Handout Buổi 9: Đường đi và đồ thị Hamilton

🎯 Mục tiêu buổi học

Sau buổi học, sinh viên có thể:

1. **Hiểu rõ định nghĩa** của đường đi Hamilton và đồ thị Hamilton.
2. **Áp dụng các định lý** (Rédei, Dirac, Ore) để xác định tính Hamilton của đồ thị.
3. **Phân biệt** giữa đồ thị Euler và Hamilton.
4. **Giải các bài toán thực tế** như bài toán sắp xếp chỗ ngồi dưới góc nhìn đồ thị Hamilton.
5. **Phát triển tư duy thuật toán**, lập trình kiểm tra đường đi Hamilton.

PHẦN 1: LÝ THUYẾT

4.2.1. Định nghĩa

Trong lý thuyết đồ thị, các khái niệm liên quan đến đường đi Hamilton và đồ thị Hamilton có vai trò quan trọng trong việc nghiên cứu cấu trúc và khả năng kết nối của đồ thị. Dưới đây là các định nghĩa cơ bản:

Định nghĩa 1 (Đường đi Hamilton)

Cho đồ thị $G = (V,E)$, một **đường đi Hamilton** là một dãy đỉnh $v_1, v_2, \dots, v_k \in V$, sao cho:

- Mỗi cặp đỉnh liên tiếp $(v_i, v_{i+1}) \in E$ với $1 \leq i < k$, và
- Mỗi đỉnh trong V xuất hiện **chính xác một lần** trong dãy (không lặp lại đỉnh nào).

Nói cách khác, đường đi Hamilton là một đường đi trong đồ thị mà **đi qua tất cả các đỉnh một lần duy nhất** (trừ trường hợp đặc biệt, có thể không là chu trình).

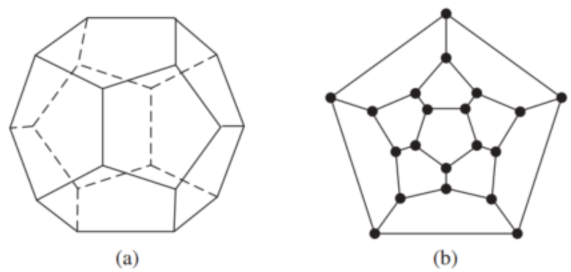
Định nghĩa 2 (Chu trình Hamilton)

Một **chu trình Hamilton** là một đường đi Hamilton mà trong đó **đỉnh đầu tiên và đỉnh cuối cùng trùng nhau** và cũng được nối bởi một cạnh trong đồ thị.
Tức là, chu trình Hamilton là một dãy các đỉnh $v_1, v_2, \dots, v_k$, trong đó:

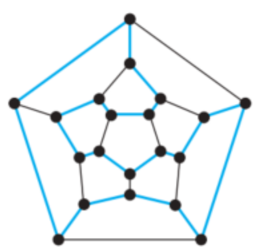
- $(v_i, v_{i+1}) \in E$ với $1 \leq i < k$, và $(v_k, v_1) \in E$;
- Các đỉnh $v_1, v_2, \dots, v_k$ là phân biệt (không lặp), trừ $v_{k+1} = v_1$.

Thuật ngữ này xuất phát từ một trò chơi có tên là **Icosian puzzle**, được phát minh vào năm 1857 bởi nhà toán học người Ireland **Sir William Rowan Hamilton**. Trò chơi này gồm một khối **mười hai mặt đều bằng gỗ** (*dodecahedron* – một khối đa diện có 12 mặt

là các hình ngũ giác đều, như minh họa trong Hình 8(a)), với một **chốt** ở mỗi đỉnh và một **sợi dây**.
Hai mươi đỉnh của khối dodecahedron được gán tên của các thành phố khác nhau trên thế giới. Mục tiêu của trò chơi là bắt đầu từ một thành phố, di chuyển dọc theo các cạnh của khối dodecahedron sao cho **ghé thăm mỗi thành phố còn lại đúng một lần duy nhất**, và **kết thúc tại thành phố ban đầu**. Đường đi này sẽ được đánh dấu bằng cách **quấn sợi dây qua các chốt** tương ứng với các đỉnh đã đi qua.
Vì tác giả không thể cung cấp cho mỗi người đọc một mô hình bằng gỗ kèm theo chốt và dây, nên chúng ta sẽ xét một câu hỏi tương đương: **Có tồn tại một chu trình trong đồ thị minh họa ở Hình 8(b) đi qua mỗi đỉnh đúng một lần không?** Việc trả lời câu hỏi này cũng đồng nghĩa với việc giải được trò chơi, bởi vì đồ thị đó là **đồ thị đẳng cấu** với đồ thị tạo thành từ các đỉnh và cạnh của khối dodecahedron. Một lời giải cho **bài toán Hamilton** này được thể hiện trong Hình 9.

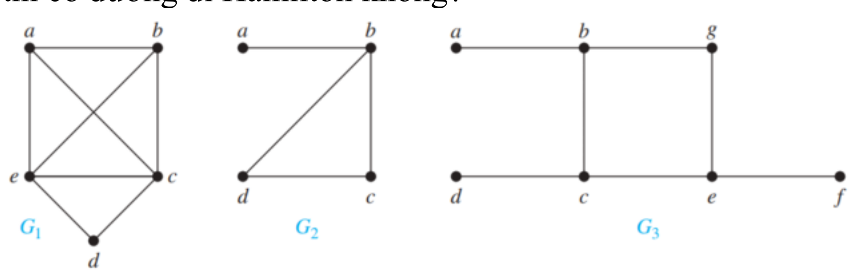


Hình 1: Trò chơi "Một chuyến du hành vòng quanh thế giới" của Hamilton.



Hình 2: Một lời giải cho trò chơi "Một chuyến du hành vòng quanh thế giới".

Bài toán: Trong số các đồ thị đơn trong Hình 3, đồ thị nào có chu trình Hamilton, hoặc nếu không có thì có đường đi Hamilton không?



Hình 3: Ba đồ thị đơn

Lời giải bổ sung:

- **G1** có **chu trình Hamilton**: a → b → c → d → e → a.

- **G2 không có chu trình Hamilton** (có thể thấy điều này vì bất kỳ chu trình nào đi qua tất cả các đỉnh đều phải đi qua cạnh {a, b} hai lần),
  nhưng **G2 có đường đi Hamilton**, ví dụ: a → b → c → d.
- **G3 không có cả chu trình Hamilton lẫn đường đi Hamilton**, vì bất kỳ đường đi nào đi qua tất cả các đỉnh đều phải đi qua một trong các cạnh {a, b}, {e, f}, hoặc {c, d} nhiều hơn một lần.

Định nghĩa 3 (Đồ thị Hamilton)

Một đồ thị G được gọi là **đồ thị Hamilton** nếu tồn tại ít nhất một chu trình Hamilton trong G.

Nhận xét

- Không phải mọi đồ thị liên thông đều là đồ thị Hamilton.
- Việc xác định một đồ thị có phải là đồ thị Hamilton hay không là một bài toán khó trong khoa học máy tính, thuộc lớp các bài toán **NP-đầy đủ**.
- Đường đi Hamilton khác với đường đi Euler: đường đi Hamilton liên quan đến **đỉnh**, còn đường đi Euler liên quan đến **cạnh**.

4.2.2. Định lý Rédei

Trong ngữ cảnh của đồ thị định hướng, khái niệm về **liên thông mạnh** đóng vai trò then chốt trong việc đảm bảo sự tồn tại của các đường đi Hamilton. Định lý Rédei cung cấp một điều kiện đủ quan trọng cho sự tồn tại của đường đi Hamilton trong đồ thị định hướng.

Định lý (Rédei, 1934)

Trong mọi đồ thị có hướng mạnh liên thông mạnh, luôn tồn tại ít nhất một đường đi Hamilton.
Cụ thể, nếu $D = (V,A)$ là một **đồ thị có hướng liên thông mạnh**, tức là với mọi cặp đỉnh $u, v \in V$ luôn tồn tại đường đi định hướng từ u đến v, thì D chứa một dãy đỉnh $v_1, v_2, \ldots, v_n$ sao cho:

- $(v_i,v_{i+1}) \in A$ với mọi $1 \leq i < n$, và
- Mỗi đỉnh trong V xuất hiện **một lần duy nhất** trong dãy.

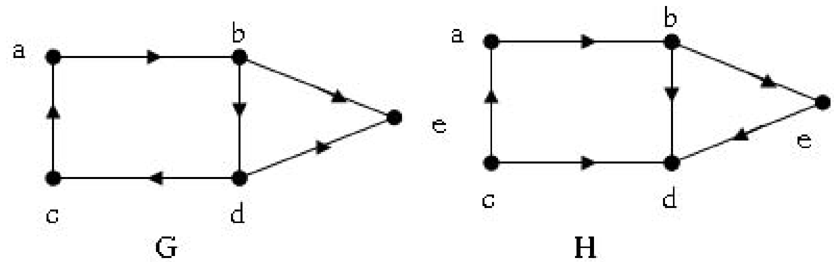
Đường đi này là một **đường đi Hamilton**, tuy nhiên **không yêu cầu phải quay trở lại đỉnh đầu tiên** nên nó **không nhất thiết là chu trình Hamilton**.

Hệ quả và ý nghĩa

- Định lý Rédei cho thấy rằng tính chất liên thông mạnh trong đồ thị định hướng là **điều kiện đủ để đảm bảo sự tồn tại của một đường đi Hamilton**.

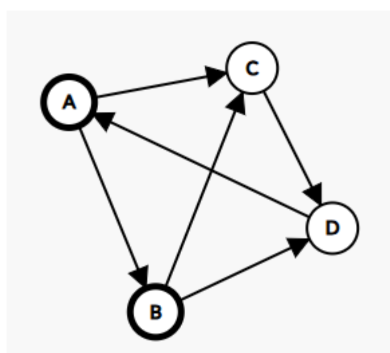
- Tuy nhiên, **không phải mọi đồ thị định hướng mạnh liên thông đều có chu trình Hamilton**.
- Điều này có ý nghĩa quan trọng trong ứng dụng, ví dụ như trong bài toán lập lịch tuyến đường hoặc tổ chức thứ tự công việc có quan hệ phụ thuộc định hướng.

Ví dụ minh họa



Xét đồ thị định hướng D gồm 4 đỉnh $\{A,B,C,D\}$ với các cung:

- $A \rightarrow B$, $B \rightarrow C$, $C \rightarrow D$, $D \rightarrow A$, $B \rightarrow D$



Ta thấy D là đồ thị định hướng mạnh liên thông (có đường đi định hướng giữa mọi cặp đỉnh), và tồn tại đường đi Hamilton: **A → B → C → D**

4.2.3. Định lý Dirac (1952)

Một trong những kết quả cơ bản và nổi bật trong lý thuyết đồ thị liên quan đến chu trình Hamilton là **định lý của Dirac**. Đây là một định lý đưa ra **điều kiện đủ** để một đồ thị đơn là đồ thị Hamilton, dựa trên **bậc tối thiểu** của các đỉnh.

Định lý (Dirac, 1952)
Cho $G = (V,E)$ là một đồ thị đơn có $n \geq 3$ đỉnh. Nếu với mọi đỉnh $v \in V$, ta có $\deg(v) \geq \frac{n}{2}$, thì G là một đồ thị Hamilton.

Giải thích các giả thiết

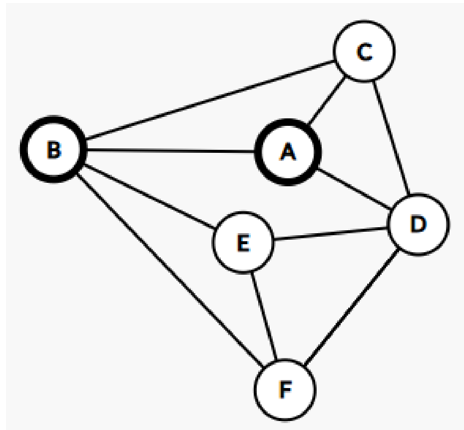
- **Đồ thị đơn**: không có khuyên (loop) và không có nhiều cạnh nối giữa cùng một cặp đỉnh.
- n: số đỉnh của đồ thị.
- $\deg(v)$: bậc của đỉnh v, tức là số lượng đỉnh kề với v.
  Điều kiện $\deg(v) \geq \frac{n}{2}$ phải đúng với **mọi đỉnh** trong đồ thị.

Ý nghĩa

- Định lý Dirac không chỉ cung cấp một tiêu chí hình thức để kiểm tra tính Hamilton của đồ thị, mà còn là một trong những điều kiện đơn giản và mạnh mẽ nhất.
- Định lý này là một ví dụ tiêu biểu về **điều kiện bậc toàn cục** ảnh hưởng đến **cấu trúc toàn cục** của đồ thị.

Ví dụ minh họa

Xét đồ thị G có 6 đỉnh $\{A,B,C,D,E,F\}$. Nếu mỗi đỉnh có bậc ít nhất là 3, tức là $\deg(v) \geq \frac{6}{2} = 3$, thì theo định lý Dirac, đồ thị này chắc chắn là Hamilton.



Một chu trình Hamilton khả dĩ có thể là: **A → B → C → D → E → F → A**

Lưu ý

- Điều kiện trong định lý là **điều kiện đủ**, không phải là điều kiện cần. Nghĩa là:
    - Nếu thỏa mãn điều kiện thì chắc chắn đồ thị là Hamilton.
    - Nhưng nếu **không thỏa mãn** điều kiện thì **không thể kết luận** được gì (đồ thị vẫn *có thể* là Hamilton).

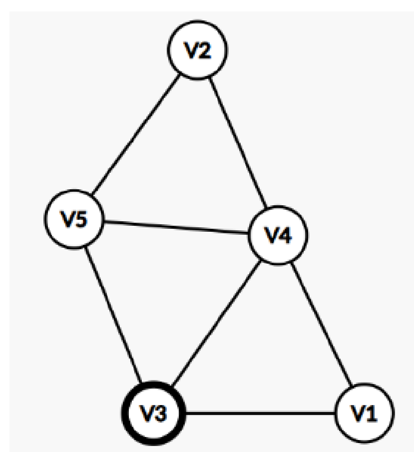
📌 Ví dụ phản ví dụ:

Đồ thị 5 đỉnh:

- deg(v1)=2, deg(v2)=2, deg (v4)=4,
- Các đỉnh còn lại có bậc 3

→ Không thỏa Dirac

✅ Nhưng vẫn có thể tồn tại chu trình Hamilton!



4.2.4. Định lý Ore (1960)

Định lý Ore mở rộng và làm mạnh thêm kết quả của định lý Dirac bằng cách thay điều kiện bậc riêng lẻ của từng đỉnh bằng điều kiện tổng bậc của các cặp đỉnh không kề nhau. Đây tiếp tục là một **điều kiện đủ** để đảm bảo một đồ thị đơn là Hamilton.

Định lý (Ore, 1960)

Cho $G = (V,E)$ là một đồ thị đơn có $n \geq 3$ đỉnh. Nếu với mọi cặp đỉnh $u, v \in V$, mà u và v ***không kề nhau****, ta có:*

$$\deg(u)+\deg(v)\geq n,$$

thì G là một đồ thị Hamilton.

Giải thích điều kiện

- **Không kề nhau**: Hai đỉnh u, v được gọi là không kề nếu không có cạnh nối trực tiếp giữa chúng, tức là $\{u,v\} \notin E$.
- Điều kiện tổng bậc $\deg(u)+\deg(v)\geq n$ phải đúng cho **mọi cặp đỉnh không kề nhau** trong đồ thị.

Mối liên hệ với định lý Dirac

- Định lý Ore là **tổng quát hóa** của định lý Dirac:
  Nếu với mọi đỉnh $v \in V$, ta có $\deg(v) \geq \frac{n}{2}$, thì với mọi cặp đỉnh không kề u, v, ta sẽ có:

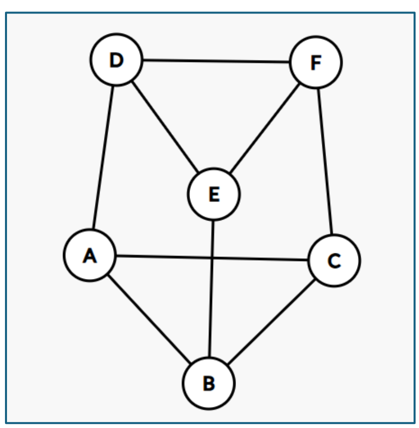
$$\deg(u)+\deg(v) \geq \frac{n}{2} + \frac{n}{2} = n.$$

  - Do đó, định lý Dirac là **trường hợp đặc biệt** của định lý Ore.

Ví dụ minh họa

Xét đồ thị đơn G có $n = 6$ đỉnh. Giả sử tồn tại các cặp đỉnh không kề nhau và với mọi cặp như vậy $\deg(u)+\deg(v)\geq 6$, thì theo định lý Ore, G là đồ thị Hamilton.

Ví dụ về chu trình Hamilton có thể là: **A → B → C → D → E → F → A**



Đỉnh	**Bậc (deg)**	**Nối với**
A	3	B, C, D
B	3	A, C, E
C	3	A, B, F
D	3	A, E, F
E	3	B, D, F
F	3	C, D, E

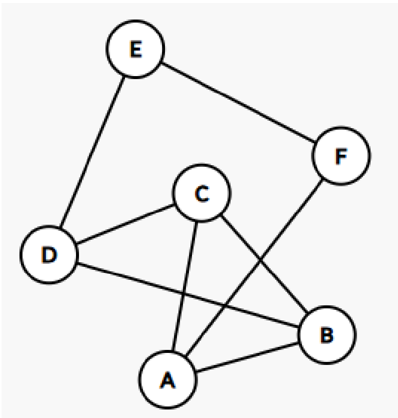
Cặp không kề	deg(u) + deg(v)	Thỏa Ore?
A – E	3 + 3 = 6	✅
A – F	3 + 3 = 6	✅
B – D	3 + 3 = 6	✅
B – F	3 + 3 = 6	✅
C – D	3 + 3 = 6	✅
C – E	3 + 3 = 6	✅

Lưu ý

- Giống như định lý Dirac, định lý Ore là **điều kiện đủ**: nếu thỏa mãn điều kiện thì đồ thị chắc chắn là Hamilton, nhưng không thỏa mãn thì không thể kết luận ngược lại.
- Điều kiện trong định lý Ore thường dễ kiểm tra hơn trong thực tế vì chỉ cần xét các cặp đỉnh không kề.

Định lý Ore "vào trận" – minh họa và ứng dụng

Cho đồ thị đơn G có 6 đỉnh: A, B, C, D, E, F.



📌 Bậc các đỉnh:

Đỉnh	Bậc
A	3
B	3
C	3
D	3
E	2
F	2

👉 Kiểm tra các cặp không kề:

(A,D): deg(A) + deg(D) = 3 + 3 = 6 ✅

(B,E): deg(B) + deg(E) = 3 + 2 = 5 ❌ → Không thỏa Ore.

4.2.5. Bài toán sắp xếp chỗ ngồi

Trong thực tế, nhiều bài toán liên quan đến việc sắp xếp tuần tự các đối tượng thỏa mãn một số điều kiện nhất định có thể được mô hình hóa bằng các khái niệm trong lý thuyết đồ thị. Một ví dụ tiêu biểu là **bài toán sắp xếp chỗ ngồi**, có thể được diễn giải như một bài toán tìm **chu trình Hamilton** trong một đồ thị không trọng số.

Mô tả bài toán

Giả sử có n người tham dự một buổi tiệc và cần sắp xếp họ ngồi quanh một bàn tròn. Mỗi người có một danh sách những người mà họ **thích được ngồi cạnh**, hoặc ngược lại là **không muốn ngồi cạnh**. Mục tiêu là tìm một cách sắp xếp sao cho:

- Mỗi người ngồi cạnh đúng **hai người,**
- Và hai người này nằm trong danh sách mà họ **chấp nhận** hoặc **ưu tiên ngồi gần**.

Mô hình hóa bằng đồ thị

Ta xây dựng một **đồ thị đơn không có hướng** $G = (V,E)$ như sau:

- **Đỉnh** V: Mỗi đỉnh đại diện cho một người trong nhóm n người.
- **Cạnh** E: Hai đỉnh $u, v \in V$ được nối bởi một cạnh nếu **cả hai người u và v đều đồng ý ngồi cạnh nhau** (tức là nằm trong danh sách chấp nhận lẫn nhau).

Yêu cầu: Tìm một **chu trình Hamilton** trong đồ thị G, tức là một chuỗi sắp xếp vòng tròn của n người sao cho mỗi người ngồi cạnh đúng hai người khác và tất cả đều là người họ chấp nhận.

Ý nghĩa

- Nếu đồ thị G có chu trình Hamilton, thì tồn tại một **sắp xếp chỗ ngồi hợp lệ**.
- Nếu không tồn tại chu trình Hamilton, thì:
    - Không có cách sắp xếp toàn bộ nhóm thỏa mãn yêu cầu.
    - Có thể cần loại bỏ một số ràng buộc hoặc cho phép sự linh hoạt trong quy tắc (ví dụ: cho phép một số người ngồi cạnh người họ không ưa).

Ví dụ minh họa

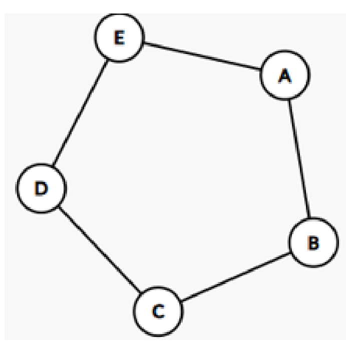
Giả sử có 5 người: A, B, C, D, E. Quan hệ chấp nhận ngồi cạnh nhau được cho như sau (đối xứng hai chiều):

- A chấp nhận B, E
- B chấp nhận A, C
- C chấp nhận B, D
- D chấp nhận C, E
- E chấp nhận D, A

Người	Cạnh
A	B, E
B	A, C
C	B, D
D	C, E
E	D, A

Từ đó, đồ thị GG có các cạnh: $\{A,B\},\{A,E\},\{B,C\},\{C,D\},\{D,E\}$

Ta thấy đồ thị này có chu trình Hamilton: **A → B → C → D → E → A**



→ Vậy tồn tại cách sắp xếp chỗ ngồi vòng tròn hợp lệ.

Ứng dụng mở rộng

- Lập lịch thuyết trình có thứ tự ràng buộc.
- Tổ chức chuỗi nhiệm vụ trong sản xuất.
- Tối ưu hóa thứ tự xử lý yêu cầu trong hệ thống mạng.

PHẦN 2: BÀI TẬP CÓ LỜI GIẢI

🧮 Bài 1: Kiểm tra đồ thị có phải là Hamilton không bằng định lý Dirac

Cho một đồ thị đơn G có $n = 6$ đỉnh, với tập bậc các đỉnh như sau:

$$\deg(v_1) = 3, \deg(v_2)=3, \deg(v3)=4, \deg(v4)=3, \deg(v5)=3, \deg(v6)=3$$

Hỏi: Theo định lý **Dirac**, đồ thị G có phải là đồ thị Hamilton không?

🔍 Phân tích chi tiết

- **Định lý Dirac**: Với đồ thị đơn G có $n \geq 3$, nếu với mọi đỉnh $v \in V$, ta có: $\deg(v) \geq \frac{n}{2}$ thì G là đồ thị Hamilton.
- Với $n = 6$ ta có: $\frac{n}{2} = 3$
- Kiểm tra điều kiện bậc:
    - $\deg(v_i) \geq 3$ với mọi $i \in \{1,2,3,4,5,6\}$ → Điều kiện Dirac được **thỏa mãn**.

➡ **Kết luận**: Theo định lý Dirac, đồ thị G **chắc chắn là đồ thị Hamilton**.

💻 Code kiểm tra điều kiện Dirac (Python)

```
def is_dirac_hamiltonian(degrees):
    n = len(degrees)
    threshold = n / 2
    for deg in degrees:
        if deg < threshold:
            return False
    return True

# Bậc của các đỉnh
degrees = [3, 3, 4, 3, 3, 3]

if is_dirac_hamiltonian(degrees):
    print("Đồ thị thỏa mãn định lý Dirac → là đồ thị Hamilton.")
else:
```

print("Đồ thị không thỏa mãn định lý Dirac → không kết luận được.")

🧠 Ghi chú thêm cho sinh viên

- Kết quả trên chỉ đảm bảo đồ thị là **Hamilton** nếu **tất cả các đỉnh** thỏa điều kiện Dirac.
- Nếu có ít nhất một đỉnh có bậc $< n/2$, thì **không thể áp dụng định lý**, nhưng đồ thị vẫn *có thể* là Hamilton.

🧮 Bài 2: Chứng minh đồ thị định hướng có đường đi Hamilton (áp dụng định lý Rédei)

Cho đồ thị định hướng D có 4 đỉnh với các cung như sau:

- $A \to B$
- $B \to C$
- $C \to D$
- $D \to A$
- $A \to C$
- $B \to D$

Hãy chứng minh rằng D **có đường đi Hamilton**, bằng cách áp dụng định lý **Rédei**.

🔍 Phân tích chi tiết

- **Định lý Rédei** phát biểu rằng:

  > Mọi đồ thị định hướng mạnh liên thông đều chứa ít nhất một đường đi Hamilton.

- Vậy ta cần:
  1. Kiểm tra xem đồ thị có **mạnh liên thông** không: tức là với mọi cặp đỉnh (u,v), luôn tồn tại **đường đi có hướng** từ $u \to v$.
  2. Nếu có, thì suy ra theo định lý Rédei, đồ thị có chứa **đường đi Hamilton**.
- **Kiểm tra mạnh liên thông**:

- Có thể kiểm tra bằng **thuật toán DFS hoặc BFS toàn bộ**, từ mỗi đỉnh xem có đến được tất cả các đỉnh khác không.
- Với đồ thị nhỏ (4 đỉnh), ta có thể kiểm tra thủ công:
    - Từ A: đi đến B, C, D
    - Từ B: đi đến C, D → từ D quay lại A → từ A đến B
    - Từ C: đi đến D → D → A → A → B
    - Từ D: đi đến A → A → B, C

✅ **Kết luận**: Đồ thị **mạnh liên thông**.

➡ Theo **Rédei**, đồ thị **có đường đi Hamilton**.

💻 Code kiểm tra mạnh liên thông (Python)

Dưới đây là mô phỏng thuật toán kiểm tra **mạnh liên thông** bằng BFS toàn phần:

```
from collections import defaultdict, deque

def is_strongly_connected(graph, n):
    def bfs(start):
        visited = [False] * n
        queue = deque([start])
        visited[start] = True
        while queue:
            u = queue.popleft()
            for v in graph[u]:
                if not visited[v]:
                    visited[v] = True
                    queue.append(v)
        return all(visited)

    # Kiểm tra từ mọi đỉnh
    for i in range(n):
        if not bfs(i):
            return False
    return True

# Đồ thị định hướng: đỉnh A=0, B=1, C=2, D=3
graph = defaultdict(list)
graph[0] = [1, 2]      # A -> B, C
graph[1] = [2, 3]      # B -> C, D
```

```
graph[2] = [3]        # C -> D
graph[3] = [0]        # D -> A

n = 4

if is_strongly_connected(graph, n):
    print("Đồ thị định hướng mạnh liên thông → Có đường đi Hamilton (theo định lý Rédei).")
else:
    print("Không mạnh liên thông → Không áp dụng được định lý Rédei.")
```

🧠 Ghi chú cho sinh viên

- Định lý Rédei chỉ đảm bảo sự tồn tại của **đường đi Hamilton**, không phải **chu trình Hamilton**.
- Với đồ thị nhỏ, ta có thể **liệt kê thủ công** các đường đi Hamilton:
  - Ví dụ: A → B → C → D

🧮 Bài 3: Tìm tất cả chu trình Hamilton của đồ thị đơn nhỏ (n ≤ 5)

Cho đồ thị đơn $G = (V, E)$ có tập đỉnh $V = \{A, B, C, D\}$ và tập cạnh:

$$E=\{(A,B),(B,C),(C,D),(D,A),(A,C),(B,D)\}$$

Yêu cầu: Liệt kê **tất cả các chu trình Hamilton** của đồ thị G (bắt đầu tại đỉnh A), và viết chương trình Python để tìm các chu trình đó.

🔍 Phân tích chi tiết

- Số đỉnh $n = 4$ → số lượng chu trình Hamilton tối đa (nếu đồ thị đầy đủ): $\frac{(n-1)!}{2} = 3$
- Một **chu trình Hamilton** là một dãy các đỉnh: $A \to v_2 \to v_3 \to v_4 \to A$

  sao cho mỗi cặp liên tiếp đều nối với nhau bằng một cạnh trong đồ thị và **không có đỉnh nào bị lặp lại** (trừ A xuất hiện 2 lần: đầu và cuối).
- Giải pháp: Duyệt toàn bộ các hoán vị của 3 đỉnh còn lại $\{B, C, D\}$, và kiểm tra xem có tạo thành một chu trình hợp lệ hay không.

💻 Lời giải bằng Python

```
from itertools import permutations
```

```python
# Đồ thị biểu diễn bằng danh sách kề
graph = {
    'A': ['B', 'C', 'D'],
    'B': ['A', 'C', 'D'],
    'C': ['A', 'B', 'D'],
    'D': ['A', 'B', 'C']
}

def is_valid_cycle(path):
    for i in range(len(path) - 1):
        if path[i+1] not in graph[path[i]]:
            return False
    # Kiểm tra quay lại đỉnh đầu
    return path[0] in graph[path[-1]]

# Tìm tất cả hoán vị của các đỉnh còn lại (không gồm A)
vertices = ['B', 'C', 'D']
start = 'A'
cycles = []

for perm in permutations(vertices):
    path = [start] + list(perm) + [start]
    if is_valid_cycle(path):
        cycles.append(path)

# In kết quả
print("Các chu trình Hamilton bắt đầu tại A:")
for cycle in cycles:
    print(" → ".join(cycle))
```

📃 **Kết quả đầu ra**

Các chu trình Hamilton bắt đầu tại A:
A → B → C → D → A
A → B → D → C → A
A → C → B → D → A
A → C → D → B → A
A → D → B → C → A
A → D → C → B → A

→ Vì đồ thị là đồ thị đầy đủ (complete graph) với 4 đỉnh, nên tồn tại đúng $\frac{(n-1)!}{2} = \mathbf{3}$ chu trình Hamilton khác nhau (nếu không tính hoán vị ngược chiều là khác biệt), hoặc 6 nếu phân biệt chiều đi.

🧠 Ghi chú cho sinh viên

- Với đồ thị nhỏ ($n \leq 7$), có thể liệt kê toàn bộ các chu trình Hamilton bằng brute-force.
- Với đồ thị lớn hơn, bài toán thuộc lớp **NP-đầy đủ** và khó giải bằng duyệt toàn bộ.
- Bạn có thể dùng thêm **backtracking** để tối ưu hóa khi tìm các chu trình hợp lệ.

🧮 Bài tập lý thuyết có lời giải – Bài 4

Phân tích sự khác biệt giữa đồ thị Euler và đồ thị Hamilton qua ví dụ cụ thể. Trình bày rõ định nghĩa, điều kiện tồn tại, ví dụ minh họa và ứng dụng.

🔍 Phân tích chi tiết

1. 🧠 Định nghĩa:

Đặc điểm	Đồ thị Euler	Đồ thị Hamilton
Loại đường đi	Đi qua tất cả các cạnh đúng 1 lần	Đi qua tất cả các đỉnh đúng 1 lần
Có thể lặp đỉnh không?	Có	Không
Có thể lặp cạnh không?	Không	Không
Chu trình?	Chu trình Euler kết thúc tại điểm xuất phát	Chu trình Hamilton kết thúc tại điểm xuất phát

2. 📜 Điều kiện tồn tại:

- Chu trình Euler (trên đồ thị vô hướng):
  Đồ thị liên thông và mọi đỉnh đều có bậc chẵn.
- Chu trình Hamilton:
  Không có điều kiện đơn giản tổng quát. Một số định lý như:
  - Dirac: Nếu $\deg(v) \geq n/2$ với mọi đỉnh v ($n \geq 3$), thì đồ thị có chu trình Hamilton.
  - Ore: Nếu $\deg(u) + \deg(v) \geq n$ với mọi cặp đỉnh không kề nhau u, v.

3. 🎲 Ví dụ minh họa:

Ví dụ 1: Đồ thị Euler nhưng không phải Hamilton

Xét đồ thị G gồm 4 đỉnh và các cạnh sau:

```
A---B
|   |
D---C
```

Đồ thị này có chu trình Euler: A → B → C → D → A (tất cả các cạnh đúng 1 lần).
Tuy nhiên, không thể tìm đường đi đi qua tất cả đỉnh đúng 1 lần và quay về điểm xuất phát mà không lặp đỉnh → không có chu trình Hamilton.

Ví dụ 2: Đồ thị Hamilton nhưng không phải Euler

Đồ thị hình ngũ giác:

```
A---B
|   |
E---C
 \ /
  D
```

Chu trình Hamilton: A → B → C → D → E → A (đi qua mọi đỉnh đúng 1 lần).
Tuy nhiên, vì một số đỉnh có bậc lẻ → không có chu trình Euler.

---

4. 💻 Code Python minh họa sự khác biệt

```
import networkx as nx
import matplotlib.pyplot as plt
from itertools import permutations

# Tạo đồ thị đơn giản
G = nx.Graph()
edges = [('A', 'B'), ('B', 'C'), ('C', 'D'), ('D', 'E'), ('E', 'A')]
G.add_edges_from(edges)

# Vẽ đồ thị
nx.draw(G, with_labels=True, node_color='lightblue', node_size=800)
plt.title("Đồ thị ví dụ có chu trình Hamilton")
plt.show()

# Hàm kiểm tra Hamilton Cycle (brute-force)
def has_hamiltonian_cycle(G):
    nodes = list(G.nodes)
    for perm in permutations(nodes):
        if all(G.has_edge(perm[i], perm[i+1]) for i in range(len(perm)-1)) and
G.has_edge(perm[-1], perm[0]):
```

```
        return True, perm + (perm[0],)
    return False, None

# Kiểm tra chu trình Hamilton
result, cycle = has_hamiltonian_cycle(G)
print("Có chu trình Hamilton:" if result else "Không có chu trình Hamilton.")
if cycle:
    print("Chu trình:", " -> ".join(cycle))
```

5. 📌 Kết luận và ứng dụng

Khía cạnh	Euler	Hamilton
Ứng dụng	Đi bộ dọc các cây cầu (Euler)	Sắp xếp chỗ ngồi, du lịch vòng quanh (TSP)
Kiểm tra	Đa thức	NP-đầy đủ

🧮 Bài tập lý thuyết có lời giải – Bài 5

Lập mô hình bài toán sắp xếp chỗ ngồi bằng đồ thị và giải bằng thuật toán tìm chu trình Hamilton.

🔍 Phân tích bài toán

🔷 Bài toán thực tế:

Giả sử có n người muốn ngồi thành một vòng tròn. Mỗi người có danh sách những người họ muốn ngồi cạnh hoặc không muốn ngồi cạnh.
Yêu cầu: Sắp xếp chỗ ngồi sao cho mọi người đều ngồi cạnh người họ thích (nếu có) và vòng tròn được hoàn thành (tức là chu trình).

📐 Bước 1: Mô hình hóa thành đồ thị

6. Đỉnh (V): Mỗi người là một đỉnh.
7. Cạnh (E): Có một cạnh giữa hai người nếu họ muốn ngồi cạnh nhau (có thể đơn phương hoặc song phương).
8. Yêu cầu: Tìm chu trình Hamilton trên đồ thị — mỗi đỉnh xuất hiện đúng một lần và các đỉnh liên tiếp có cạnh nối → tương ứng với việc mọi người ngồi cạnh nhau theo vòng.

🔢 Ví dụ cụ thể

Danh sách người:

Người	Muốn ngồi gần
A	B, C
B	A, D
C	A, D
D	B, C

Mô hình hóa:

- Đỉnh: {A, B, C, D}
- Cạnh: {(A,B), (A,C), (B,D), (C,D)}

💡 **Nhận xét:**

- Đồ thị có 4 đỉnh, các cạnh phản ánh sở thích.
- Yêu cầu: Tìm một chu trình Hamilton như: A → B → D → C → A

💻 **Python – Tìm chu trình Hamilton (brute-force)**

```
import networkx as nx
from itertools import permutations

# Danh sách người và sở thích
preferences = {
    'A': {'B', 'C'},
    'B': {'A', 'D'},
    'C': {'A', 'D'},
    'D': {'B', 'C'}
}

# Tạo đồ thị từ preferences
G = nx.Graph()
for person, neighbors in preferences.items():
    for neighbor in neighbors:
        G.add_edge(person, neighbor)

# Hàm tìm chu trình Hamilton
def find_hamiltonian_cycle(graph):
    nodes = list(graph.nodes)
    for perm in permutations(nodes):
        # Kiểm tra chu trình Hamilton
```

```
        if all(graph.has_edge(perm[i], perm[i+1]) for i in range(len(perm)-1)) and
graph.has_edge(perm[-1], perm[0]):
            return perm + (perm[0],)
    return None

# Chạy thuật toán
cycle = find_hamiltonian_cycle(G)
if cycle:
    print("Chu trình Hamilton (chỗ ngồi hợp lý):", " -> ".join(cycle))
else:
    print("Không tìm được chu trình Hamilton phù hợp.")

# Vẽ đồ thị
import matplotlib.pyplot as plt
nx.draw(G, with_labels=True, node_color='lightgreen', node_size=800, font_size=12)
plt.title("Đồ thị sắp xếp chỗ ngồi theo sở thích")
plt.show()
```

✅ **Kết quả mẫu**

Chu trình Hamilton (chỗ ngồi hợp lý): A -> C -> D -> B -> A

→ Có thể sắp xếp A ngồi cạnh C, C cạnh D, D cạnh B và B cạnh A — thỏa điều kiện sở thích.

🧠 **Kết luận**

Thành phần thực tế	Tương ứng trong đồ thị
Người	Đỉnh (vertex)
Quan hệ muốn ngồi cạnh	Cạnh (edge)
Dãy chỗ ngồi hợp lý	Chu trình Hamilton trên đồ thị

- Mô hình hóa bài toán sắp xếp chỗ ngồi bằng đồ thị Hamilton giúp dễ dàng ứng dụng các thuật toán đồ thị để giải quyết bài toán xã hội/phân công một cách chính xác và tối ưu.

PHẦN 3: TRẮC NGHIỆM

🧪 **Câu hỏi trắc nghiệm (1–10)**

PHẦN 4.2.1 – Định nghĩa

🧪 **Câu 1 (Dễ):** Trong đồ thị, một đường đi Hamilton là:

A. Một đường đi qua mọi cạnh đúng một lần
B. Một đường đi qua mọi đỉnh đúng một lần
C. Một chu trình qua mọi cạnh đúng một lần
D. Một chu trình qua mọi đỉnh bất kỳ số lần nào

🧪 **Câu 2 (Dễ):** Một chu trình Hamilton là:

A. Một đường đi đi qua tất cả các cạnh
B. Một chu trình đi qua tất cả đỉnh và cạnh
C. Một đường đi đi qua tất cả đỉnh đúng một lần và quay về điểm xuất phát
D. Một chu trình có thể lặp lại đỉnh bất kỳ

🧪 **Câu 3 (Trung bình):** Đồ thị nào sau đây không chắc chắn là Hamilton?

A. Đồ thị đầy đủ với $n \geq 3$
B. Đồ thị liên thông và mỗi đỉnh bậc $\geq n/2$
C. Đồ thị có đúng 2 đỉnh bậc lẻ
D. Đồ thị thỏa mãn định lý Ore

PHẦN 4.2.2 – Định lý Rédei

🧪 **Câu 4 (Dễ):** Theo định lý Rédei, đồ thị định hướng nào sau đây luôn có đường đi Hamilton?

A. Đồ thị không có chu trình
B. Đồ thị định hướng mạnh liên thông
C. Đồ thị có số cạnh lẻ
D. Đồ thị có số đỉnh nguyên tố

PHẦN 4.2.3 – Định lý Dirac

🧪 **Câu 5 (Dễ):** Theo định lý Dirac (1952), nếu đồ thị có n đỉnh ($n \geq 3$) và mọi đỉnh đều có bậc $deg(v) \geq n/2$, thì:

A. Đồ thị có chu trình Euler
B. Đồ thị có ít nhất một chu trình Hamilton
C. Đồ thị có ít nhất một đường đi Hamilton
D. Đồ thị là đồ thị định hướng mạnh

🧪 **Câu 6 (Trung bình):** Giả sử một đồ thị có 10 đỉnh, mỗi đỉnh có bậc ≥ 5. Theo định lý Dirac, điều gì xảy ra?

A. Đồ thị không Hamilton
B. Chưa đủ điều kiện để kết luận

C. Đồ thị là Euler
D. Đồ thị có chu trình Hamilton

PHẦN 4.2.4 – Định lý Ore

🧪 **Câu 7 (Trung bình):** Định lý Ore nói rằng: Nếu với mọi cặp đỉnh không kề nhau u và v, ta có deg(u) + deg(v) ≥ n thì:

A. Đồ thị có chu trình Euler
B. Đồ thị có đường đi Hamilton
C. Đồ thị có chu trình Hamilton
D. Đồ thị định hướng mạnh liên thông

🧪 **Câu 8 (Khó):** Cho đồ thị có 6 đỉnh. Với mọi cặp đỉnh không kề nhau, tổng bậc của chúng ≥ 6. Có thể kết luận gì?

A. Không có chu trình Hamilton
B. Đồ thị có chu trình Euler
C. Đồ thị chắc chắn có chu trình Hamilton
D. Không thể kết luận gì

PHẦN 4.2.5 – Bài toán sắp xếp chỗ ngồi

🧪 **Câu 9 (Dễ):** Trong bài toán sắp xếp chỗ ngồi theo sở thích, nếu mỗi người muốn ngồi cạnh 2 người khác, bài toán có thể được mô hình hóa bằng:

A. Ma trận vuông cấp n
B. Đồ thị và tìm đường đi Euler
C. Đồ thị và tìm chu trình Hamilton
D. Danh sách tuần tự

Tổng hợp – So sánh Euler và Hamilton

🧪 **Câu 10 (Khó):** Phát biểu nào sau đây sai?

A. Mọi đồ thị đầy đủ có n ≥ 3 đều là Hamilton
B. Mọi đồ thị Euler đều là Hamilton
C. Một đồ thị có thể là Euler nhưng không Hamilton
D. Một đồ thị có thể là Hamilton nhưng không Euler

🧪 **Câu hỏi trắc nghiệm (11–20)**

PHẦN 4.2.1 – Định nghĩa (nâng cao)

🧪 **Câu 11 (Trung bình):** Điều nào sau đây là đúng về đường đi Hamilton?

A. Luôn tồn tại trong mọi đồ thị liên thông
B. Không thể tồn tại nếu đồ thị có chu trình
C. Có thể tồn tại ngay cả khi đồ thị không phải đầy đủ
D. Chỉ tồn tại nếu đồ thị có bậc đều nhau tại các đỉnh

🧪 **Câu 12 (Khó):** Đường đi Hamilton trong đồ thị có thể là chu trình khi nào?

A. Khi đỉnh đầu và cuối trùng nhau
B. Khi đồ thị đầy đủ
C. Khi tồn tại ít nhất 1 đỉnh bậc ≥ n/2
D. Khi đồ thị là Euler

PHẦN 4.2.2 – Định lý Rédei (nâng cao)

🧪 **Câu 13 (Trung bình):** Trong đồ thị định hướng mạnh liên thông với n đỉnh, kết luận nào sau đây chắc chắn đúng?

A. Có chu trình Euler
B. Có đường đi Hamilton
C. Có chu trình Hamilton
D. Có đỉnh bậc lẻ

PHẦN 4.2.3 – Định lý Dirac (nâng cao)

🧪 **Câu 14 (Khó):** Giả sử G là đồ thị đơn có 8 đỉnh. Nếu mỗi đỉnh có bậc ≥ 3, có thể kết luận gì?

A. G chắc chắn có chu trình Hamilton
B. G chắc chắn không có chu trình Hamilton
C. G có thể có hoặc không
D. G là Euler nếu có 2 đỉnh bậc lẻ

PHẦN 4.2.4 – Định lý Ore (nâng cao)

🧪 **Câu 15 (Khó):** Trong đồ thị G có 7 đỉnh, nếu tồn tại cặp đỉnh không kề nhau u, v mà deg(u) + deg(v) = 6, thì:

A. G thỏa mãn điều kiện Ore
B. G không thỏa mãn Ore
C. G chắc chắn không Hamilton
D. G là đồ thị đầy đủ

PHẦN 4.2.5 – Bài toán sắp xếp chỗ ngồi (nâng cao)

🧪 **Câu 16 (Trung bình):** Trong bài toán sắp xếp chỗ ngồi, nếu mô hình hóa bằng đồ thị và không tồn tại chu trình Hamilton, điều gì có thể xảy ra?

A. Tồn tại đường đi Hamilton
B. Không tồn tại cách sắp xếp vòng tròn hợp lý
C. Vẫn có thể xếp chỗ bằng thuật toán greedy
D. Chuyển bài toán sang tìm chu trình Euler

🧪 **Câu 17 (Khó):** Nếu mô hình bài toán sắp xếp chỗ ngồi bằng đồ thị có chu trình Hamilton, điều nào sau đây không đúng?

A. Có ít nhất một cách xếp chỗ theo vòng tròn
B. Có thể dùng brute-force để tìm giải
C. Luôn tồn tại duy nhất một chu trình Hamilton
D. Có thể dùng backtracking để kiểm tra

SO SÁNH EULER – HAMILTON (nâng cao)

🧪 **Câu 18 (Trung bình):** Đâu là điểm khác biệt cốt lõi giữa đồ thị Euler và Hamilton?

A. Euler đi qua đỉnh, Hamilton đi qua cạnh
B. Euler yêu cầu đồ thị đầy đủ
C. Euler đi qua cạnh, Hamilton đi qua đỉnh
D. Hamilton chỉ áp dụng cho đồ thị định hướng

🧪 **Câu 19 (Khó):** Cho đồ thị có 4 đỉnh, là một chu trình đơn: A–B–C–D–A. Phát biểu nào đúng?

A. Đồ thị có chu trình Hamilton và chu trình Euler
B. Chỉ có chu trình Hamilton
C. Chỉ có chu trình Euler
D. Không có chu trình nào cả

Ứng dụng thực tế & thuật toán

🧪 **Câu 20 (Khó):** Vì sao việc kiểm tra chu trình Hamilton là bài toán khó tính toán (NP-đầy đủ)?

A. Vì thuật toán kiểm tra cần giải hệ phương trình
B. Vì không có thuật toán tổng quát chạy đa thức cho mọi n
C. Vì đồ thị cần phải đầy đủ mới kiểm tra được
D. Vì chu trình Hamilton không thể biểu diễn bằng ma trận

PHẦN 5: BÀI TẬP LUYỆN TẬP

📦 Bài 1 – Kiểm tra định lý Dirac.
Cho đồ thị G gồm 6 đỉnh, các bậc đỉnh là: 3, 4, 4, 3, 3, 3.
Hỏi G có thỏa mãn điều kiện của định lý Dirac không? Có thể kết luận G là đồ thị Hamilton không?

Phân tích và hướng dẫn:

1. Định lý Dirac áp dụng với đồ thị đơn, $n \geq 3$, và yêu cầu mọi đỉnh có bậc $deg(v) \geq n/2$.
2. Với $n = 6 \rightarrow$ điều kiện Dirac yêu cầu $deg(v) \geq 3$.
3. So sánh từng bậc đỉnh với ngưỡng này.
4. Sau đó, đánh giá xem có thể áp dụng định lý để kết luận Hamilton hay không, hay chỉ có thể suy luận một phần.

📦 Bài 2 – Kiểm tra định lý Ore

Cho đồ thị G có 5 đỉnh: A, B, C, D, E. Danh sách cạnh: AB, AC, BD, CD, DE.
Kiểm tra xem G có thỏa định lý Ore không. Có thể suy ra điều gì?

Phân tích và hướng dẫn:

1. Định lý Ore: Nếu với mọi cặp đỉnh không kề nhau (không có cạnh nối trực tiếp), tổng bậc của chúng $\geq n$, thì G là Hamilton.
2. Bước 1: Tính bậc của từng đỉnh dựa vào danh sách cạnh.
3. Bước 2: Xét tất cả cặp đỉnh không kề nhau, kiểm tra tổng bậc.
4. Cuối cùng, đưa ra kết luận: thỏa hay không, và nếu không thỏa, liệu có thể suy ra gì thêm không?

📦 Bài 3 – Tìm tất cả đường đi Hamilton

Với đồ thị G có 4 đỉnh: A, B, C, D và cạnh: AB, AC, AD, BC, BD, CD.
Liệt kê tất cả đường đi Hamilton bắt đầu từ A.

Phân tích và hướng dẫn:

- Đường đi Hamilton là dãy các đỉnh đi qua tất cả đỉnh đúng 1 lần.
- Đề bài yêu cầu bắt đầu từ A → chỉ xét các hoán vị bắt đầu từ A.
- Với 4 đỉnh → số lượng đường đi Hamilton có thể tối đa là $(n–1)! = 6$.

- Học sinh nên liệt kê tất cả các dãy có dạng: A → … → … → … và kiểm tra từng dãy xem các cặp liên tiếp có cạnh nối không.

Bài 4 – Tìm chu trình Hamilton

Cho đồ thị đơn K_5 (đồ thị đầy đủ 5 đỉnh).
Viết ra một chu trình Hamilton bất kỳ trong đồ thị này. Có bao nhiêu chu trình Hamilton khác nhau?

Phân tích và hướng dẫn:

- K_5 là đồ thị đầy đủ: mọi cặp đỉnh đều có cạnh nối → luôn có chu trình Hamilton.
- Chu trình Hamilton là đường đi Hamilton quay lại đỉnh đầu.
- Với n = 5, tổng số chu trình Hamilton có công thức:
  $(n-1)!/2(n-1)!/2$ (do quay vòng và hoán vị đối xứng được xem là trùng nhau).
- Yêu cầu: viết ra một chu trình cụ thể, sau đó dùng công thức để đếm số lượng khác nhau.

Bài 5 – So sánh Euler và Hamilton

Cho đồ thị G như sau:
Đỉnh: A, B, C, D
Cạnh: AB, BC, CD, DA, AC

a) G có chu trình Euler không?
b) G có chu trình Hamilton không?
c) Giải thích sự khác biệt qua đồ thị này.

Phân tích và hướng dẫn:

- Euler: Chu trình Euler yêu cầu tất cả đỉnh có bậc chẵn (với đồ thị vô hướng).
- Hamilton: Chu trình Hamilton yêu cầu tồn tại chu trình đi qua tất cả đỉnh đúng một lần.
- Vẽ đồ thị G hoặc phân tích cấu trúc bằng sơ đồ/ma trận kề để xác định:
  - Bậc từng đỉnh → Euler
  - Có thể xây dựng chu trình Hamilton?
- Cuối cùng, dùng ví dụ này để sinh viên thấy rõ sự khác biệt giữa điều kiện tồn tại Euler vs. Hamilton.

📦 Bài 6 – Viết code brute-force

Viết chương trình Python kiểm tra chu trình Hamilton cho đồ thị nhỏ ($n \leq 6$) với input là danh sách kề.

Phân tích và hướng dẫn:

- Ý tưởng: Duyệt tất cả hoán vị của n đỉnh, kiểm tra xem dãy hoán vị đó tạo thành chu trình Hamilton hay không.
- Các bước chính:
  - Nhập danh sách đỉnh và danh sách kề.
  - Sinh tất cả các hoán vị của các đỉnh (bắt đầu từ một đỉnh cố định để tránh lặp vòng đối xứng).
  - Kiểm tra tính hợp lệ của từng hoán vị (liên tiếp có cạnh nối).
  - Kiểm tra thêm cạnh từ đỉnh cuối về đầu để tạo chu trình.
- Gợi ý sử dụng: itertools.permutations, set, dict (dạng adjacency list).
- Lưu ý: Do tính chất NP-đầy đủ, phương pháp này chỉ áp dụng được cho $n \leq 6$.

📦 Bài 7 – Đánh giá đồ thị không Hamilton

Vẽ một đồ thị liên thông gồm 6 đỉnh sao cho nó không có chu trình Hamilton. Giải thích lý do.

Phân tích và hướng dẫn:

- Mục tiêu: thiết kế một đồ thị vẫn liên thông (có đường giữa mọi cặp đỉnh) nhưng không thể đi qua tất cả đỉnh đúng 1 lần rồi quay về.
- Một số cách tư duy:
  - Tạo một đỉnh nối vào đồ thị chính bằng 1 cạnh → trở thành "nút cụt".
  - Thiết kế đồ thị hình “chạc ba” hoặc "cánh tay rẽ nhánh" khiến việc quay vòng khó.
- Sau khi vẽ xong, phân tích tại sao không thể đi theo chu trình Hamilton:
  - Vì có đỉnh bậc 1?
  - Vì khi đi qua một nhánh thì không quay lại được?

- Nên mô tả đồ thị bằng sơ đồ hoặc danh sách cạnh.

🧱 Bài 8 – Áp dụng vào bài toán sắp xếp chỗ ngồi

Có 5 người: A, B, C, D, E.
Danh sách người muốn ngồi cạnh nhau:

- A thích B, C
- B thích C, D
- C thích D, E
- D thích E
- E thích A
  Hãy mô hình hóa bài toán thành đồ thị và xác định xem có thể sắp xếp vòng tròn hay không.

Phân tích và hướng dẫn:

- Bước 1 – Mô hình hóa:
  - Mỗi người là một đỉnh.
  - Nếu A thích B → vẽ cạnh AB.
  - Có thể hiểu là cạnh có hướng hoặc vô hướng (tùy bài toán yêu cầu quan hệ một chiều hay hai chiều).
- Bước 2 – Tìm chu trình Hamilton:
  - Đặt bài toán thành việc tìm chu trình Hamilton trong đồ thị vừa tạo.
  - Gợi ý vẽ sơ đồ đồ thị để sinh viên trực quan hóa.
- Nếu đồ thị có chu trình Hamilton → có thể sắp xếp chỗ theo vòng tròn thỏa điều kiện.

🧱 Bài 9 – Phân tích thuật toán

Giả sử có thuật toán kiểm tra chu trình Hamilton bằng cách duyệt toàn bộ hoán vị.
a) Độ phức tạp thời gian là bao nhiêu?
b) Với n = 10, số lượng hoán vị cần kiểm tra là bao nhiêu?

Phân tích và hướng dẫn:

- a) Phân tích độ phức tạp:
  - Với n đỉnh → số lượng đường đi có thể: (n–1)! hoặc n!
  - Với mỗi hoán vị → cần kiểm tra tính liên thông liên tiếp → O(n)
  - → Tổng độ phức tạp: O(n! · n)
- b) Với n = 10:
  - Tính giá trị 10! → học sinh nên tính bằng tay hoặc dùng Python.
- Nhấn mạnh: phương pháp này không hiệu quả cho n lớn, minh chứng cho tính NP-đầy đủ của bài toán.

🧱 Bài 10 – Ứng dụng thực tiễn

Nêu một bài toán thực tế (khác sắp xếp chỗ ngồi) có thể mô hình hóa bằng chu trình Hamilton.
Mô tả cách mô hình hóa và giải bài toán đó bằng đồ thị.

Phân tích và hướng dẫn:
Gợi ý một số bài toán thực tế:

- Vấn đề du lịch (Traveling Salesman Problem):
  - Người bán hàng cần đi qua n thành phố 1 lần và quay về.
- Lập kế hoạch kiểm tra định kỳ thiết bị:
  - Một kỹ thuật viên phải kiểm tra tất cả thiết bị 1 lần.
- Lập tuyến đường xe buýt vòng quanh khu dân cư:
  - Xe buýt đi qua mỗi điểm dừng 1 lần rồi quay lại bến.

Hướng dẫn mô hình hóa:

- Mỗi địa điểm/thiết bị → 1 đỉnh.
- Mỗi cặp địa điểm có đường nối nếu đi được → cạnh.
- Tìm chu trình Hamilton tương ứng với đường đi tối ưu qua tất cả đỉnh đúng 1 lần.

PHẦN 6: BÀI TẬP DỰ ÁN

📊 Dự án 1 – Lập trình kiểm tra đồ thị có chu trình Hamilton

Viết chương trình kiểm tra một đồ thị (cho dưới dạng danh sách kề) có tồn tại chu trình Hamilton hay không.

- Input: Danh sách kề (adjacency list) hoặc ma trận kề của đồ thị đơn vô hướng ($n \leq 10$).
- Output:
  - Trả về một chu trình Hamilton nếu có (dưới dạng danh sách đỉnh).
  - Hoặc thông báo không tồn tại chu trình Hamilton.
- Ngôn ngữ lập trình: Python hoặc C++

🧠 Mục tiêu bài học:

- Vận dụng kiến thức về lý thuyết Hamilton vào giải bài toán thực tế bằng lập trình.
- Nắm được cách mô hình hóa đồ thị và áp dụng thuật toán brute-force hoặc backtracking.
- Phân biệt được hiệu suất và giới hạn tính toán (n nhỏ → dùng brute-force; n lớn → cần heuristic hoặc xấp xỉ).

🔍 Phân tích và hướng dẫn triển khai:

✅ **1.** Mô hình dữ liệu:

- Đồ thị lưu dưới dạng dictionary trong Python (hoặc vector<vector<int>> trong C++).
- Mỗi đỉnh gắn với danh sách các đỉnh kề.

✅ **2.** Ý tưởng thuật toán:

- Duyệt tất cả các hoán vị (hoặc dùng backtracking) để kiểm tra:
  - Mỗi đỉnh chỉ đi qua 1 lần.
  - Các đỉnh liên tiếp trong đường đi phải có cạnh nối.
  - Đỉnh cuối phải nối với đỉnh đầu để tạo chu trình.

✅ **3.** Các bước chính:

- Nhập danh sách kề từ người dùng hoặc từ file.
- Sinh các hoán vị bắt đầu từ một đỉnh cố định để giảm lặp.
- Kiểm tra hợp lệ của từng dãy hoán vị.
- Trả về chu trình nếu hợp lệ, ngược lại in ra “Không tồn tại”.

✅ **4.** Gợi ý mở rộng nâng cao:

- Hiển thị chu trình bằng đồ thị với thư viện networkx (Python).
- Kiểm thử với đồ thị K_4, K_5, lưới, hoặc đồ thị ngẫu nhiên.

📊 Dự án 2 – Ứng dụng đồ thị Hamilton vào bài toán sắp xếp chỗ ngồi

Có một danh sách các người tham dự và danh sách những người họ muốn/không muốn ngồi cạnh.
Yêu cầu: Tìm một cách sắp xếp chỗ ngồi theo vòng tròn sao cho mỗi người ngồi cạnh người họ thích (hoặc không ngồi cạnh người họ không thích).

- Input:
  - Danh sách người tham dự.
  - Danh sách các cặp quan hệ: “muốn ngồi gần” hoặc “không muốn ngồi gần”.
- Output:
  - Một dãy sắp xếp vòng tròn hợp lệ nếu có.
  - Hoặc thông báo “Không thể sắp xếp”.
- Ngôn ngữ lập trình: Python, C++ hoặc JavaScript

🧠 Mục tiêu bài học:

- Mô hình hóa bài toán thực tế bằng cấu trúc đồ thị.
- Vận dụng lý thuyết chu trình Hamilton vào tình huống xã hội/phân công.
- Tư duy giải quyết bài toán có ràng buộc (constraint satisfaction).

🔍 Phân tích và hướng dẫn triển khai:

✅ **1.** Mô hình hóa bài toán bằng đồ thị:

- Mỗi người là một đỉnh.
- Nếu A muốn ngồi cạnh B → tạo cạnh AB.
- Nếu A không muốn ngồi cạnh B → không tạo cạnh, hoặc đánh dấu “cấm”.

✅ **2.** Chiến lược giải:

- Tìm chu trình Hamilton trên đồ thị có điều kiện (ràng buộc).
- Có thể dùng:
  - Brute-force (với n nhỏ).
  - Backtracking có pruning: bỏ qua đường đi nếu vi phạm điều kiện “không muốn ngồi gần”.
  - Heuristic: ưu tiên chọn người có nhiều “bạn muốn ngồi gần”.

✅ **3.** Các bước thực hiện:

- Nhập danh sách người và quan hệ.
- Xây dựng đồ thị:
  - Cạnh có hướng hoặc vô hướng tùy bài toán.
  - Có thể dùng ma trận kề hoặc danh sách kề.
- Triển khai hàm tìm chu trình Hamilton có kiểm tra điều kiện.
- Xuất kết quả: danh sách sắp xếp vòng hoặc thông báo lỗi.

✅ **4.** Gợi ý nâng cao:

- Cho phép người dùng nhập quan hệ qua file CSV.
- Tích hợp đồ thị tương tác (sử dụng matplotlib, networkx hoặc Tkinter).
- Thử nghiệm với các bộ quan hệ mẫu: hợp tác, đối kháng, trung lập.