

CÁC GIẢI THUẬT LÝ THUYẾT ĐỒ THỊ

Mô tả yêu cầu của chương trình

1. Mục tiêu tổng quát

Bạn hãy xây dựng chương trình cài đặt một số giải thuật lý thuyết đồ thị. Bạn có thể đã học những giải thuật này trong bài giảng hoặc bạn cần phải tham khảo thêm tài liệu để nắm bắt kiến thức mới.

Chương trình nhận đầu vào là một tập tin văn bản do người dùng chỉ định, nội dung tập tin này thể hiện danh sách kề của một đồ thị.

Chương trình hiển thị các thông tin tương ứng với đồ thị đã cho, bao gồm

- 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 nhỏ nhất
- Yêu cầu 4: Tìm đường đi ngắn nhất
- Yêu cầu 5: Tìm chu trình hoặc đường đi Euler

2. Mô tả dữ liệu đầu vào

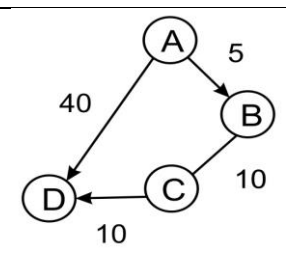
Tập tin văn bản chứa danh sách kề biểu diễn một **đồ thị bất kỳ**. Tập tin có định dạng như sau:

- Dòng đầu tiên chứa số nguyên n ($n > 2$) thể hiện số đỉnh của đồ thị.
- n dòng tiếp theo lần lượt chứa thông tin danh sách kề của đỉnh 0 đến đỉnh $n-1$.
- Danh sách kề của mỗi đỉnh i được biểu diễn bằng $2 \cdot m_i + 1$ số nguyên. Số nguyên đầu tiên là số lượng đỉnh j có cạnh nối từ i đến j (tức là m_i). Các cặp số nguyên tiếp theo là chỉ mục của đỉnh j (chỉ mục tính từ 0) và trọng số tương ứng của cạnh từ i đến j .
 - Đối với những yêu cầu không cần làm việc trên đồ thị trọng số (chẳng hạn như xác định thành phần liên thông mạnh) thì đặt trọng số cho tất cả cạnh bằng 1.
- Các số nguyên liền kề đều cách nhau bằng một khoảng trắng.

Sinh viên có thể chuyển từ danh sách kề sang ma trận kề **bên trong mã nguồn** tùy theo nhu cầu.

Dữ liệu đầu vào **luôn luôn là danh sách kề**.

Ví dụ: quy ước các đỉnh A, B, C tương ứng với chỉ mục 0, 1, 2.

	4	Số lượng đỉnh của đồ thị
	2 1 5 3 40	Danh sách đỉnh kề của đỉnh A
	1 2 10	Danh sách đỉnh kề của đỉnh B
	1 3 10	Danh sách đỉnh kề của đỉnh C
	0	Danh sách đỉnh kề của đỉnh D

3. Yêu cầu 1: Nhận diện một số dạng đồ thị đặc biệt (2 điểm)

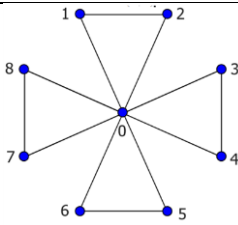
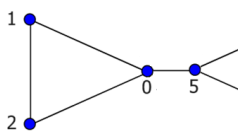
Yêu cầu này chạy trên một đồ thị vô hướng không có cạnh bội và không có cạnh khuyên. Do đó, sinh viên phải kiểm tra đồ thị được cho có phải là đồ thị vô hướng không có cạnh bội và không có cạnh khuyên hay không? Nếu không kiểm tra thì trừ 0.5đ.

Ngoài ra, đồ thị không yêu cầu trọng số, nhưng sinh viên không cần kiểm tra điều này.

Kiểm tra đồ thị được cho có thuộc về các dạng đồ thị đặc biệt dưới đây hay không. Nếu là đồ thị đặc biệt thì cần chỉ rõ tham số tương ứng.

- (0.5đ) Kiểm tra đồ thị cối xay gió $Wd(k,n)$ (windmill graph) với $k = 3$ (cố định). Xác định tham số n . Tham khảo [1].
- (0.5đ) Kiểm tra đồ thị Barbell bậc k (Barbell graph). Xác định tham số k . Tham khảo [2].
- (1đ) Kiểm tra đồ thị k -phân (k -partite graph) với $k > 2$. Xác định tham số k và xác định chỉ mục của các đỉnh nằm trong từng tập hợp con không giao nhau.
 - Đồ thị k -phân là khái niệm được mở rộng từ đồ thị lưỡng phân, trong đó tập hợp đỉnh được chia thành k tập hợp con không giao nhau, thay vì chỉ 2 tập hợp con, và các đỉnh thuộc cùng một tập hợp không có liên kết với nhau.

Ví dụ minh họa (Lưu ý: mục Kết quả hiển thị trên màn hình chỉ có tính chất minh họa, sinh viên không cần xử lý dấu tiếng Việt và định dạng font chữ)

Đồ thị	Kết quả hiển thị trên màn hình
	Đồ thị cối xay gió: $Wd(3,4)$ Đồ thị Barbell: Không Đồ thị k -phân: 3-partite $\{0\}$ $\{1, 3, 5, 7\}$ $\{2, 4, 6, 8\}$
	Đồ thị cối xay gió: Không Đồ thị Barbell: Bậc 3 Đồ thị k -phân: 3-partite $\{0, 3\}$ $\{1, 5\}$ $\{2, 4\}$

Tài liệu tham khảo

[1] Đồ thị cối xay gió: https://en.wikipedia.org/wiki/Windmill_graph

[2] Đồ thị Barbell: <https://mathworld.wolfram.com/BarbellGraph.html>

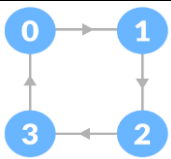
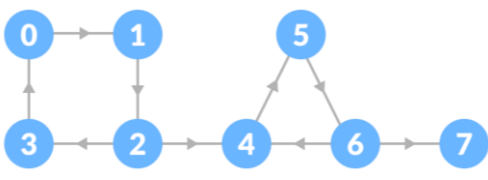
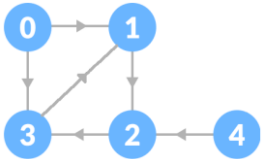
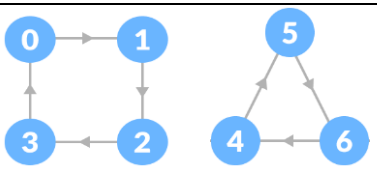
4. Yêu cầu 2: Xác định thành phần liên thông mạnh (2 điểm)

Yêu cầu này chạy trên một đồ thị có hướng không có cạnh bội và không có cạnh khuyên. Do đó, sinh viên phải kiểm tra đồ thị được cho có phải là đồ thị có hướng không có cạnh bội và không có cạnh khuyên hay không. Nếu không kiểm tra thì trừ 0.5đ.

Ngoài ra, đồ thị không yêu cầu trọng số, nhưng sinh viên không cần kiểm tra điều này.

- (1đ) Xác định đồ thị được cho là đồ thị liên thông mạnh, liên thông một phần, liên thông yếu hay không liên thông. Tham khảo [3].
- (1đ) Xác định các thành phần liên thông mạnh có trong đồ thị. Với mỗi thành phần liên thông mạnh, cho biết chỉ mục của các đỉnh thuộc về thành phần liên thông đó. Tham khảo [4].

Ví dụ minh họa (Lưu ý: mục Kết quả hiển thị trên màn hình chỉ có tính chất minh họa, sinh viên không cần xử lý dấu tiếng Việt và định dạng font chữ)

Đồ thị	Kết quả hiển thị trên màn hình
	Đồ thị liên thông mạnh Thành phần liên thông mạnh 1: 0, 1, 2, 3
	Đồ thị liên thông từng phần Thành phần liên thông mạnh 1: 0, 1, 2, 3 Thành phần liên thông mạnh 2: 4, 5, 6 Thành phần liên thông mạnh 3: 7
	Đồ thị liên thông yếu Thành phần liên thông mạnh 1: 0 Thành phần liên thông mạnh 2: 1, 2, 3 Thành phần liên thông mạnh 3: 4
	Đồ thị không liên thông Thành phần liên thông mạnh 1: 0, 1, 2, 3 Thành phần liên thông mạnh 2: 4, 5, 6

Tài liệu tham khảo

[3] Check if a graph is Strongly, Unilaterally or Weakly connected:

<https://www.geeksforgeeks.org/check-if-a-graph-is-strongly-unilaterally-or-weakly-connected/>

[4] Strongly connected components:

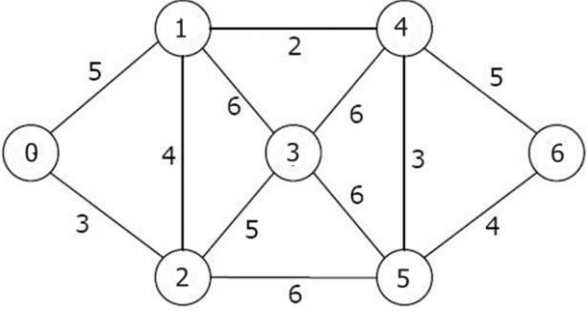
<https://www.geeksforgeeks.org/strongly-connected-components/>

5. Yêu cầu 3: Tìm cây khung lớn nhất (2 điểm)

Yêu cầu này chạy trên đồ thị vô hướng liên thông. Do đó, sinh viên phải kiểm tra đồ thị được cho có phải là đồ thị vô hướng liên thông hay không. Nếu không kiểm tra thì trừ 0.5đ.

- a. (1đ) Tìm *cây khung lớn nhất* trên đồ thị đã cho bằng cách hiệu chỉnh *giải thuật Prim*. Đỉnh bắt đầu do người dùng nhập vào từ bàn phím. Xuất ra màn hình thông tin của cây khung tìm được.
- Danh sách cạnh thuộc cây khung theo thứ tự được phát hiện bởi giải thuật. Quy ước: In mỗi cạnh trên một dòng riêng biệt theo mẫu $\langle x \rangle - \langle y \rangle : \langle z \rangle$, với x và y là hai đỉnh của cạnh và z là trọng số của cạnh.
 - Trọng số của cây khung.
- b. (1đ) Tìm *cây khung lớn nhất* trên đồ thị đã cho bằng cách hiệu chỉnh *giải thuật Kruskal*. Xuất ra màn hình thông tin của cây khung tìm được theo quy cách tương tự như ở câu a.

Ví dụ minh họa (Lưu ý: mục Kết quả hiển thị trên màn hình chỉ có tính chất minh họa, sinh viên không cần xử lý dấu tiếng Việt và định dạng font chữ)

Đồ thị	Kết quả hiển thị trên màn hình	
	Giải thuật Prim	Giải thuật Kruskal
	Tập cạnh của cây khung	Tập cạnh của cây khung
	0 – 1: 5	1 – 3: 6
	1 – 3: 6	2 – 5: 6
	3 – 4: 6	3 – 4: 6
	3 – 5: 6	3 – 5: 6
	5 – 2: 6	0 – 1: 5
	4 – 6: 5	4 – 6: 5
	Trọng số của cây khung:	Trọng số của cây khung:
	34	34

6. Yêu cầu 4: Tìm đường đi ngắn nhất (2 điểm)

Yêu cầu này chạy trên đồ thị có trọng số dương. Do đó, sinh viên phải kiểm tra đồ thị được cho có phải là đồ thị có trọng số dương hay không. Nếu không kiểm tra thì trừ 0.5đ.

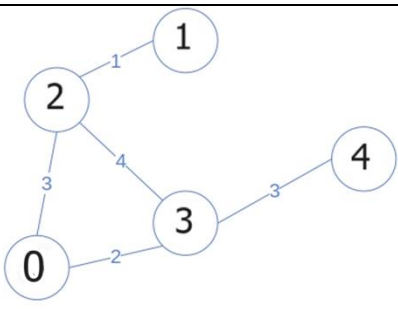
(2đ) Nếu đó là đồ thị có trọng số dương, tìm đường đi ngắn nhất từ **source** đến các đỉnh còn lại bằng giải thuật *Floyd-Warshall*. Ngược lại, dừng. Tham khảo [5].

Với mỗi cặp đỉnh $\langle \text{source}, \text{dest} \rangle$ (dest khác source), nếu tồn tại đường đi ngắn nhất từ source đến dest thì in ra màn hình chi phí của đường đi và chi tiết đường đi theo một trong hai định dạng sau

- Đường đi ngược, ví dụ $3 \leftarrow 2 \leftarrow 1 \leftarrow 0$ hoặc
- Đường đi thuận. ví dụ $0 \rightarrow 1 \rightarrow 2 \rightarrow 3$

Ngược lại, thông báo “Không có đường đi”.

Ví dụ minh họa (Lưu ý: mục Kết quả hiển thị trên màn hình chỉ có tính chất minh họa, sinh viên không cần xử lý dấu tiếng Việt và định dạng font chữ)

Đồ thị	Kết quả hiển thị trên màn hình	
	<p>Đường đi xuất phát từ 0:</p> <p>$0 \rightarrow 2 \rightarrow 1: 3$</p> <p>$0 \rightarrow 2: 3$</p> <p>$0 \rightarrow 3: 2$</p> <p>$0 \rightarrow 3 \rightarrow 4: 5$</p> <p>Đường đi xuất phát từ 1:</p> <p>$1 \rightarrow 2 \rightarrow 0: 4$</p> <p>$1 \rightarrow 2: 1$</p> <p>$1 \rightarrow 2 \rightarrow 3: 5$</p> <p>$1 \rightarrow 2 \rightarrow 3 \rightarrow 4: 8$</p> <p>Đường đi xuất phát từ 2:</p> <p>$2 \rightarrow 0: 3$</p> <p>$2 \rightarrow 1: 1$</p> <p>$2 \rightarrow 3: 4$</p> <p>$2 \rightarrow 3 \rightarrow 4: 7$</p>	<p>Đường đi xuất phát từ 3:</p> <p>$3 \rightarrow 0: 2$</p> <p>$3 \rightarrow 2 \rightarrow 1: 5$</p> <p>$3 \rightarrow 2: 4$</p> <p>$3 \rightarrow 4: 3$</p> <p>Đường đi xuất phát từ 4:</p> <p>$4 \rightarrow 3 \rightarrow 0: 5$</p> <p>$4 \rightarrow 3 \rightarrow 2 \rightarrow 1: 8$</p> <p>$4 \rightarrow 3 \rightarrow 2: 7$</p> <p>$4 \rightarrow 3: 3$</p>

Tài liệu tham khảo

[5] Giải thuật Floy-Warshall:

https://en.wikipedia.org/wiki/Floyd%E2%80%93Warshall_algorithm

7. Yêu cầu 5: Tìm chu trình hoặc đường đi Euler (2 điểm)

Yêu cầu này chạy trên đồ thị vô hướng liên thông không. Do đó, sinh viên phải kiểm tra đồ thị được cho có phải là đồ thị vô hướng liên thông hay không. Nếu không kiểm tra thì trừ 0.5đ.

a. (1đ) Kiểm tra tính chất Euler của đồ thị dựa vào các định lý, tính chất đã học trong lý thuyết (chưa chạy các giải thuật tìm chu trình/đường đi Euler). Tham khảo [6].

Nếu đó là đồ thị Euler, in thông báo “Do thi Euler”. Nếu đó là đồ thị nửa Euler, in thông báo “Do thi nua Euler”. Ngược lại, in thông báo “Do thi khong Euler”.

b. (0.5đ) Nếu đồ thị đã cho là đồ thị Euler thì tiếp tục xác định chu trình Euler xuất phát từ đỉnh **source**. Người dùng nhập từ bàn phím đỉnh bắt đầu **source** (chỉ mục bắt đầu từ 0). R=

- Nếu tồn tại lời giải, in ra chuỗi đỉnh thuộc về chu trình.
- Ngược lại, in thông báo, “Không có lời giải”.

(0.5đ) Nếu đồ thị đã cho là đồ thị nửa Euler thì tiếp tục xác định đường đi Euler xuất phát từ đỉnh **source**. Người dùng nhập từ bàn phím đỉnh bắt đầu **source** (chỉ mục bắt đầu từ 0).

- Nếu tồn tại lời giải, in ra chuỗi đỉnh thuộc về đường đi.
- Ngược lại, in thông báo, “Không có lời giải”.

Tham khảo [7].

Quy ước: Tại một đỉnh bất kỳ, nếu bước tiếp theo có nhiều lựa chọn để đi đến đỉnh kề, hãy chọn đỉnh theo thứ tự chỉ mục từ nhỏ đến lớn.

Ví dụ minh họa (Lưu ý: mục Kết quả hiển thị trên màn hình chỉ có tính chất minh họa, sinh viên không cần xử lý dấu tiếng Việt và định dạng font chữ)

Đồ thị			
Kết quả hiển thị trên màn hình	Đây là đồ thị Euler. Chu trình Euler: a e b d e c a	Đồ thị không Euler.	Đồ thị nửa Euler. Đường đi: d b a c d e

Tài liệu tham khảo

[6] Fleury's Algorithm for printing Eulerian Path or Circuit:

<https://www.geeksforgeeks.org/fleurys-algorithm-for-printing-eulerian-path/>

[7] Fleury's Algorithm:

<https://www.tutorialspoint.com/Fleury-s-Algorithm>

8. Yêu cầu tổ chức chương trình

- Sinh viên tùy ý tổ chức giao diện của chương trình (console hoặc đồ họa, tuy nhiên không có điểm cộng cho lựa chọn đồ họa). Sinh viên cần viết hướng dẫn sử dụng chi tiết để giáo viên có thể chạy chương trình và đánh giá mức độ hoàn thành từng yêu cầu.
- Mã nguồn cho mỗi yêu cầu được để riêng trong từng tập tin. Sinh viên tùy ý thiết kế lớp và hàm, nhưng *không sử dụng kỹ thuật hướng đối tượng quá nhiều* (kế thừa, đa xạ, v.v.) vì như thế thì giáo viên chấm phải đọc nhiều thứ không cần thiết.

Mô tả yêu cầu tổ chức bài nộp

Sinh viên đặt tất cả những tài liệu dưới đây vào thư mục đặt tên theo mã nhóm và nén theo định dạng zip hoặc rar.

- Tập tin PDF trình bày nội dung lý thuyết và ghi chú quan trọng về việc chạy mã nguồn
- Mã nguồn chương trình, tham khảo quy định thực hành để nộp đúng nội dung
- Tập tin Excel thể hiện phân công việc cho các thành viên trong nhóm.

Nhóm trưởng của nhóm (tự quyết định) đại diện nhóm nộp bài làm. Giáo viên sẽ chấm điểm trên bài nộp này (tương ứng theo số liệu có trong bảng phân công), không giải quyết mọi xung đột.

Các trường hợp sau sẽ không được chấm toàn bộ bài làm và nhận điểm 0

- Bài làm tham gia vào việc sao chép: sao chép với lượng chỉnh sửa > 50% từ những nguồn tài liệu sẵn có trên Internet, hoặc sao chép bài làm của người khác
- Sử dụng thư viện phức tạp khiến cho không thể chạy mặc định trên Visual Studio.
- Bài làm còn lỗi cú pháp. Nếu sinh viên sử dụng IDE khác Visual Studio thì phải ghi chú rõ trong báo cáo đi kèm.
- **Định dạng dữ liệu đầu vào không đúng với đặc tả đề bài, ví dụ tự ý chỉnh sửa đặc tả, nhập đồ thị từ bàn phím, v.v.**