ và $v \in V \setminus Y$, ta chọn cạnh $e_i = \{u_i, v_i\}$ có độ dài lớn nhất.

Bước 3. Gán $Y = Y \cup \{v_i\}$ và $E_T = E_T \cup \{e_i\}$

Bước 4. Nếu E_T đủ $n - 1$ phần tử thì dừng, ngược lại làm tiếp bước 2.

$T = (V, E_T)$ chính là cây khung lớn nhất.

2.3.2 Tìm cây khung lớn nhất bằng thuật toán Kruskal

Cho $G = (V, E)$ là một đồ thị liên thông có trọng số gồm n đỉnh.

Bước 1. Sắp xếp các cạnh theo thứ tự độ dài giảm dần và khởi tạo: $E_T = \Phi$.

Bước 2. Lần lượt lấy từng cạnh e trong danh sách đã sắp xếp. Nếu $E_T \cup \{e\}$ không tạo thành chu trình trong T thì gán $E_T = E_T \cup \{e\}$

Bước 3. Nếu E_T đủ $n - 1$ phần tử thì dừng, ngược lại làm tiếp bước 2.

$T = (V, E_T)$ chính là cây khung lớn nhất

2.4 Yêu cầu 4: Tìm đường đi ngắn nhất

2.4.1 Trình bày lý thuyết:

-Đường đi ngắn nhất giữa 2 đỉnh bất kỳ sẽ có giá trị là ∞ nếu không có đường nối trực tiếp giữa chúng, hoặc nếu có đường nối thì giá trị ban đầu sẽ là trọng số của đường nối đó. Các tính toán sau này sẽ là so sánh giữa đường đi ngắn nhất hiện tại với một đường đi trung gian (giả sử, với 2 đỉnh A và B, ta so sánh đường đi ngắn nhất hiện tại giữa 2 đỉnh này với tổng độ dài đường đi ngắn nhất từ A và từ B tới 1 đỉnh C trung gian nào đó). Từ đây, quá trình thực hiện thuật toán Floyd-Warshall sẽ diễn ra như sau:

- + Chọn lần lượt từng đỉnh của đồ thị làm đỉnh trung gian (ta quy ước là C).
- + Chọn một cặp 2 đỉnh phân biệt và không trùng với đỉnh trung gian (ta quy ước lần lượt là A và B).
- + Thực hiện so sánh như ở trên: đường đi ngắn nhất giữa A và B sẽ bằng giá trị nhỏ nhất của:
 - + Giá trị đường đi ngắn nhất hiện tại giữa A và B.
 - + Tổng của giá trị đường đi ngắn nhất hiện tại giữa A và C, và đường đi ngắn nhất hiện tại giữa B và C.

Quá trình cài đặt thuật toán Floyd-Warshall khá đơn giản, có thể chia làm 2 công đoạn:

- Khởi tạo mảng 2 chiều chứa đường đi ngắn nhất (ở code này thì đồ thị được biểu diễn ở dạng ma trận kề, distance là tên mảng, $INF = \infty$)

```
for (int i = 1; i <= n; i++) {
    for (int j = 1; j <= n; j++) {
        if (i == j) distance[i][j] = 0;
        else if (adj[i][j]) distance[i][j] = adj[i][j];
        else distance[i][j] = INF;
    }
}
```

-Thực hiện so sánh như ở trên: Đường đi ngắn nhất giữa I và J sẽ bằng giá trị nhỏ nhất của một trong hai giá trị sau:

Giá trị đường đi ngắn nhất hiện tại giữa I và J.

Tổng của giá trị đường đi ngắn nhất hiện tại giữa I và K, và đường đi ngắn nhất hiện tại giữa K và J

```

for (int k = 1; k <= n; k++) {
    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= n; j++) {
            distance[i][j] = min(distance[i][j],
                                distance[i][k]+distance[k][j]);
        }
    }
}

```

2.5 Yêu cầu 5: Tìm chu trình hoặc đường đi Euler

2.5.1 Kiểm tra đầu vào

Trong nội dung yêu cầu cần kiểm tra đồ thị vô hướng liên thông để thực hiện việc tìm kiếm chu trình hoặc đường đi Euler (Không thực hiện tìm chu trình Euler đối với đồ thị có hướng).

Đồ thị vô hướng: với danh sách kề ta duyệt tất cả các cặp đỉnh i và j trong tập V nếu tất cả trọng số cạnh $E[i,j] = E[j,i]$ thì là đồ thị vô hướng (ngược lại là có hướng).

Đồ thị liên thông: thực hiện thuật toán duyệt thăm đỉnh đồ thị

2.5.2 Kiểm tra tính chất Euler của đồ thị

Đối với input đã kiểm tra đồ thị liên thông có hướng. Ta tiếp tục kiểm tra đồ thị có chu trình Euler hoặc đường đi Euler hay không?

- Bước 1: Đếm bậc của tất cả các đỉnh trong đồ thị
- Bước 2: Kiểm tra nếu số đỉnh bậc lẻ:
 - + Nếu số đỉnh bậc lẻ = 0 thì là đồ thị có chu trình Euler (đồ thị Euler).
 - + Nếu số đỉnh bậc lẻ = 2 thì là đồ thị có đường đi Euler (đồ thị nửa Euler).
 - + Nếu số đỉnh bậc lẻ > 2 thì là đồ thị không có chu trình Euler hay đường đi Euler (đồ thị không Euler).

2.5.3 Tìm chu trình/đường đi Euler

(a). Thuật toán Fleury:

- 1./ Chọn điểm bắt đầu
- 2./ Chọn cạnh kế bên
- 3./ Đánh dấu cạnh đã chọn
- 4./ Lặp lại thuật toán đến khi toàn bộ không còn cạnh

(b). Kiểm tra trường hợp bridge

Từ a, b tổng hợp, sử dụng stack để viết code:

Bước 1: khởi tạo Stack<int> và List<int> output

Bước 2: chọn điểm bắt đầu Stack.Push(startVertex)

Bước 3: Với đỉnh trên cùng của Stack ta kiểm tra các đỉnh kề.

+ Nếu có đỉnh kề ta đẩy Stack.Push(i) và xóa cạnh.

+ Nếu không có đỉnh kề ta đẩy đỉnh trong Stack ra - Stack.Pop(i) và lưu đỉnh hiện tại vào List output.

Lặp lại bước 3 cho đến khi nào hết toàn bộ Stack

Đối với đồ thị Euler: chọn đỉnh bắt đầu từ đỉnh 0.

Đối với đồ thị nửa Euler: chọn đỉnh bắt đầu từ đỉnh có bậc lẻ.

Chương 3 Kết quả

3.1 Yêu cầu 1 Nhận diện một số dạng đồ thị đặc biệt

B1: Sau khi lựa chọn nhấn 1 để chọn yêu cầu 1

B2: Nhập tên đồ thị tương ứng với các trường hợp đã được tổ chức sẵn trong folder “Requirement1”

```

ĐỒ ÁN MÔN HỌC NHÓM P14
=====
Vui lòng chọn yêu cầu
Nhập 1 - Yêu cầu 1: Nhận diện một số dạng đồ thị đặc biệt
Nhập 2 - Yêu cầu 2: Xác định thành phần liên thông mạnh
Nhập 3 - Yêu cầu 3: Tìm cây khung lớn nhất
Nhập 4 - Yêu cầu 4: Tìm đường đi ngắn nhất
Nhập 5 - Yêu cầu 5: Tìm chu trình hoặc đường đi Euler
1
Danh sách các input file cho yêu cầu này là:
1. BarbellGraph.txt
2. DirectedGraph.txt
3. HasLoopGraph.txt
4. MultiGraph.txt
5. WindmillGraph.txt
Chọn input file bằng cách nhập chỉ mục tương ứng. Ví dụ 1 cho file đầu tiên trong danh sách
1
=====
    
```

Kiểm tra đầu vào:

- Đồ thị chứa cạnh bội: tên đồ thị “**MultiGraph.txt**”
- Đồ thị có hướng: “**DirectedGraph.txt**”
- Đồ thị có cạnh khuyên: “**HasLoopGraph.txt**”

Thực hiện yêu cầu

- Trường hợp 1: tên đồ thị “**WindmillGraph.txt**”
- Trường hợp 2: “**BarbellGraph.txt**”

Đồ thị chứa cạnh bội:

MultiGraph.txt
4
4 1 1 1 1 3 1 3 1
3 0 1 0 1 2 1
2 1 1 3 1
3 0 1 0 1 2 1

	<p>Chọn input file bằng cách nhập chỉ mục tương ứng. Ví dụ 1 cho file đầu tiên trong danh sách 4</p> <pre> ----- Đồ thị: 4 4 1 1 1 1 3 1 3 1 3 0 1 0 1 2 1 2 1 1 3 1 3 0 1 0 1 2 1 ----- Đồ thị được cho không thỏa yêu cầu: Đồ thị input không phải là đồ thị không có cạnh bội </pre>
Đồ thị có hướng:	<div> <div> <p>DirectedGraph.txt</p> <pre> 4 1 1 1 1 2 1 1 3 1 1 0 1 </pre> </div> <div> <pre> graph TD 0 --> 1 1 --> 2 2 --> 3 3 --> 0 </pre> </div> </div> <div> <p>Chọn input file bằng cách nhập chỉ mục tương ứng. Ví dụ 1 cho file đầu tiên trong danh sách 2</p> <pre> ----- Đồ thị: 4 1 1 1 1 2 1 1 3 1 1 0 1 ----- Đồ thị được cho không thỏa yêu cầu: Đồ thị input không phải là đồ thị vô hướng </pre> </div>
Đồ thị có cạnh khuyên	<div> <div> <p>HasLoopGraph.txt</p> <pre> 4 4 0 1 0 1 1 1 3 1 2 0 1 2 1 2 1 1 3 1 2 0 1 2 1 </pre> </div> <div> <pre> graph TD 0 --> 0 0 --> 1 1 --> 2 2 --> 3 3 --> 0 </pre> </div> </div> <div> <p>Chọn input file bằng cách nhập chỉ mục tương ứng. Ví dụ 1 cho file đầu tiên trong danh sách 3</p> <pre> ----- Đồ thị: 4 4 0 1 0 1 1 1 3 1 2 0 1 2 1 2 1 1 3 1 2 0 1 2 1 ----- Đồ thị được cho không thỏa yêu cầu: Đồ thị input không phải là đồ thị không có cạnh khuyên </pre> </div>

Đồ thị windmill	<div data-bbox="268 219 571 633"> <p>WindmillGraph.txt</p> <pre> 9 8 1 1 2 1 3 1 4 1 5 1 6 1 7 1 8 1 2 0 1 2 1 2 0 1 1 1 2 0 1 4 1 2 0 1 3 1 2 0 1 6 1 2 0 1 5 1 2 0 1 8 1 2 0 1 7 1 </pre> </div> <div data-bbox="603 253 970 600"> </div>
	<p>Chọn input file bằng cách nhập chỉ mục tương ứng. Ví dụ 1 cho file đầu tiên trong danh sách</p> <pre> 5 ----- Đồ thị: 9 8 1 1 2 1 3 1 4 1 5 1 6 1 7 1 8 1 2 0 1 2 1 2 0 1 1 1 2 0 1 4 1 2 0 1 3 1 2 0 1 6 1 2 0 1 5 1 2 0 1 8 1 2 0 1 7 1 ----- Đồ thị cối xay gió: Wd(3,4) Đồ thị barbell: Không Đồ thị k-phân: 3-partite {0} {1, 3, 5, 7} {2, 4, 6, 8} </pre>
Đồ thị barbell	<div data-bbox="255 1216 555 1496"> <p>BarbellGraph.txt</p> <pre> 6 3 1 1 2 1 5 1 2 0 1 2 1 2 0 1 1 1 2 4 1 5 1 2 3 1 5 1 3 0 1 3 1 4 1 </pre> </div> <div data-bbox="587 1272 1082 1451"> </div>

```
-----
Đồ thị:
6
3 1 1 2 1 5 1
2 0 1 2 1
2 0 1 1 1
2 4 1 5 1
2 3 1 5 1
3 0 1 3 1 4 1
-----
Đồ thị cối xay gió: Không
Đồ thị barbell: Bậc 3
Đồ thị k-phân: 3-partite {0, 3} {1, 4} {2, 5}
```

3.2 Yêu cầu 2: Xác định thành phần liên thông mạnh

B1: Sau khi lựa chọn nhấn 2 để chọn yêu cầu 2

B2: Nhập tên đồ thị tương ứng với các trường hợp đã được tổ chức sẵn trong folder “Requirement2”

```
ĐỒ ÁN MÔN HỌC NHÓM P14
-----
Vui lòng chọn yêu cầu
Nhập 1 - Yêu cầu 1: Nhận diện một số dạng đồ thị đặc biệt
Nhập 2 - Yêu cầu 2: Xác định thành phần liên thông mạnh
Nhập 3 - Yêu cầu 3: Tìm cây khung lớn nhất
Nhập 4 - Yêu cầu 4: Tìm đường đi ngắn nhất
Nhập 5 - Yêu cầu 5: Tìm chu trình hoặc đường đi Euler
2
Danh sách các input file cho yêu cầu này là:
1. HasLoopGraph.txt
2. MatrixStrongly.txt
3. MatrixUnconneced.txt
4. MatrixUnilaterally.txt
5. MatrixWeakly.txt
6. MultiGraph.txt
7. UndirectedGraph.txt
Chọn input file bằng cách nhập chỉ mục tương ứng. Ví dụ 1 cho file đầu tiên trong danh sách
```

Kiểm tra đầu vào:

- Đồ thị vô hướng: “**UndirectedGraph.txt**”
- Đồ thị có hướng không cạnh bội: tên đồ thị “**MultiGraph.txt**”
- Đồ thị có hướng không cạnh khuyên: “**HasLoopGraph.txt**”

Thực hiện yêu cầu

- Trường hợp 1: “**MatrixStrongly.txt**”
- Trường hợp 2: “**MatrixUnconneced.txt**”
- Trường hợp 3: “**MatrixUnilaterally.txt**”
- Trường hợp 4: “**MatrixWeakly.txt**”

<p>Đồ thị vô hướng</p>	<div data-bbox="316 212 614 504"> </div> <div data-bbox="316 526 1396 817"> <pre> 7 ----- Đồ thị: 4 2 1 1 3 1 2 0 1 2 1 2 1 1 3 1 2 0 1 2 1 ----- Đồ thị được cho không thỏa yêu cầu: Do thị input không phải là đồ thị có hướng </pre> </div>
<p>Đồ thị có hướng không cạnh khuyên</p>	<div data-bbox="343 929 614 1243"> <pre> HasLoopGraph 4 1 1 1 2 1 1 2 1 1 3 1 1 0 1 </pre> </div> <div data-bbox="710 862 1173 1265"> </div> <div data-bbox="316 1310 1396 1601"> <pre> Chọn input file bằng cách nhập chỉ mục tương ứng. Ví dụ 1 cho file đầu tiên trong danh sách 1 ----- Đồ thị: 4 1 1 1 2 1 1 2 1 1 3 1 1 0 1 ----- Đồ thị được cho không thỏa yêu cầu: Đồ thị input không phải là đồ thị không có cạnh khuyên </pre> </div>

<p>Đồ thị có hướng không cạnh bội</p>	<div data-bbox="349 230 643 568"> <pre>MultiGraph 4 3 1 1 3 1 3 1 1 2 1 1 3 1 0</pre> </div> <div data-bbox="758 203 1150 595"> <pre> graph TD 0((0)) --> 3((3)) 0((0)) --> 3((3)) 1((1)) --> 2((2)) 2((2)) --> 3((3)) </pre> </div> <div data-bbox="320 636 1297 1068"> <pre>6 ----- Đồ thị: 4 3 1 1 3 1 3 1 1 2 1 1 3 1 0 ----- Đồ thị được cho không thỏa yêu cầu: Đồ thị input không phải là đồ thị không có cạnh bội</pre> </div>
<p>Đồ thị liên thông một phần</p>	<div data-bbox="349 1126 1050 1346"> <pre> graph TD 0((0)) --> 1((1)) 1((1)) --> 2((2)) 2((2)) --> 3((3)) 3((3)) --> 4((4)) 4((4)) --> 5((5)) 5((5)) --> 6((6)) 6((6)) --> 7((7)) 7((7)) --> 4((4)) </pre> </div> <div data-bbox="1098 1088 1362 1361"> <pre>MatrixUnilaterally 8 1 1 1 1 2 1 2 3 1 4 1 1 0 1 1 5 1 1 6 1 2 4 1 7 1 0</pre> </div> <div data-bbox="320 1384 793 1951"> <pre>4 ----- Đồ thị: 8 1 1 1 1 2 1 2 3 1 4 1 1 0 1 1 5 1 1 6 1 2 4 1 7 1 0 ----- Đồ thị liên thông một phần Thanh phần liên thông mạnh thu 1: 0 3 2 1 Thanh phần liên thông mạnh thu 2: 4 6 5 Thanh phần liên thông mạnh thu 3: 7</pre> </div>

<p>Đồ thị liên thông yếu</p>	<div data-bbox="352 192 1366 546"> <pre> graph TD 0((0)) --> 1((1)) 0((0)) --> 3((3)) 1((1)) --> 2((2)) 2((2)) --> 3((3)) 2((2)) --> 4((4)) </pre> </div> <div data-bbox="991 203 1286 501"> <p>MatrixWeakly</p> <pre> 5 2 1 1 3 1 1 2 1 1 3 1 1 1 1 1 2 1 </pre> </div> <div data-bbox="320 562 775 1088"> <pre> 5 ----- Đồ thị: 5 2 1 1 3 1 1 2 1 1 3 1 1 1 1 1 2 1 ----- Đồ thị liên thông yếu Thanh phần liên thông mạnh thu 1: 4 Thanh phần liên thông mạnh thu 2: 0 Thanh phần liên thông mạnh thu 3: 1 3 2 </pre> </div>
<p>Đồ thị không liên thông</p>	<div data-bbox="328 1122 959 1469"> <pre> graph TD 0((0)) --> 1((1)) 1((1)) --> 2((2)) 2((2)) --> 3((3)) 3((3)) --> 0((0)) 4((4)) --> 5((5)) 5((5)) --> 6((6)) 6((6)) --> 4((4)) </pre> </div> <div data-bbox="1031 1122 1366 1447"> <p>MatrixUnconnected</p> <pre> 7 1 1 1 1 2 1 1 3 1 1 0 1 1 5 1 1 6 1 1 4 1 </pre> </div>

```

3
-----
Đồ thị:
7
1 1 1
1 2 1
1 3 1
1 0 1
1 5 1
1 6 1
1 4 1
-----
Đồ thị không liên thông
Thanh phần liên thông mạnh thu 1:
4 6 5
Thanh phần liên thông mạnh thu 2:
0 3 2 1

```

3.3 Yêu cầu 3: Tìm cây khung lớn nhất

B1: Sau khi lựa chọn nhấp 3 để chọn yêu cầu 3

B2: Nhập tên đồ thị tương ứng với các trường hợp đã được tổ chức sẵn trong folder “Requirement3”

```

ĐỒ ÁN MÔN HỌC NHÓM P14
-----
Vui lòng chọn yêu cầu
Nhập 1 - Yêu cầu 1: Nhận diện một số dạng đồ thị đặc biệt
Nhập 2 - Yêu cầu 2: Xác định thành phần liên thông mạnh
Nhập 3 - Yêu cầu 3: Tìm cây khung lớn nhất
Nhập 4 - Yêu cầu 4: Tìm đường đi ngắn nhất
Nhập 5 - Yêu cầu 5: Tìm chu trình hoặc đường đi Euler
3
Danh sách các input file cho yêu cầu này là:
1. DirectedGraph.txt
2. Matrix_3.txt
3. UndirectedUnConnectedGraph.txt
Chọn input file bằng cách nhập chỉ mục tương ứng. Ví dụ 1 cho file đầu tiên trong danh sách

```

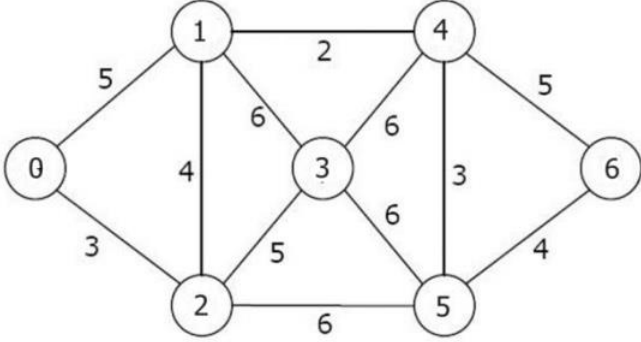
Kiểm tra đầu vào:

- Đồ thị có hướng: “DirectedGraph.txt”
- Đồ thị vô hướng không liên thông: “UndirectedUnConnectedGraph.txt”

Thực hiện yêu cầu

- Trường hợp 1: “Matrix_3.txt”

<p>Đồ thị có hướng</p>	<div data-bbox="263 208 699 622"> <pre> graph TD 0((0)) -- 5 --> 1((1)) 0((0)) -- 40 --> 3((3)) 1((1)) -- 10 --> 2((2)) 2((2)) -- 10 --> 3((3)) </pre> </div> <div data-bbox="252 658 1465 967"> <p>Chọn input file bằng cách nhập chỉ mục tương ứng. Ví dụ 1 cho file đầu tiên trong danh sách</p> <p>1</p> <p>-----</p> <p>Đồ thị:</p> <pre> 4 2 1 5 3 40 1 2 10 1 3 10 0 </pre> <p>-----</p> <p>Đồ thị được cho không thỏa yêu cầu: Do thị input không phải đồ thị vô hướng</p> </div>
<p>Đồ thị vô hướng không liên thông</p>	<div data-bbox="263 1008 699 1422"> <pre> graph TD 0((0)) --- 1((1)) 1((1)) --- 2((2)) 3((3)) </pre> </div> <div data-bbox="252 1444 1465 1706"> <p>Đồ thị:</p> <pre> 4 1 1 1 2 0 1 2 1 1 1 1 0 </pre> <p>-----</p> <p>Đồ thị được cho không thỏa yêu cầu: Do thị input không phải là đồ thị liên thông</p> </div>

Matrix 3		
	<pre> 2 ----- Đồ thị: 7 2 1 5 2 3 4 0 5 2 4 3 6 4 2 4 0 3 1 4 3 5 5 6 4 1 6 2 5 4 6 5 6 4 1 2 3 6 5 3 6 5 4 2 6 3 6 4 3 6 4 2 4 5 5 4 ----- Dinh bat dau cua giai thuat Prim: 0 </pre>	<pre> Dinh bat dau cua giai thuat Prim: 0 Giải thuật Prim Tập cạnh của cây khung 0 - 1: 5 5 - 2: 6 1 - 3: 6 3 - 4: 6 3 - 5: 6 4 - 6: 5 Trọng số của cây khung: 34 Giải thuật Kruskal Tập cạnh của cây khung 5 - 3 6 5 - 2 6 1 - 3 6 4 - 3 6 0 - 1 5 4 - 6 5 Trọng số của cây khung: 34 </pre>

3.4 Yêu cầu 4: Tìm đường đi ngắn nhất

B1: Sau khi lựa chọn nhấp 4 để chọn yêu cầu 4

B2: Nhập tên đồ thị tương ứng với các trường hợp đã được tổ chức sẵn trong folder “Requirement4”

```

ĐỒ ÁN MÔN HỌC NHÓM P14
-----
Vui lòng chọn yêu cầu
Nhập 1 - Yêu cầu 1: Nhận diện một số dạng đồ thị đặc biệt
Nhập 2 - Yêu cầu 2: Xác định thành phần liên thông mạnh
Nhập 3 - Yêu cầu 3: Tìm cây khung lớn nhất
Nhập 4 - Yêu cầu 4: Tìm đường đi ngắn nhất
Nhập 5 - Yêu cầu 5: Tìm chu trình hoặc đường đi Euler
4
Danh sách các input file cho yêu cầu này là:
1. Matrix4.txt
2. Matrix4DoThiCoHuong.txt
3. Matrix4TrongSoAm.txt
Chọn input file bằng cách nhập chỉ mục tương ứng. Ví dụ 1 cho file đầu tiên trong danh sách

```

Kiểm tra đầu vào:

- Đồ thị trọng số âm: “**Matrix4TrongSoAm.txt**”

Thực hiện yêu cầu

- Trường hợp 1: “Matrix4.txt”
- Trường hợp 2: “Matrix4DoThiCoHuong.txt”

<p>Đồ thị có trọng số âm</p>	<div data-bbox="263 331 443 515"> <p>Matrix_4 trong so am.txt</p> <pre>4 2 1 2 2 -4 0 1 1 5 1 0 -1</pre> </div> <div data-bbox="486 309 774 548"> </div> <div data-bbox="263 593 1476 907"> <p>Microsoft Visual Studio Debug Console</p> <p>ĐỒ ÁN MÔN HỌC NHÓM 4</p> <p>-----</p> <p>Vui lòng chọn yêu cầu</p> <p>Nhấn 1 - Yêu cầu 1: Nhận diện một số dạng đồ thị đặc biệt</p> <p>Nhấn 2 - Yêu cầu 2: Xác định thành phần liên thông mạnh</p> <p>Nhấn 3 - Yêu cầu 3: Tìm cây khung nhỏ nhất</p> <p>Nhấn 4 - Yêu cầu 4: Tìm đường đi ngắn nhất</p> <p>Nhấn 5 - Yêu cầu 5: Tìm chu trình hoặc đường đi Euler</p> <p>4</p> <p>Do thi khong thoa yeu cau vi trong so am</p> </div>
<p>Đồ thị vô hướng</p>	<div data-bbox="279 985 630 1176"> </div> <div data-bbox="383 1164 510 1288"> <p>Matrix_4.txt</p> <pre>4 2 1 2 3 1 3 0 2 2 5 3 7 2 1 5 3 6 3 0 1 1 7 2 6</pre> </div>

	<pre> Danh sách ke: Đỉnh 1: (1, 3) Đỉnh 2: (2, 2) Đỉnh 3: (0, 1) Đỉnh 4: (0, 6) (2, 5) Đỉnh số 0: Chỉ mục: 1, Trong số: 3 Đỉnh số 1: Chỉ mục: 2, Trong số: 2 Đỉnh số 2: Chỉ mục: 0, Trong số: 1 Đỉnh số 3: Chỉ mục: 0, Trong số: 6 Chỉ mục: 2, Trong số: 5 Trong số đỉnh 0 dương: 3 Trong số đỉnh 1 dương: 2 Trong số đỉnh 2 dương: 1 Trong số đỉnh 3 dương: 6 Trong số đỉnh 3 dương: 5 </pre>
<p>Đồ thị có hướng</p>	<pre> Matrix_4 do thi co huong.txt 4 1 1 3 1 2 2 1 0 1 2 0 6 2 5 </pre>
	<div> <div> <p>Microsoft Visual Studio Debug Console</p> <p>ĐỒ ÁN MÔN HỌC NHÓM 4</p> <p>Vui lòng chọn yêu cầu</p> <p>Nhấn 1 - Yêu cầu 1: Nhận diện một số dạng đồ thị đặc biệt</p> <p>Nhấn 2 - Yêu cầu 2: Xác định thành phần liên thông mạnh</p> <p>Nhấn 3 - Yêu cầu 3: Tìm cây khung nhỏ nhất</p> <p>Nhấn 4 - Yêu cầu 4: Tìm đường đi ngắn nhất</p> <p>Nhấn 5 - Yêu cầu 5: Tìm chu trình hoặc đường đi Euler</p> <p>4</p> <p>Danh sách ke:</p> <p>Đỉnh 1: (1, 3)</p> <p>Đỉnh 2: (2, 2)</p> <p>Đỉnh 3: (0, 1)</p> <p>Đỉnh 4: (0, 6) (2, 5)</p> <p>Đỉnh số 0:</p> <p> Chỉ mục: 1, Trong số: 3</p> <p>Đỉnh số 1:</p> <p> Chỉ mục: 2, Trong số: 2</p> <p>Đỉnh số 2:</p> <p> Chỉ mục: 0, Trong số: 1</p> <p>Đỉnh số 3:</p> <p> Chỉ mục: 0, Trong số: 6</p> <p> Chỉ mục: 2, Trong số: 5</p> <p>Trong số đỉnh 0 dương: 3</p> <p>Trong số đỉnh 1 dương: 2</p> <p>Trong số đỉnh 2 dương: 1</p> </div> <div> <p>Microsoft Visual Studio Debug Console</p> <p>Chỉ mục: 1, Trong số: 3</p> <p>Đỉnh số 1:</p> <p> Chỉ mục: 2, Trong số: 2</p> <p>Đỉnh số 2:</p> <p> Chỉ mục: 0, Trong số: 1</p> <p>Đỉnh số 3:</p> <p> Chỉ mục: 0, Trong số: 6</p> <p> Chỉ mục: 2, Trong số: 5</p> <p>Trong số đỉnh 0 dương: 3</p> <p>Trong số đỉnh 1 dương: 2</p> <p>Trong số đỉnh 2 dương: 1</p> <p>Trong số đỉnh 3 dương: 6</p> <p>Trong số đỉnh 3 dương: 5</p> <p>Kết quả:</p> <p>Từ đỉnh 4 đến đỉnh 2: 4 -> 2, tổng trọng số: 0</p> </div> </div>

3.5 Yêu cầu 5: Tìm chu trình hoặc đường đi Euler

B1: Sau khi lựa chọn nhấp 5 để chọn yêu cầu 5

B2: Nhập tên đồ thị tương ứng với các trường hợp đã được tổ chức sẵn trong folder “Requirement5”

```

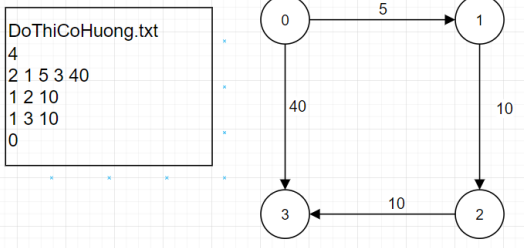
ĐỒ ÁN MÔN HỌC NHÓM P14
-----
Vui lòng chọn yêu cầu
Nhập 1 - Yêu cầu 1: Nhận diện một số dạng đồ thị đặc biệt
Nhập 2 - Yêu cầu 2: Xác định thành phần liên thông mạnh
Nhập 3 - Yêu cầu 3: Tìm cây khung lớn nhất
Nhập 4 - Yêu cầu 4: Tìm đường đi ngắn nhất
Nhập 5 - Yêu cầu 5: Tìm chu trình hoặc đường đi Euler
5
Danh sách các input file cho yêu cầu này là:
1. DoThiCoHuong.txt
2. DoThiEuler.txt
3. DoThiEuler1.txt
4. DoThiKhongEuler.txt
5. DoThiNuaEuler.txt
6. DoThiVoHuongKhongLienThong.txt
Chọn input file bằng cách nhập chỉ mục tương ứng. Ví dụ 1 cho file đầu tiên trong danh sách
    
```

Kiểm tra đầu vào:

- Đồ thị có hướng: “DoThiCoHuong.txt”
- Đồ thị vô hướng không liên thông: “DoThiVoHuongKhongLienThong.txt”

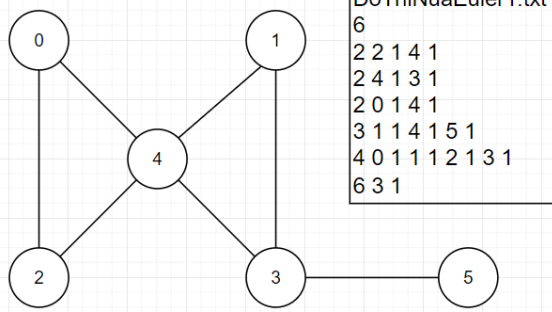
Thực hiện yêu cầu

- Trường hợp 1: “DoThiEuler1.txt”
- Trường hợp 2: “DoThiNuaEuler.txt”

Đồ thị có hướng:	
	<pre> Chọn input file bằng cách nhập chỉ mục tương ứng. Ví dụ 1 cho file đầu tiên trong danh sách 1 ----- Đồ thị: 4 2 1 5 3 40 1 2 10 1 3 10 0 ----- Bắt đầu tìm chu trình hoặc đường đi Euler ----- Đồ thị được cho không thỏa yêu cầu: Đồ thị nhập là đồ thị có hướng </pre>

Đồ thị vô hướng không liên thông	<div data-bbox="288 264 627 436"> <p>DoThiVoHuongKhongLienThong.txt</p> <pre> 4 1 1 1 2 0 1 2 1 1 1 1 0 </pre> </div> <div data-bbox="651 241 850 425"> </div> <div data-bbox="252 526 1461 907"> <p>Chọn input file bằng cách nhập chỉ mục tương ứng. Ví dụ 1 cho file đầu tiên trong danh sách</p> <p>6</p> <p>-----</p> <p>Đồ thị:</p> <pre> 4 1 1 1 2 0 1 2 1 1 1 1 0 </pre> <p>-----</p> <p>Bắt đầu tìm chu trình hoặc đường đi Euler</p> <p>-----</p> <p>Đồ thị được cho không thỏa yêu cầu: Đồ thị nhập là đồ thị không liên thông</p> <p>-----</p> </div>
Đồ thị Euler	<div data-bbox="264 974 464 1160"> <p>DoThiEuler1.txt</p> <pre> 5 2 2 1 4 1 2 4 1 3 1 2 0 1 4 1 2 1 1 4 1 4 0 1 1 1 2 1 3 1 </pre> </div> <div data-bbox="517 972 807 1258"> </div> <div data-bbox="252 1299 1461 1742"> <p>Chọn input file bằng cách nhập chỉ mục tương ứng. Ví dụ 1 cho file đầu tiên trong danh sách</p> <p>3</p> <p>-----</p> <p>Đồ thị:</p> <pre> 5 2 2 1 4 1 2 4 1 3 1 2 0 1 4 1 2 1 1 4 1 4 0 1 1 1 2 1 3 1 </pre> <p>-----</p> <p>Bắt đầu tìm chu trình hoặc đường đi Euler</p> <p>-----</p> <p>Đồ thị Euler</p> <p>Chu trình Euler: 0 2 4 1 3 4 0</p> <p>-----</p> </div>

Đồ thị nửa Euler



DoThiNuaEuler1.txt
6
2 2 1 4 1
2 4 1 3 1
2 0 1 4 1
3 1 1 4 1 5 1
4 0 1 1 1 2 1 3 1
6 3 1

Chọn input file bằng cách nhập chỉ mục tương ứng. Ví dụ 1 cho file đầu tiên trong danh sách
5

Đồ thị:

6
2 2 1 4 1
2 4 1 3 1
2 0 1 4 1
3 1 1 4 1 5 1
4 0 1 1 1 2 1 3 1
6 3 1

Bắt đầu tìm chu trình hoặc đường đi Euler

Đồ thị nửa Euler

Đường đi Euler: 3 1 4 0 2 4 3 5

Chương 4 Tự đánh giá

Bảng tự đánh giá các yêu cầu đã hoàn thành

Nội dung	Hoàn thành	Chưa hoàn thành
Yêu cầu 1: Nhận diện một số dạng đồ thị đặc biệt <ul style="list-style-type: none"> - Kiểm tra đầu vào: đồ thị vô hướng không cạnh bội không cạnh khuyên - Kiểm tra đồ thị cối xay gió - Kiểm tra đồ thị Barbell - Kiểm tra đồ thị K-phân 	Đã hoàn thành	
Yêu cầu 2: Xác định thành phần liên thông mạnh <ul style="list-style-type: none"> - Kiểm tra đầu vào: đồ thị có hướng không cạnh bội không cạnh khuyên - Xác định tính liên thông của đồ thị - Xác định thành phần liên thông trong đồ thị 	Đã hoàn thành	
Yêu cầu 3: Tìm cây khung lớn nhất <ul style="list-style-type: none"> - Kiểm tra đầu vào: đồ thị vô hướng liên thông - Tìm cây khung lớn nhất bằng giải thuật Prim - Tìm cây khung lớn nhất bằng giải thuật Krusal 	Đã hoàn thành	
Yêu cầu 4: Tìm đường đi ngắn nhất <ul style="list-style-type: none"> - Kiểm tra đầu vào: đồ thị trọng số dương - Tìm đường đi ngắn nhất 	Đã hoàn thành	
Yêu cầu 5: Tìm chu trình hoặc đường đi Euler <ul style="list-style-type: none"> - Kiểm tra đầu vào: đồ thị vô hướng liên thông - Kiểm tra tính chất Euler của đồ thị - Tìm chu trình/đường đi Euler của đồ thị (nếu có) 	Đã hoàn thành	

Đánh giá tổng thể mức độ hoàn thành của bài nộp: **100%**