TRƯỜNG ĐẠI HỌC TRÀ VINH
KHOA KĨ THUẬT VÀ CÔNG NGHỆ
BỘ MÔN CÔNG NGHỆ THÔNG TIN

TÀI LIỆU GIẢNG DẠY
MÔN LÝ THUYẾT ĐỒ THỊ

GV biên soạn: Trầm Hoàng Nam

Trà Vinh, 01 / 2013

Lưu hành nội bộ

MỤC LỤC

CHƯƠNG 1
ĐẠI CƯƠNG VỀ ĐỒ THỊ
BÀI 1
ĐỒ THỊ VÀ SỰ BIỂU DIỄN ĐỒ THỊ

❖ **Mục tiêu học tập:** Sau khi học xong bài này, người học có thể:

- Xác định các khái niệm liên quan đến đồ thị
- Biểu diễn được các dạng đồ thị cơ bản

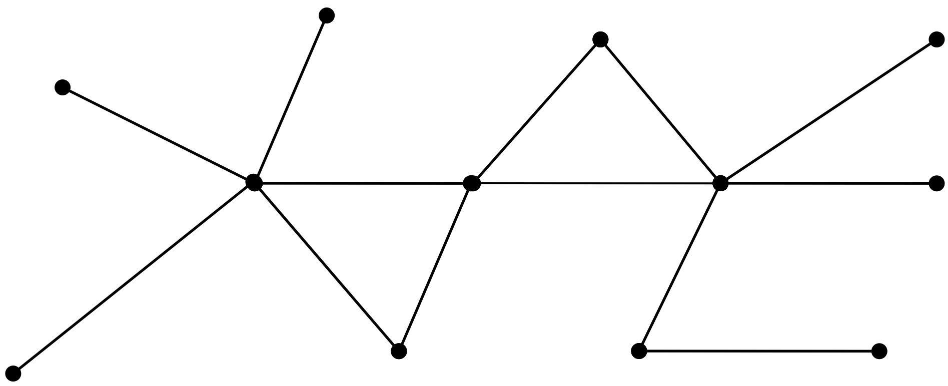
1. Mở đầu

Lý thuyết đồ thị là một lĩnh vực đã có từ lâu và có nhiều ứng dụng hiện đại. Những tư tưởng cơ bản của lý thuyết đồ thị được đề xuất vào những năm đầu của thế kỷ 18 bởi nhà toán học lỗi lạc người Thụy Sỹ Leonhard Eurler. Chính ông là người đã sử dụng đồ thị để giải bài toán nổi tiếng về các cái cầu ở thành phố Konigsberg.

Đồ thị được sử dụng để giải các bài toán trong nhiều lĩnh vực khác nhau. Chẳng hạn, đồ thị có thể sử dụng để xác định các mạch vòng trong vấn đề giải tích mạch điện. Chúng ta có thể phân biệt các hợp chất hóa học hữu cơ khác nhau với cùng công thức phân tử nhưng khác nhau về cấu trúc phân tử nhờ đồ thị. Chúng ta có thể xác định hai máy tính trong mạng có thể trao đổi thông tin được với nhau hay không nhờ mô hình đồ thị của mạng máy tính. Đồ thị có trọng số trên các cạnh có thể sử dụng để giải các bài toán như: Tìm đường đi ngắn nhất giữa hai thành phố trong mạng giao thông. Chúng ta cũng còn sử dụng đồ thị để giải các bài toán về lập lịch, thời khóa biểu, và phân bố tần số cho các trạm phát thanh và truyền hình

2. Định nghĩa đồ thị

Đồ thị là một cấu trúc rời rạc bao gồm các đỉnh và các cạnh nối các đỉnh này. Chúng ta phân biệt các loại đồ thị khác nhau bởi kiểu và số lượng cạnh nối hai đỉnh nào đó của đồ thị. Để có thể hình dung được tại sao lại cần đến các loại đồ thị khác nhau, chúng ta sẽ nêu ví dụ sử dụng chúng để mô tả một mạng máy tính. Giả sử ta có một mạng gồm các máy tính và các kênh điện thoại (gọi tắt là kênh thoại) nối các máy tính này. Chúng ta có thể biểu diễn các vị trí đặt máy tính bởi các điểm và các kênh thoại nối chúng bởi các đoạn nối, xem hình sau



Hình 1. Sơ đồ mạng máy tính

Ta nhận thấy rằng trong mạng ở hình trên, giữa hai máy bất kỳ chỉ có nhiều nhất là một kênh thoại nối chúng, kênh thoại này cho phép liên lạc cả hai chiều và không có máy tính nào lại được nối với chính nó. Sơ đồ mạng máy cho trong hình 1 được gọi là đồ thị vô hướng. Ta đi đến định nghĩa sau:

2.1 Định nghĩa 1

Đồ thị vô hướng G = (V,E) bao gồm V là tập các đỉnh, và E là tập các cặp không có thứ tự gồm hai phần tử khác nhau của V gọi là các cạnh.

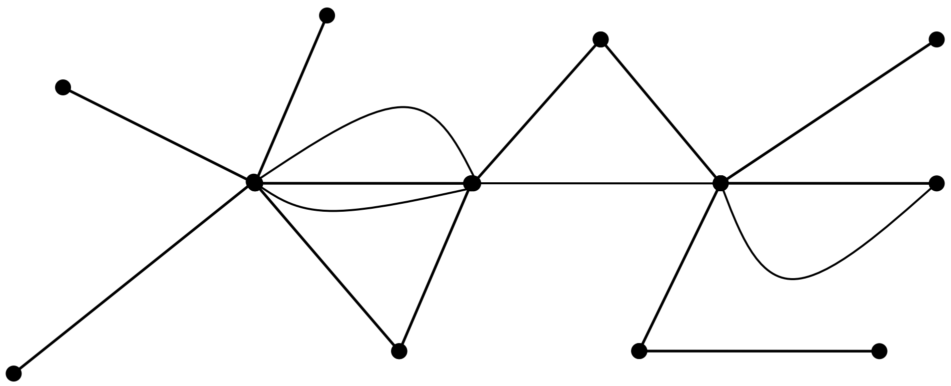
2.2 Định nghĩa 2

- Hai đỉnh được gọi là liền kề nhau nếu có cạnh nối hai đỉnh đó với nhau. Cạnh nối 2 đỉnh lại với nhau được gọi là cạnh liên thuộc.
- Hai cạnh được gọi là kề nhau nếu giữa chúng có đỉnh chung.
- Nếu e=(v,v) là một cạnh của đồ thị thì e được gọi là một khuyên.

2.3 Định nghĩa 3

- Nếu mỗi cạnh $e = (u, v)$ là không phân biệt thứ tự của các đỉnh u và v (từ u tới v không kể hướng) thì ta nói đồ thị G=(V,E) là đồ thị vô hướng.
- Nếu mỗi cạnh $e = (u, v)$ có phân biệt thứ tự của các đỉnh u và v (tức là từ u tới v khác từ v tới u) thì ta nói đồ thị G=(V,E) là đồ thị có hướng. Cạnh của đồ thị có hướng được gọi là cung.

Trong trường hợp giữa hai máy tính nào đó thường xuyên phải truyền tải nhiều thông tin người ta phải nối hai máy này bởi nhiều kênh thoại. Mạng với đa kênh thoại giữa các máy được cho trong dưới đây.



Hình 2. Sơ đồ mạng máy tính với đa kênh thoại

Từ đó ta có định nghĩa đơn đồ thị và đa đồ thị như sau:

2.4 Định nghĩa 4

- Đồ thị vô hướng G=(V,E) gọi là *đơn đồ thị* nếu giữa hai đỉnh bất kì của đồ thị được nối với nhau không quá một cạnh và mọi đỉnh đều không có khuyên.
- Đồ thị vô hướng G=(V,E) gọi là *đa đồ thị* nếu có ít nhất một cặp đỉnh nối với nhau bởi 2 cạnh e_1, e_2 trở lên và mọi đỉnh đều không có khuyên. Cạnh e_1 và e_2 được gọi là cạnh song song hay cạnh bội.
- Đồ thị vô hướng G=(V,E) gọi là giả đồ thị nếu nó là đa đồ thị có khuyên.

2.5 Định nghĩa 5

- Đồ thị có hướng G=(V,E) gọi là đơn đồ thị có hướng nếu giữa hai đỉnh bất kì của đồ thị được nối với nhau không quá một cung

  ☞ *Đơn đồ thị có hướng có thể có khuyên không?*
- Đồ thị có hướng G=(V,E) gọi là đa đồ thị có hướng nếu tồn tại 2 đỉnh khác nhau được nối với nhau bởi hai cung (cùng hướng) trở lên.

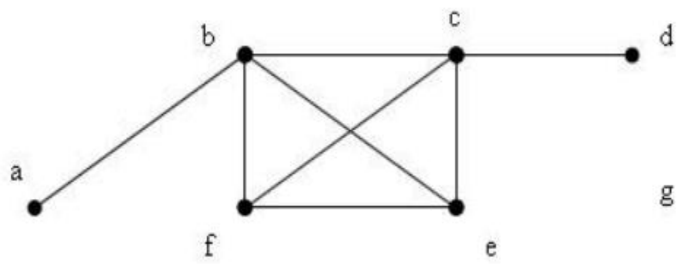
  ☞ *Đa đồ thị có hướng có thể có khuyên không?*

3. Bậc của đỉnh trong đồ thị

3.1 Đồ thị vô hướng

Định nghĩa

Đỉnh v của đồ thị G được gọi là có bậc n nếu v kề với n đỉnh khác (v là đầu mút của n cạnh). Ký hiệu: deg(v) hay d(v).



Hình 3. Đồ thị vô hướng G

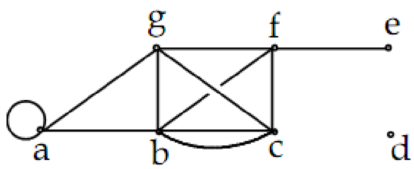
Ví dụ: xét ví dụ cho ở hình 3, ta có

deg(a)=deg(d)=1, deg(b)=deg(c)=4, deg(e)=deg(f)=3, deg(g)=0

☞ **Chú ý:**

- *Mỗi vòng (khuyên) tại v được kể là 2 cạnh tới v.*
- *Đỉnh có bậc 0 gọi là đỉnh cô lập (isolated vertex). (đỉnh g)*
- *Đỉnh có bậc 1 gọi là đỉnh treo (pendant vertex). (đỉnh a,d)*
- *Cạnh tới đỉnh treo gọi là cạnh treo (pendant edge). (cạnh ab và cd)*
- *Đồ thị mà mọi đỉnh đều là đỉnh cô lập gọi là đồ thị rỗng (null graph).*

Ví dụ: Xác định bậc của đỉnh trong đồ thị sau đây:



Hình 4. Đồ thị vô hướng có khuyên

3.1.1 Định lý 1

Trong mọi đồ thị G = (V, E), tổng số bậc của các đỉnh của G bằng 2 lần số cạnh. Nghĩa là:

$$\sum_{i=1}^{|V|} \deg(v_i) = 2|E|$$

Hệ quả: Trong mọi đồ thị G=(V,E), ta có:

- Số các đỉnh bậc lẻ là một số chẵn
- Tổng bậc của các đỉnh bậc lẻ là một số chẵn

3.1.2 Định lý 2

Trong mọi đồ thị không khuyên G=(V,E) có $|V| \geq 2$ thì tồn tại ít nhất hai đỉnh cùng bậc.

3.1.3 Định lý 3

Trong mọi đồ thị G=(V,E) có $|V| = n \geq 2$, có đúng hai đỉnh cùng bậc thì hai đỉnh này không thể đồng thời có bậc 0 hoặc bậc n-1.

Ví dụ 1: Chứng minh rằng trong một cuộc họp tùy ý có ít nhất 02 đại biểu tham gia trở lên, luôn luôn có ít nhất hai đại biểu mà họ có số người quen bằng nhau trong các đại biểu đã đến dự họp.

Chứng minh:

Bước 1: Xây dựng đồ thị G = (V, E) mô tả đầy đủ các thông tin của bài toán:

+ Đỉnh: Lấy các điểm trên mặt phẳng hay trong không gian tương ứng với các đại biểu đến dự họp. Đối tượng của bài toán ở đây là đại biểu dự họp. Vậy, mỗi đỉnh $v \in V$ biểu diễn cho một đại biểu trong cuộc họp.

+ Cạnh: Trong đồ thị G các đỉnh v_i và v_j được nối với nhau bằng một cạnh nếu hai đại biểu v_i và v_j quen nhau. Vậy, mối quan hệ giữa 02 đối tượng ở đây là mối quan hệ quen biết. Mỗi cạnh $e \in E$ nối 2 đỉnh v_i và v_j trong G nếu hai đại biểu v_i và v_j quen nhau.

Bước 2: Suy luận để suy ra điều cần chứng minh:

+ Với cách xây dựng đồ thị G như đã trình bày thì số đỉnh của G chính là số đại biểu đến dự họp $|V| \geq 2$ và bậc của mỗi đỉnh trong G bằng đúng số đại biểu quen với đại biểu được biểu diễn bằng đỉnh này.

+ Theo định lý 2 ta có trong G tồn tại ít nhất 02 đỉnh có cùng bậc nghĩa là luôn luôn có ít nhất hai đại biểu mà họ có số người quen bằng nhau trong các đại biểu đã đến dự họp.

Ví dụ 2: Chứng minh rằng số người mà mỗi người đã có một số lẻ lần bắt tay nhau trên trái đất này là một con số chẵn.

3.2 Đồ thị có hướng

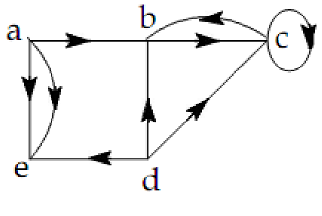
Định nghĩa: Cho G là đồ thị có hướng

+ *Bậc vào* của đỉnh v, ký hiệu: $\deg^{-}(v)$ (hoặc $d_{in}(v)$) là số cung có đỉnh cuối là v.

+ *Bậc ra* của đỉnh v, ký hiệu: $\deg^{+}(v)$ (hoặc $d_{out}(v)$) là số cung có đỉnh đầu là v.

☞ ***Chú ý:*** Một vòng tại một đỉnh sẽ góp thêm một đơn vị vào bậc vào và một đơn vị bậc ra của đỉnh này.

Ví dụ: Tìm bậc vào và bậc ra của các đỉnh trong đồ thị sau:



Hình 5. Đồ thị có hướng

Gợi ý: $d_{in}(a)=0$, $d_{out}(a)=3$.

Định lý:

Cho G = (V,E) là một đồ thị có hướng. Tổng bậc vào của các đỉnh bằng tổng bậc ra và bằng số cạnh của đồ thị. Nghĩa là:

$$\sum_{i=1}^{|V|}\deg^{-}(v_i)=\sum_{i=1}^{|V|}\deg^{+}(v_i)=|E|$$

Một đồ thị có hướng được gọi là cân bằng (balanced) nếu mọi đỉnh của nó đều có bậc vào và bậc ra bằng nhau.

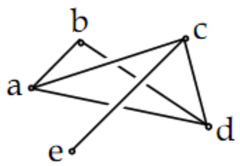
4. Biểu diễn đồ thị

4.1 Biểu diễn hình học

Người ta thường biểu diễn hình học của đồ thị như sau:

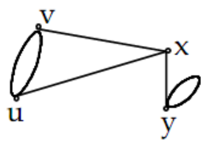
- Biểu diễn mỗi đỉnh của đồ thị bằng một điểm (vòng tròn nhỏ, ô vuông nhỏ).
- Một cạnh được biểu diễn bởi một đường (cong hay thẳng) nối 2 đỉnh liên thuộc với cạnh đó.

Ví dụ 1: Đồ thị G có: V = {a, b, c, d, e}, E = {ab, ac, ad, bd, cd, ce} được biểu diễn hình học như sau:



Hình 6. Đơn đồ thị được biểu diễn bằng hình học

Ví dụ 2: Đồ thị G có V = {u, v, x, y}, E = {uv, uv, ux, vx, xy, yy} được biểu diễn hình học như sau:



Hình 7. Đa đồ thị được biểu diễn bằng hình học

4.2 Biểu diễn bằng ma trận

4.2.1 Ma trận kề

Cho G = (V,E) có n đỉnh $v_1, v_2, ..., v_n$. Ma trận liền kề của G tương ứng với thứ tự các đỉnh $v_1, v_2, ..., v_n$ là một ma trận vuông cấp n, $A = (a_{ij})_n$ trong đó:

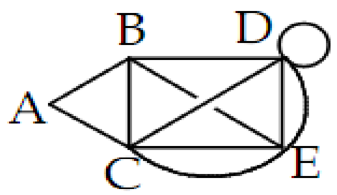
$$a_{ij}=\begin{cases}0 & \text{nếu } v_iv_j \text{ không là một cạnh của G}\\ 1 & \text{nếu } v_iv_j \text{ là một cạnh của G}\end{cases}$$

Chú ý:

- Ma trận kề của một đồ thị khác nhau tùy thuộc vào thứ tự liệt kê các đỉnh. Do đó, có tới n! ma trận kề khác nhau của một đồ thị n đỉnh vì có n! cách sắp xếp n đỉnh.

- Ma trận kề của một đồ thị là một ma trận đối xứng vì nếu v_i được nối với v_j thì v_j cũng được nối v_i và ngược lại
- Một vòng được tính là một cạnh từ đỉnh v vào đỉnh v.

Ví dụ 1:



Đồ thị có ma trận kề là

$$\begin{array}{c|c} & \begin{matrix} A & B & C & D & E \end{matrix} \\ \begin{matrix} A \\ B \\ C \\ D \\ E \end{matrix} & \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 2 \\ 0 & 1 & 1 & 1 & 2 \\ 0 & 1 & 2 & 2 & 0 \end{pmatrix} \end{array}$$

Ví dụ 2:

Vẽ đồ thị có ma trận kề sau:

$$\begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

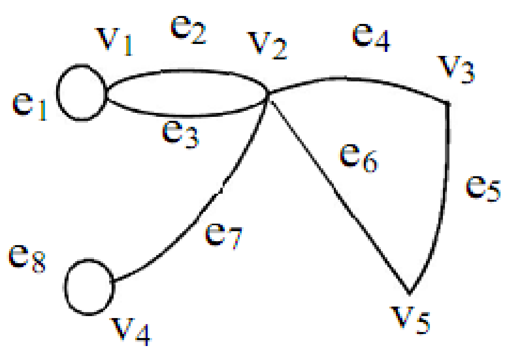
4.2.2 Ma trận liên thuộc

Người ta còn dùng ma trận liên thuộc để biểu diễn đồ thị. Cho G = (V,E) là một đồ thị với $v_1, v_2, ..., v_n$ là các đỉnh và $e_1, e_2, ..., e_m$ là các cạnh của G. Khi đó ma trận liên thuộc của G theo thứ tự trên của V và E là một ma trận $M = (m_{ij})_{nxm}$ với:

$$m_{ij} = \begin{cases} 0 & \text{nếu cạnh } e_j \text{ không nối với đỉnh } v_i \\ 1 & \text{nếu cạnh } e_j \text{ nối với đỉnh } v_i \end{cases}$$

☞ ***Chú ý:*** Các ma trận liên thuộc cũng có thể được dùng để biểu diễn các cạnh bội (song song) và khuyên (vòng). Các cạnh bội được biểu diễn trong ma trận liên thuộc bằng cách dùng các cột có các phần tử giống hệt nhau vì các cạnh này được nối với cùng một cặp các đỉnh. Các vòng được biểu diễn bằng cách dùng một cột với đúng một phần tử bằng 1 tương ứng với đỉnh nối với khuyên đó.

Ví dụ: Cho đồ thị:



Đồ thị trên có ma trận liên thuộc như sau:

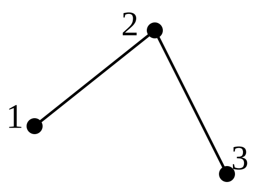
$$
\begin{array}{c|cccccccc|}
 & e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 \\
\hline
v_1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
v_2 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\
v_3 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
v_4 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
v_5 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0
\end{array}
$$

4.2.3 Ma trận trọng số

Cách biểu diễn đồ thị bằng ma trận trọng số được áp dụng trong trường hợp các cạnh (cung) của đồ thị được gán giá trị là một số w(e), ta gọi w(e) là trọng lượng của e. Khi đó ma trận trọng số của G là ma trận vuông cấp n, $W = (w_{ij})_n$ với:

$$w_{ij} = \begin{cases} 0 & \text{nếu } v_i v_j \text{ không là một cạnh của G} \\ w(e_{ij}) & \text{nếu } v_i v_j \text{ là một cạnh của G} \end{cases}$$

Ví dụ:



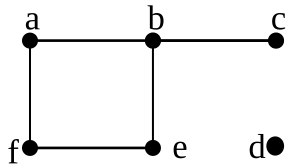
Ma trận trọng số là $\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 5 \\ 0 & 5 & 0 \end{bmatrix}$

Ngoài ra ta còn có thể dùng danh sách liên kết để biểu diễn đồ thị (SV tự xem ở tài liệu tham khảo)

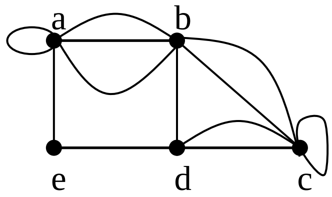
❖ **Bài tập củng cố:**

Bài 1. Tìm số đỉnh, số cạnh, bậc của mỗi đỉnh trong các đồ thì vô hướng sau: (chỉ rõ đỉnh cô lập và đỉnh treo, nếu có)

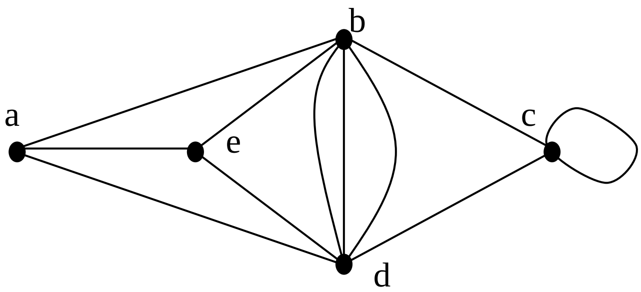
a.



b.

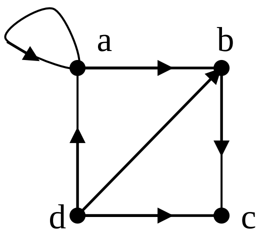


c.

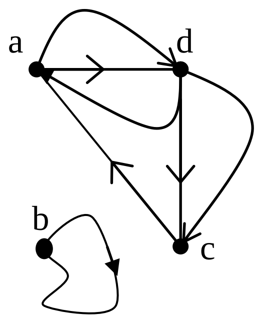


Bài 2. Xác định số đỉnh, số cạnh, số bậc vào và số bậc ra của mỗi đỉnh đối với các đồ thị có hướng sau:

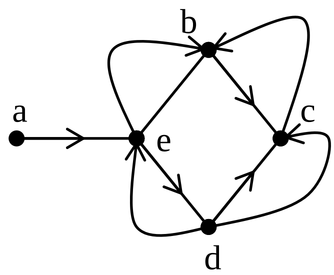
a.



b.



c.



Bài 3. Tính số đỉnh của một đồ thị đều bậc 4 và có 10 cạnh.

Bài 4. Vẽ đồ thị vô hướng và đồ thị có hướng cho bởi G=(V,E)

+ V={A,B,C,D,E,G,H}

+ G={(A,B),(B,C),(A,C),(G,H),(H,E),(E,A),(D,A)}

Bài 5. Một đồ thị vô hướng có các đỉnh có các bậc lần lượt là: 4, 3, 3, 2, 2. Tính số cạnh và vẽ đồ thị này.

Bài 6. Vẽ đơn đồ thị có 6 đỉnh, trong đó có

a. 3 đỉnh bậc 3 và 3 đỉnh bậc 1

b. Bậc các đỉnh là 1, 2, 3, 3, 4, 5

c. Bậc các đỉnh là 2, 2, 4, 4, 4, 4

Bài 7. Tìm số đỉnh của đồ thị G biết rằng G có:

a. 12 cạnh và mỗi đỉnh đều có bậc 2

b. 15 cạnh, 3 đỉnh bậc 4 và các đỉnh còn lại bậc 3

Bài 8. Có thể có một nhóm 9 người trong đó mỗi người đều chỉ quen biết đúng 5 người khác trong nhóm hay không?

Bài 9. Một đồ thị có 100 đỉnh, mỗi đỉnh đều có bậc 50. Tính số cạnh của đồ thị này.

Bài 10. Trong một cuộc liên hoan, mọi người bắt tay nhau. Chứng minh rằng: tổng số lượt người được bắt tay là một số chẵn (quy ước không ai tự bắt tay mình)

Bài 11. Có bao nhiêu cạnh trong một đồ thị có 6 đỉnh mà hai đỉnh bậc 4, hai đỉnh bậc 6, hai đỉnh bậc 8

Bài 12. Vẽ các đồ thị có hướng biểu diễn bằng các ma trận liền kề sau:

a. $\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$ b. $\begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$ c. $\begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 2 & 0 \\ 1 & 1 & 1 & 0 \end{bmatrix}$

Bài 13. Vẽ các đồ thị vô hướng biểu diễn bằng ma trận liền kề:

a. $\begin{bmatrix} 1 & 2 & 3 \\ 2 & 0 & 4 \\ 3 & 4 & 0 \end{bmatrix}$ b. $\begin{bmatrix} 1 & 2 & 0 & 1 \\ 2 & 0 & 3 & 0 \\ 0 & 3 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$

BÀI 2
ĐƯỜNG ĐI, CHU TRÌNH VÀ TÍNH LIÊN THÔNG

❖ **Mục tiêu học tập:** Sau khi học xong bài này, người học có thể:
- Xác định được đường đi, chu trình trong đồ thị
- Xác định được đồ thị liên thông

1. Đường đi và chu trình

1.1 Đường đi

Đường đi (path) có độ dài n từ v_o đến v_n (với n là một số nguyên dương) trong một đồ thị vô hướng là một dãy các cạnh liên tiếp $v_o v_1$, $v_1 v_2$, ... , $v_{n-1} v_n$. Đỉnh v_o được gọi là đỉnh đầu, đỉnh v_n được gọi là đỉnh cuối. Đường đi này thường được viết gọn: $v_o v_1 v_2 \ldots v_{n-1} v_n$. Khi chỉ cần nêu ra đỉnh đầu v_o và đỉnh cuối v_n của đường đi, ta viết: đường đi $v_o - v_n$.

- Một đường đi không qua cạnh nào lần thứ hai được gọi là ***đường đi đơn giản (đường đi đơn)***.
- Một đường đi không qua đỉnh nào lần thứ hai được gọi là ***đường đi sơ cấp***.

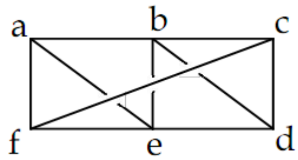
☞ ***Lưu ý:*** Một đường đi sơ cấp là một đường đi đơn giản nhưng một đường đi đơn giản có thể không là đường đi sơ cấp.

1.2 Chu trình

Một đường đi khép kín (đỉnh đầu trùng đỉnh cuối) và có độ dài n ≥ 3 được gọi là một chu trình (Cycle).

- Chu trình không đi qua cạnh nào lần thứ hai được gọi là ***chu trình đơn giản***.
- Chu trình không đi qua đỉnh nào lần thứ hai, trừ đỉnh đầu ≡ đỉnh cuối, được gọi là một ***chu trình sơ cấp***.

Ví dụ: Trong đồ thị sau:



Hình 1. Đường đi trên đồ thị

abcdbe là một đường đơn giản.

eabdc là một đường đi sơ cấp.

abea là một chu trình

Định lý: Cho G=(V,E) là một đồ thị vô hướng có $|V| \geq 4$ và $\forall v \in V$ có $\deg(v) \geq 3$ thì trong G luôn tồn tại một chu trình sơ cấp có độ dài chẵn.

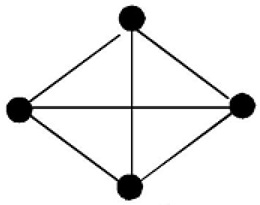
2. Tính liên thông của đồ thị

2.1 Đồ thị con và đồ thị bộ phận

Cho đồ thị G=(V,E)

- Nếu trong đồ thị G ta bỏ đi một số đỉnh nào đó và các cạnh chứa các đỉnh đó thì phần còn lại của đồ thị được gọi là đồ thị con của đồ thị G
- Nếu trong đồ thị G ta bỏ đi một số cạnh nào đó và giữ nguyên các đỉnh thì phần còn lại của đồ thị được gọi là đồ thị bộ phận của đồ thị G

Ví dụ: Tìm một vài đồ thị con và đồ thị bộ phận của đồ thị sau:

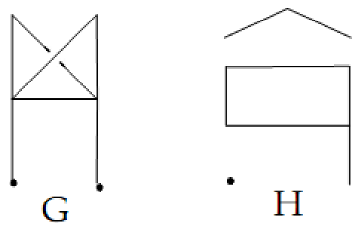


2.2 Tính liên thông trong đồ thị vô hướng

2.2.1 Định nghĩa

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

Ví dụ



G: liên thông; H: không liên thông.

Hình 2. Đồ thị liên thông G và đồ thị H gồm 3 thành phần liên thông

Ta gọi đồ thị con của G=(V,E) là đồ thị H=(W,F), trong đó $W \subseteq V$ và $F \subseteq E$

Trong trường hợp đồ thị là không liên thông, nó sẽ rã ra thành một số đồ thị con liên thông đôi một không có đỉnh chung. Những đồ thị con liên thông như vậy ta gọi là các ***thành phần liên thông của đồ thị****.*

Ví dụ: đồ thị sau có 3 thành phần liên thông

Hình 3. Các thành phần liên thông của đồ thị

2.2.2 Định lý 1

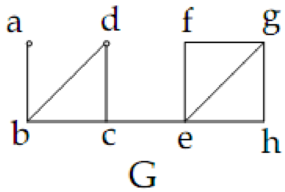
Đồ thị G=(V, E) là liên thông khi và chỉ khi G có duy nhất một thành phần liên thông.

2.2.3 Đỉnh cắt và cầu

***Đỉnh cắt*:** Nếu việc xóa đi một đỉnh $v \in V$ và tất cả các cạnh liên thuộc với nó sẽ tạo ra một đồ thị con mới có nhiều thành phần liên thông hơn đồ thị xuất phát. Các đỉnh v như thế được gọi là đỉnh cắt (cut point) hay điểm khớp.

***Cầu*:** Nếu trong đồ thị G ta bỏ đi cạnh e sẽ tạo ra nhiều thành phần liên thông hơn G thì e được gọi là cầu (bridge).

Ví dụ: Tìm các đỉnh cắt và cầu trong đồ thị:



Hình 4. Đồ thị chứa đỉnh cắt và cầu

☞ ***Chú ý:*** Không phải đồ thị nào cũng có đỉnh cắt và cầu.

2.2.4 Định lý 2

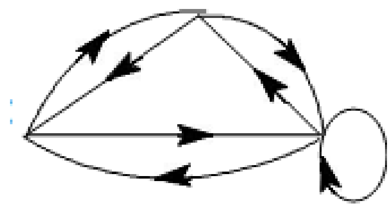
Trong mọi đồ thị G=(V,E) có $|V| \geq 2$, nếu $\forall v_1, v_2 \in V$ thỏa $\deg(v_1) + \deg(v_2) \geq n$ thì G là đồ thị liên thông.

Hệ quả: Trong mọi đồ thị G=(V, E) có n đỉnh, nếu mọi đỉnh $v \in V$ có $\deg(v) \geq \frac{n}{2}$ thì G là đồ thị liên thông.

2.3 Tính liên thông trong đồ thị có hướng

2.3.1 Liên thông mạnh (Strongly connected)

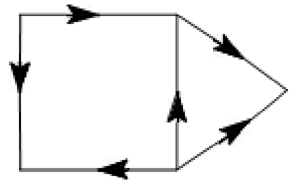
Đồ thị có hướng G được gọi là liên thông mạnh nếu có đường đi từ a đến b và từ b đến a; $\forall$ a, b $\in$ đồ thị.



Hình 5. Đồ thị liên thông mạnh

2.3.2 Liên thông yếu (Weakly connected)

Đồ thị có hướng G được gọi là liên thông yếu nếu đồ thị vô hướng tương ứng của nó là liên thông.



Hình 6. Đồ thị liên thông yếu

2.3.3 Định lý

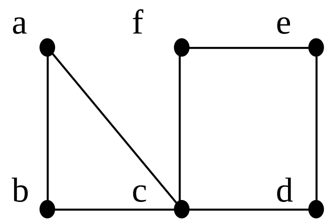
Nếu trong đồ thị G=(V, E) có đúng hai đỉnh bậc lẻ thì hai đỉnh này phải liên thông với nhau.

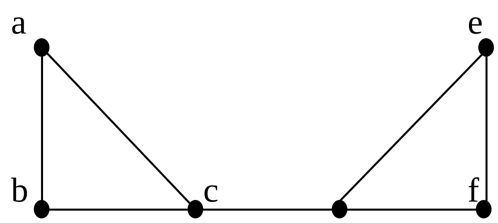
Chứng minh

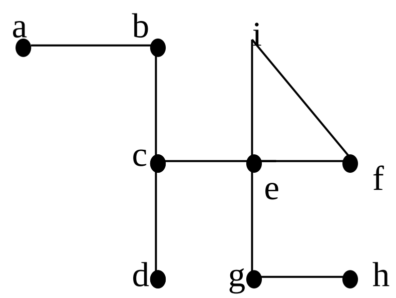
Giả sử đồ thị G=(V, E) có đúng hai đỉnh bậc lẻ v và w nhưng hai đỉnh này lại không liên thông với nhau. Khi đó v và w phải thuộc vào 2 thành phần liên thông G_1, G_2 khác nhau của G. Chẳng hạn $v \in G_1$ và $w \in G_2$. Theo giả thuyết do G chỉ có đúng 2 đỉnh bậc lẻ nên trong mỗi đồ thị con G_1 và G_2 chỉ có đúng một đỉnh bậc lẻ. Mâu thuẫn với tính chất số đỉnh bậc lẻ trong một đồ thị là một số chẵn. Vậy v và w phải liên thông với nhau.

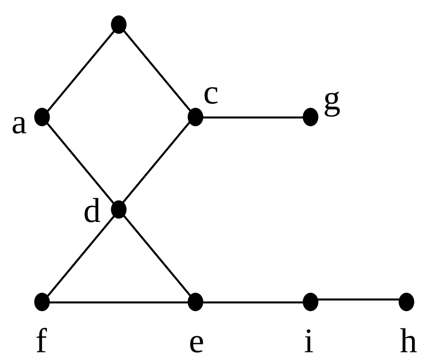
❖ **Bài tập củng cố:**

Bài 1. Tìm tất cả các đỉnh cắt và cầu của đồ thị :







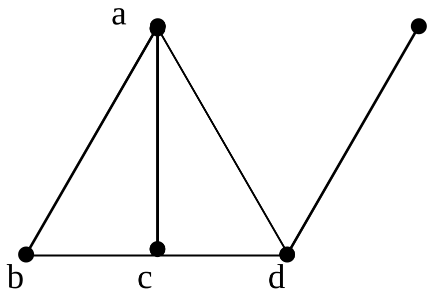


Bài 2 : Vẽ các đồ thị

a. Đều bậc 3 có 10 đỉnh.

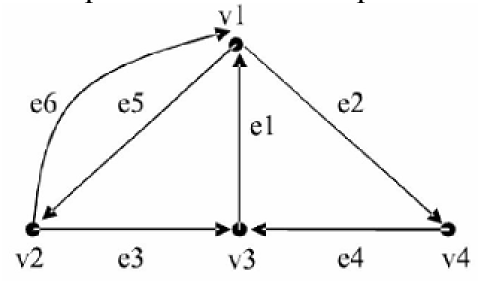
b. Đều bậc 4 có 8 đỉnh.

Bài3. Liệt kê tất cả các đồ thị con của đồ thị sau

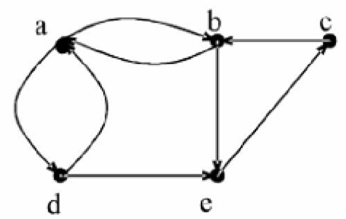


Bài 4. Một đồ thị có 19 cạnh và mỗi đỉnh đều có bậc lớn hơn hoặc bằng 3, hỏi đồ thị này có tối đa bao nhiêu đỉnh

Bài 5. Chỉ ra các đường đi sơ cấp và chu trình sơ cấp có thể có của đồ thị sau:



Bài 6: Cho đồ thị:



Mỗi danh sách các đỉnh sau đây có tạo nên đường đi trong đồ thị đã cho hay không?

a. a,b,e,c,b
b. a,d,b,e,a
c. a,d,a,d,a
d. a,b,e,c,b,d,a

BÀI 3

CÁC ĐỒ THỊ ĐẶC BIỆT VÀ SỰ ĐẲNG CẤU

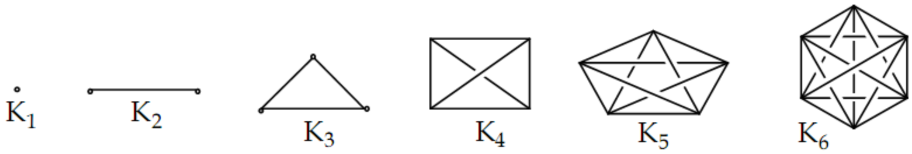
❖ **Mục tiêu học tập:** Sau khi học xong bài này, người học có thể:

- Phân biệt các loại đồ thị đặc biệt
- Xác định hai đồ thị đẳng cấu

1. Các đồ thị vô hướng đặc biệt

1.1 Đồ thị đầy đủ

Đồ thị đầy đủ (Complete graph), **ký hiệu: K_n** là một đơn đồ thị bao gồm n đỉnh mà mọi đỉnh đều có bậc n−1 (mỗi đỉnh đều nối với n−1 đỉnh còn lại).



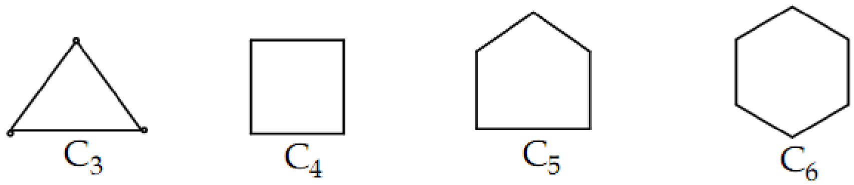
Hình 1. Đồ thị đầy đủ

Vậy K_n có:

- Số đỉnh: $|V| = n$
- Bậc của đỉnh: $\deg(v_i) = n-1, \forall v_i \in V$
- Số cạnh: $|E| = \dfrac{n*(n-1)}{2}$

1.2 Đồ thị vòng

Đồ thị vòng **ký hiệu: C_n,** $n \geq 3$ là một đồ thị với n đỉnh $v_1, v_2, ..., v_n$ và các cạnh v_1v_2, v_2v_3, …, v_nv_1.



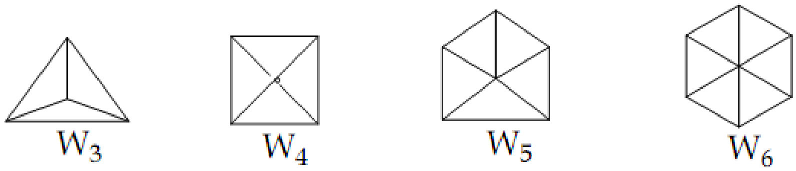
Hình 2. Đồ thị vòng

Vậy C_n có:

- Số đỉnh: $|V| \geq 3$
- Bậc của đỉnh $\deg(v_i) = 2 ; \forall v_i \in V$
- Số cạnh: $|E| = n$

1.3 Đồ thị hình bánh xe (Wheel graph)

Nếu thêm một đỉnh vào đồ thị vòng C_n $(n \geq 3)$ và nối đỉnh này với n đỉnh của C_n thì ta được đồ thị hình bánh xe, ký hiệu: W_n.



Hình 3. Đồ thị hình bánh xe

Vậy W_n có:

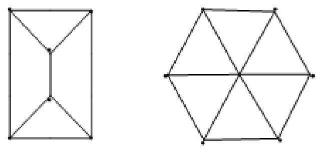
- Số đỉnh: $|V| = n+1, n \geq 3$
- Bậc của đỉnh $\deg(v_i) = 3$, và $\forall v_i \in V$ và $v_i \neq$ đỉnh được thêm vào (v_{new})
- $\deg(v_{new}) = n$
- Số cạnh: $|E| = n$

1.4 Đồ thị đều

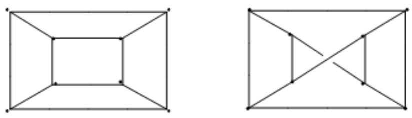
Một đồ thị đều (Regular graph) là đồ thị mà mọi đỉnh đều có cùng bậc. Nếu đồ thị G có các đỉnh có cùng bậc K thì được gọi là **đồ thị K-đều**.

Ví dụ:

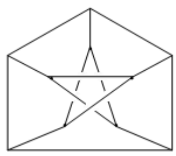
- Đồ thị rỗng gồm n đỉnh là đồ thị đều bậc 0.
- C_n là đồ thị đều bậc 2.
- K_n là đồ thị đều bậc (n−1).
- Đồ thị 3-đều 6 đỉnh:



- Đồ thị 3-đều 8 đỉnh:



- Đồ thị đều bậc 3: đồ thị Petersen:



Hình 4. Đồ thị đều

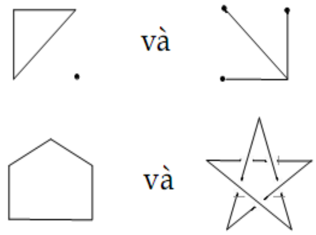
Vậy đồ thị k - đều n đỉnh có:

- Số đỉnh: $|V| = n$
- Bậc của đỉnh: $\deg(v_i) = k, \forall v_i \in V$
- Số cạnh: $|E| = \dfrac{n*k}{2}$

1.5 Đồ thị bù

Hai đơn đồ thị G và G' được gọi là bù với nhau nếu chúng có chung các đỉnh, cạnh nào thuộc G thì không thuộc G' và ngược lại. Ký hiệu: $G^{'} = \overline{G}$

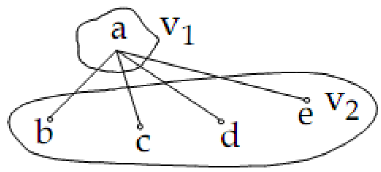
Ví dụ:



Hình 5. Đồ thị bù

1.6 Đồ thị lưỡng phân

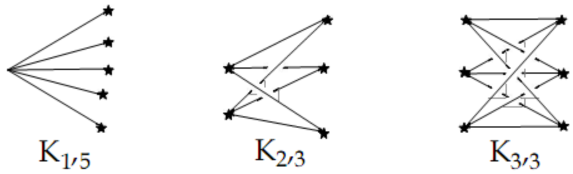
Một đồ thị G được gọi là đồ thị lưỡng phân (bipartie graph) nếu tập hợp các đỉnh V của G có thể phân thành 2 tập hợp không rỗng V_1 và V_2, $V_1 \cap V_2 = \varnothing$ sao cho mỗi cạnh của G nối một đỉnh của V_1 với một đỉnh của V_2.



Hình 6. Đồ thị lưỡng phân

Ví dụ:

- Một đồ thị lưỡng phân mà mỗi đỉnh của V_1 (có m đỉnh) đều kề với mọi đỉnh của V_2 (có n đỉnh), được gọi là một đồ thị lưỡng phân đầy đủ, ký hiệu: $K_{m,n}$.



Hình 7. Đồ thị lưỡng phân đầy đủ

- K_3 không phải là đồ thị lưỡng phân vì nếu ta chia các đỉnh của nó thành 2 phần rời nhau thì một trong 2 phần này phải chứa 2 đỉnh. Nếu đồ thị là lưỡng phân thì các đỉnh này không thể nối với nhau bằng một cạnh. Nhưng trong K_3 mỗi đỉnh được nối với đỉnh khác bằng một cạnh.

Định lý: **(Định lý về điều kiện cần và đủ của một đồ thị lưỡng phân)**

Đồ thị G = (V, E) là một đồ thị lưỡng phân khi và chỉ khi mọi chu trình của nó đều có độ dài chẵn.

Chứng minh

- Giả sử G = (V, E) là một đồ thị lưỡng phân và tập đỉnh V của G được chia thành hai tập con V_1 và V_2. Khi đó, dọc theo chu trình bất kỳ của G thì các đỉnh thuộc tập V_1 và tập V_2 sẽ lần lượt nằm liên tiếp và xen kẻ nhau. Do đó, khi trở về đỉnh xuất phát đầu tiên, ta phải đi qua một số chẵn các đỉnh và do đó chiều dài của chu trình là một số chẵn.

- Ngược lại, giả sử rằng G là một đồ thị có tính chất là tất cả các chu trình của G đều có độ dài chẵn. Ta sẽ chứng minh tất cả các thành phần liên thông của G đều là đồ thị lưỡng phân và do đó G cũng là một đồ thị lưỡng phân. Thật vậy, giả sử rằng G_1 là một thành phần liên thông của G và v_0 là một đỉnh của G_1. Với mỗi đỉnh $v \in G_1$ ta chọn một đường α nối v và v_0. Nếu đường α có độ dài chẵn thì ta đưa đỉnh v vào tập đỉnh V_1. ngược lại, nếu α có độ dài lẻ thì ta đưa v vào tập đỉnh V_2. Sự phân loại các đỉnh của G_1 không phụ thuộc vào cách chọn đường đi α. Thật vậy, nếu có hai đường α có độ dài chẵn và α' có độ dài lẻ nối 2 đỉnh v và v_0 thì đồ thị G_1 sẽ có chu trình với độ dài lẻ, mâu thuẫn với tính chất ban đầu là G chỉ có chu trình độ dài chẵn.
- Với cách thiết lập hai tập hợp đỉnh V_1 và V_2 này, các đỉnh của đồ thị G_1 hoặc thuộc tập hợp đỉnh V_1 hoặc thuộc tập hợp đỉnh V_2. Bây giờ, ta chứng minh rằng chỉ có các cạnh nối các đỉnh không thuộc cùng một tập hợp đỉnh với nhau mà thôi. Thật vậy, giả sử rằng có 2 đỉnh v và u kề nhau trong G_1 thì chúng không thể thuộc cùng một tập hợp đỉnh V_1 hoặc V_2, nếu không ta có thể đi từ đỉnh v_0 đến đỉnh v rồi đi đến đỉnh u bằng cạnh vu rồi trở về đỉnh v_0 bằng một đường đi có độ dài lẻ. Điều này không xảy ra trong đồ thị G. Vậy G là đồ thị lưỡng phân với hai tập đỉnh rời nhau là V_1 và V_2 bằng cách mà ta đã xây dựng trên.

2. Sự đẳng cấu của đồ thị

Các đồ thị $G_1=(V_1,E_1)$ và $G_2=(V_2,E_2)$ được gọi là đẳng cấu với nhau nếu có một song ánh f: $V_1 \rightarrow V_2$ sao cho nếu a và b là liền kề trong V_1 thì f(a) và f(b) liền kề trong V_2; $\forall$ a, b $\in V_1$. Khi đó song ánh f được gọi là một đẳng cấu.

Nói cách khác, nếu 2 đồ thị là đẳng cấu thì sẽ tồn tại một song ánh giữa các đỉnh của 2 đồ thị bảo toàn quan hệ liền kề.

☞ ***Chú ý:*** Nếu 2 đồ thị G_1 và G_2 là đẳng cấu thì chúng có:

- Số đỉnh bằng nhau.
- Số cạnh bằng nhau.
- Hai đỉnh tương ứng có cùng bậc.

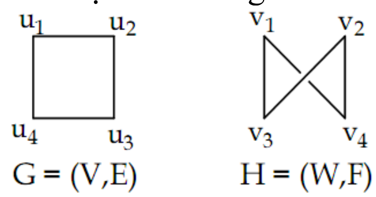
Đây là các **điều kiện cần** để hai đồ thị là đẳng cấu.

Để chứng minh hai đồ thị đẳng cấu ta cần:

- Chứng minh điều kiện cần thỏa.

- Xây dựng một song ánh bảo toàn quan hệ liền kề giữa hai đồ thị (điều kiện đủ để hai đồ thị đẳng cấu).

Ví dụ 1: Chứng minh rằng hai đồ thị sau là đẳng cấu với nhau:



Hình 8. Hai đồ thị đẳng cấu

***Xét điều kiện cần*:**

- Hai đồ thị G và H đều có 4 đỉnh,
- Hai đồ thị G và H đều có 4 cạnh,
- Các đỉnh của hai đồ thị đều có bậc 2.

Vậy điều kiện cần thỏa.

***Xét điều kiện đủ*:**

Xét hàm f: V $\rightarrow$ W

$u_1 \mapsto v_1$

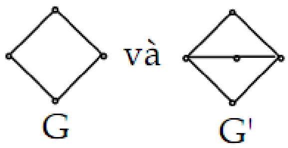
$u_2 \mapsto v_4$

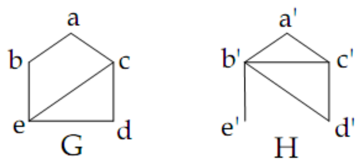
$u_3 \mapsto v_2$

$u_4 \mapsto v_3$

$\Rightarrow$ f là song ánh và bảo toàn quan hệ liền kề, điều kiện đủ thỏa. Vậy hai đồ thị G và H đẳng cấu với nhau.

Ví dụ 2: Hai đồ thị sau đây có đẳng cấu không?





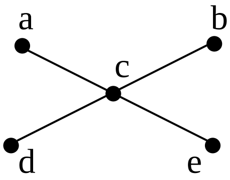
❖ **Bài tập củng cố:**

Bài 1. Hãy vẽ các đồ thị :

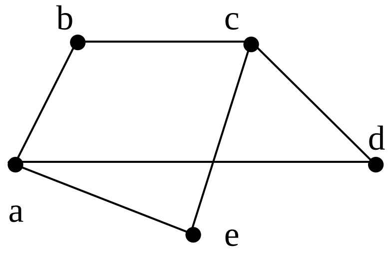
a. K_7 b. $K_{1,8}$ c. $K_{4,4}$ d. C_7 e. W_7

Bài 2. Xét xem các đồ thị sau có là đồ thị lưỡng phân không:

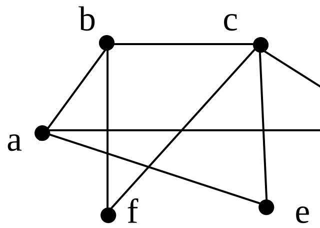
a.



b.



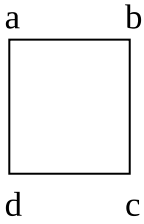
c.



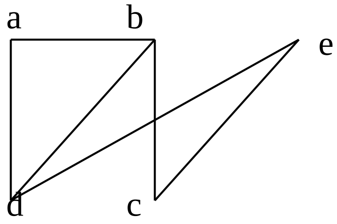
Bài 3. Nếu đơn đồ thị G có 15 cạnh và $\overline{G}$ có 13 cạnh khi đó G và $\overline{G}$ có bao nhiêu đỉnh?

Bài 4. Viết ma trận kề của các đồ thị sau

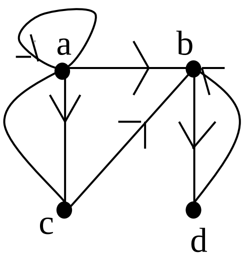
a.



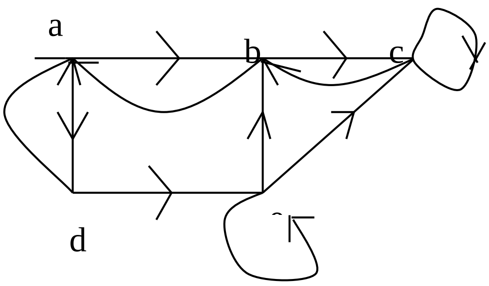
b.



c.



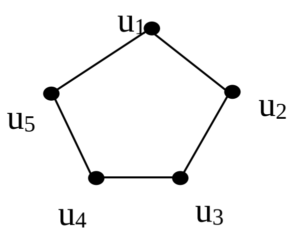
d.

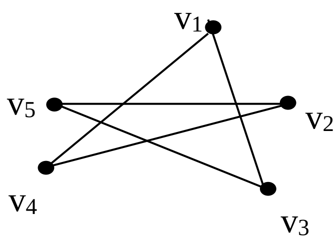


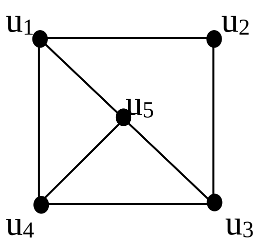
Bài 5. Biểu diễn các đồ thị sau bằng ma trận liền kề:

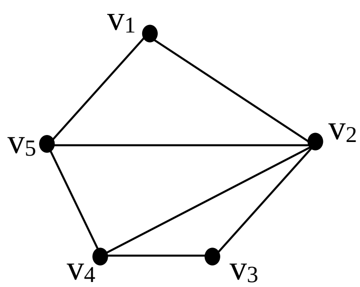
a. K_4 b. $K_{1,4}$ c. $K_{2,3}$ d. C_4 e. W_4

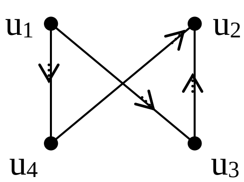
Bài 6. Hãy xét xem các cặp đồ thị sau có đẳng cấu với nhau không?

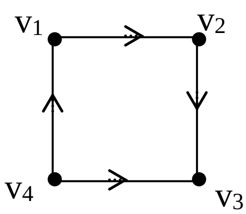
a. 

và 

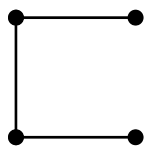
b. 

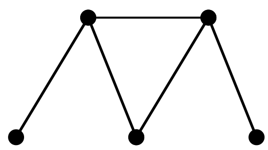
và 

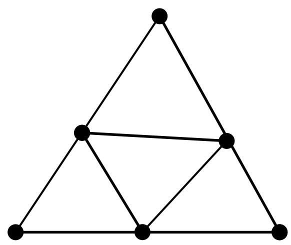
c. 



Bài 7. Vẽ đồ thị bù của các đồ thị sau:







CHƯƠNG 2
CÁC BÀI TOÁN VỀ ĐƯỜNG ĐI

<u>BÀI 1</u>
CÁC THUẬT TOÁN TÌM KIẾM TRÊN ĐỒ THỊ

❖ **Mục tiêu học tập:** Sau khi học xong bài này, người học có thể:
- Biết cách duyệt đồ thị theo chiều rộng và chiều sâu
- Áp dụng duyệt đồ thị để làm bài tập thực hành

1. Tìm kiếm theo chiều sâu (Depth First Search - DFS)

Ý tưởng chính của thuật toán có thể trình bày như sau. Ta sẽ bắt đầu tìm kiếm từ một đỉnh v_0 nào đó của đồ thị. Sau đó chọn u là một đỉnh tuỳ ý kề với v_0 và lặp lại quá trình đối với u. Ở bước tổng quát, giả sử ta đang xét đỉnh v. Nếu như trong số các đỉnh kề với v tìm được đỉnh w là chưa được xét thì ta sẽ xét đỉnh này (nó sẽ trở thành đã xét) và bắt đầu từ nó ta sẽ bắt đầu quá trình tìm kiếm còn nếu như không còn đỉnh nào kề với v là chưa xét thì ta nói rằng đỉnh này đã duyệt xong và quay trở lại tiếp tục tìm kiếm từ đỉnh mà trước đó ta đến được đỉnh v (nếu $v=v_0$, thì kết thúc tìm kiếm). Có thể nói nôm na là tìm kiếm theo chiều sâu bắt đầu từ đỉnh v được thực hiện trên cơ sở tìm kiếm theo chiều sâu từ tất cả các đỉnh chưa xét kề với v. Quá trình này có thể mô tả bởi thủ tục đệ qui sau đây:

Procedure DFS(v);

(*tim kiem theo chieu sau bat dau tu dinh v; cac bien Chuaxet, Ke la bien toan cuc*)

```
    Begin
        Tham_dinh(v);
        Chuaxet[v]:=false;
        For u∈Ke(v) do
        If Chuaxet[u] then DFS(u);
    End; (*dinh v da duyet xong*)
```

Khi đó, tìm kiếm theo chiều sâu trên đồ thị được thực hiện nhờ thuật toán sau:

```
    Begin
        (*Initialization*)
        for v∈V do Chuaxet[v]:=true;
        for v∈V do
```

if Chuaxet[v] then DFS(v);

End.

Rõ ràng lệnh gọi DFS(v) sẽ cho phép đến thăm tất cả các đỉnh thuộc cùng thành phần liên thông với đỉnh v, bởi vì sau khi thăm đỉnh là lệnh gọi đến thủ tục DFS đối với tất cả các đỉnh kề với nó. Mặt khác, do mỗi khi thăm đỉnh v xong, biến Chuaxet[v] được đặt lại giá trị false nên mỗi đỉnh sẽ được thăm đúng một lần. Thuật toán lần lượt sẽ tiến hành tìm kiếm từ các đỉnh chưa được thăm , vì vậy, nó sẽ xét qua tất cả các đỉnh của đồ thị (không nhất thiết phải là liên thông).

Để đánh giá độ phức tạp tính toán của thủ tục, trước hết nhận thấy rằng số phép toán cần thực hiện trong hai chu trình của thuật toán (hai vòng for ở chương trình chính) là cỡ n. Thủ tục DFS phải thực hiện không quá n lần. Tổng số phép toán cần phaỉ thực hiện trong các thủ tục này là O(n+m), do trong các thủ tục này ta phải xét qua tất cả các cạnh và các đỉnh của đồ thị. Vậy độ phức tạp tính toán của thuật toán là O(n+m).

2 Tìm kiếm theo chiều rộng (Breadth First Search-BFS)

Để ý rằng trong thuật toán tìm kiếm theo chiều sâu đỉnh được thăm càng muộn sẽ càng sớm trở thành đã duyệt xong. Điều đó là hệ quả tất yếu của việc các đỉnh được thăm sẽ được kết nạp vào trong ngăn xếp (STACK). Tìm kiếm theo chiều rộng trên đồ thị, nếu nói một cách ngắn gọn, được xây dựng trên cơ sở thay thế ngăn xếp (STACK) bởi hàng đợi (QUEUE). Với sự cải biên như vậy, đỉnh được thăm càng sớm sẽ càng sớm trở thành đã duyệt xong (tức là càng sớm rời khỏi hàng đợi). Một đỉnh sẽ trở thành đã duyệt xong ngay sau khi ta xét xong tất cả các đỉnh kề (chưa được thăm) với nó. Thủ tục có thể mô tả như sau:

Procedure BFS(v);

(*Tim kiem theo chieu rong bat dau tu dinh v, cac bien Chuaxet, Ke la bien cuc bo*)

begin

QUEUE:=$\emptyset$;

QUEUE$\Leftarrow$v; (*ket qua nap vao QUEUE*)

Chuaxet[v]:=false;

While QUEUE<>$\emptyset$ do

Begin

p$\Leftarrow$QUEUE:; (*lay p tu QUEUE:*)

Tham_dinh(p);

For u$\in$Ke(v) do

If Chuaxet[u] them

Begin

QUEUE $\Leftarrow$ u;

Chuaxet[u]:=false;

end;

end;

end;

Khi đó, tìm kiếm theo chiều rộng trên đồ thị được thực hiện nhờ thuật toán sau:

Begin

(*Initialization*)

for f $\in$ V do Chuaxet[v]:=true;

for v $\in$ V do

if Chuaxet[v] then BFS(v);

End.

Lập luận tương tự như trong thủ tục tìm kiếm theo chiều sâu, có thể chỉ ra được rằng lệnh gọi BFS(v) sẽ cho phép thăm đến tất cả các đỉnh thuộc cùng thành phần liên thông với đỉnh v, và mỗi đỉnh của đồ thị sẽ được thăm đúng một lần. Độ phức tạp tính toán của thuật toán là O(m+n).

3. Ứng dụng tìm đường đi và kiểm tra tính liên thông

Trong mục này ta xét ứng dụng các thuật toán tìm kiếm mô tả trong các mục trước vào việc giải bài toán cơ bản trên đồ thị: bài toán về tìm đường đi và bài toán về xác định tính liên thông của đồ thị

3.1 Bài toán tìm đường đi giữa hai đỉnh

Giả sử s và t là hai đỉnh nào đó của đồ thị. Hãy tìm đường đi từ s đến t.

Như trên đã phân tích, thủ tục DFS(s) **(BS(s))** sẽ cho thăm tất cả các đỉnh thuộc cùng một thành phần liên thông với s. vì vậy, sau khi thực hiện xong thủ tục, nếu Chuaxet[t]=true, thì điều đó có nghĩa là không có đường đi từ s đến t, còn nếu Chuaxet[t]=false thì t thuộc cùng thành phần liên thông với s, hay nói một cách khác: tồn tại đường đi từ s đến t. Trong trường hợp tồn tại đường đi, để ghi nhận đường đi, ta dùng thêm biểu thức Truoc[v] để ghi nhận đỉnh đi trước đỉnh v trong đường đi tìm kiếm tứ s đến v. Khi đó, đối với thủ tục DFS(v) cần sửa đổi câu lệnh if trong nó như sau:

If Chuaxet[u] then

Begin

Truoc[u]:=v;

DFS(u);

End;

Còn đối với thủ tục BFS(v) cần sửa đổi câu lện if trong nó như sau:

If Chuaxet [u] then

Begin

QUEUE $\Leftarrow$ u;

Chuaxet[u]:=false;

Truoc[u]:=p;

End;

Chú ý: Đường đi tìm được theo thuật toán tìm kiếm theo chiều rộng là đường đi ngắn nhất (theo số cạnh) từ s đến t. Điều này suy trực tiếp từ thứ tự thăm đỉnh theo thuật toán tìm kiếm theo chiều rộng.

3.2 Tìm các thành phần liên thông của đồ thị

Hãy cho biết đồ thị gồm bao nhiêu thành phần liên thông và từng thành phần liên thông của nó là gồm những đỉnh nào.

Do thủ tục DFS(v) (hoặc BFS(v)) cho phép thăm tất cả các đỉnh thuộc cùng một thành phần liên thông với s, nên số thành phần liên thông của đồ thị bằng số lần gọi đến thủ tục này. Vấn đề còn lại là cách ghi nhận các đỉnh trong từng thành phần liên thông. Ta dùng thêm biến Index[v] để ghi nhận chỉ số của thành phần liên thông chứa đỉnh v, và dùng thêm biến Inconnect để đếm số thành phần liên thông (biến này cần khởi tạo giá trị 0). Thủ tục Tham_dinh(v) trong các thủ tục DFS(v) và BFS(v) có nhiệm vụ gán: Index[v]:=connect, còn câu lệnh if trong các chương trình chính gọi đến các thủ tục này cần được sửa lại như sau:

Inconnect:=0;

If Chuaxet[v] then

Begin

Inconnect:=Inconnect+1;

DFS(v); (*BFS(v)*)

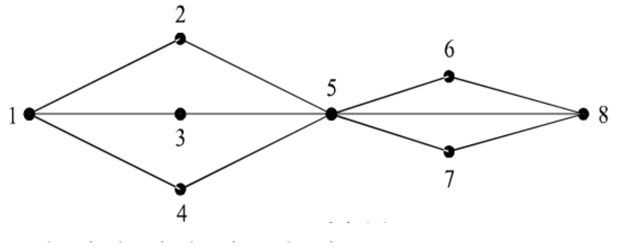
End;

Kết thúc vòng lặp thứ hai trong chương trình chính, Inconnect cho số thành phần liên thông của đồ thị, còn biến mảng Index[v], v thuộc V cho phép liệt kê các đỉnh thuôc cùng một thành phần liên thông.

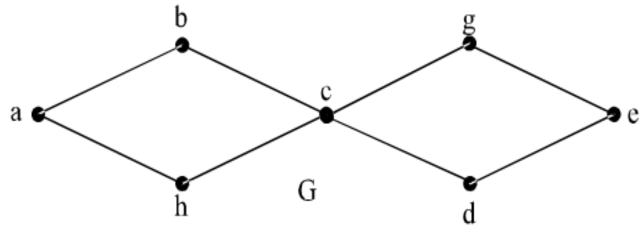
❖ **Bài tập củng cố:**

Duyệt đồ thị sau bằng thuật toán tìm kiếm theo chiều rộng và chiều sâu

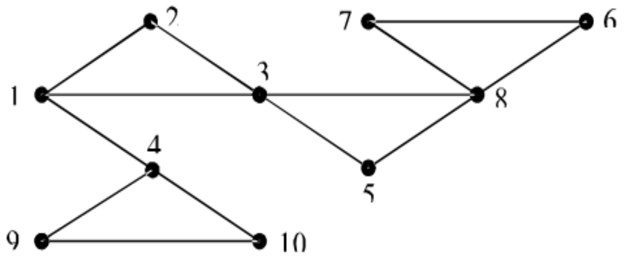
a.



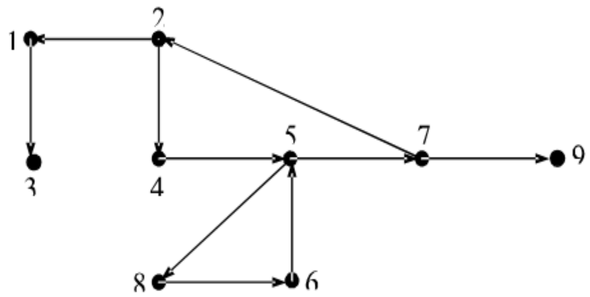
b.



c.



d.

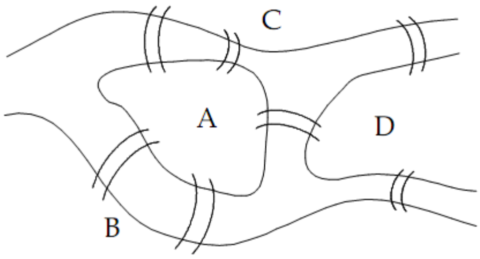


BÀI 2
ĐỒ THỊ EULER VÀ ĐỒ THỊ HAMILTON

❖ **Mục tiêu học tập:** Sau khi học xong bài này, người học có thể:
- Xác định đồ thị Euler và đồ thị Hamilton

1. Đồ thị Euler

Bài toán 7 cây cầu ở Königsberg: Thành phố Königsberg thuộc Phổ (bây giờ gọi là Kaliningrad thuộc Cộng hòa Liên bang Nga) được chia thành bốn vùng bằng các nhánh sông Pregel. Các vùng này gồm 2 vùng bên bờ sông, đảo Kneiphof và một miền nằm giữa 2 nhánh của sông Pregel. Vào thế kỷ thứ XVIII, người ta đã xây 7 cây cầu nối các vùng lại với nhau như sơ đồ sau:



Hình 1. Mô hình 7 cây cầu ở Königsberg

Vào chủ nhật, người dân ở đây thường đi bộ dọc theo các vùng trong thành phố. Họ tự hỏi "***Có thể xuất phát tại một điểm nào đó trong thành phố, đi qua tất cả 7 cây cầu, mỗi cây một lần, rồi trở về điểm xuất phát được không?***"

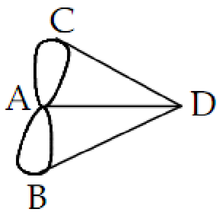
Nhà toán học Thụy Sĩ Leonard Euler đã nghiên cứu giải bài toán này. Lời giải của ông được công bố năm 1736. Bài toán này có thể được coi là một trong những ứng dụng đầu tiên của lý thuyết đồ thị.

Ta có thể xây dựng đồ thị G = (V, E) mô tả bài toán như sau:

- ***Đỉnh***: Lấy các điểm trên mặt phẳng hay trong không gian tương ứng với các vùng đất trong sơ đồ. **Đối tượng** của bài toán ở đây là một vùng đất trong sơ đồ. Vậy, mỗi đỉnh $v \in V$ biểu diễn cho một vùng đất. Đồ thị G sẽ có 4 đỉnh A, B, C, D tương ứng với 4 vùng đất.
- ***Cạnh***: Trong đồ thị G các đỉnh v_i và v_j được nối với nhau bằng một cạnh e đại diện cho một chiếc cầu nối giữa hai vùng đất. Đồ thị G sẽ có 7 cạnh tương ứng với 7 chiếc cầu nối giữa các vùng đất trong sơ đồ.

Euler đã nghiên cứu bài toán này, mô hình nó bằng một đa đồ thị, bốn vùng được biểu diễn bằng 4 đỉnh, các cầu là các cạnh như đồ thị sau:

Bài toán tìm đường đi qua tất cả các cầu mỗi cầu không quá một lần có thể được phát biểu lại bằng mô hình này như sau: ***"Tồn tại hay không một chu trình đơn trong đa đồ thị G= (V, E) có chứa tất cả các cạnh?"***



Hình 2. Mô hình hóa bài toán 7 cây cầu

1.1 Định nghĩa

1.1.1 Chu trình Euler

Cho đồ thị G=(V,E). Chu trình đơn trong đồ thị G đi qua mỗi cạnh của nó đúng một lần được gọi là chu trình Euler. Đồ thị có chứa một chu trình Euler được gọi là đồ thị Euler.

1.1.2 Đường đi Euler

Cho đồ thị G=(V,E). Đường đi đơn trong đồ thị G đi qua mỗi cạnh của nó đúng một lần được gọi là đường đi Euler. Đồ thị có chứa một đường đi Euler được gọi là đồ thị nửa Euler.

1.2 Định lý về chu trình và đường đi Euler trong đồ thị vô hướng

Khi giải bài toán cầu Königsberg, Euler đã phát hiện ra các tiêu chuẩn để khẳng định một đa đồ thị có chu trình và đường đi Euler hay không?

1.2.1 Định lý về chu trình Euler

Một đa đồ thị liên thông G =(V, E) có chu trình Euler khi và chỉ khi **mỗi đỉnh của nó đều có bậc chẵn.**

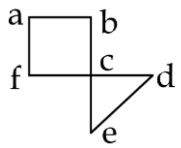
Chứng minh

(⇒) Ta sẽ chứng minh nếu đồ thị G có chu trình Euler thì mọi đỉnh của G đều có bậc chẵn.

Thật vậy, trước tiên giả sử G có chu trình Euler bắt đầu từ đỉnh a và tiếp tục bằng cạnh liên thuộc với a, chẳng hạn ab, khi đó cạnh ab góp một vào deg(a). Mỗi lần khi chu trình đi qua một đỉnh, nó tăng thêm 2 đơn vị cho bậc của đỉnh đó. Vì chu trình đi vào một đỉnh bằng một cạnh liên thuộc và rời khỏi đỉnh này bằng một cạnh liên thuộc khác. Cuối cùng chu trình kết thúc ở đỉnh mà nó xuất phát, do đó nó tăng thêm một đơn vị vào deg(a). Nghĩa là deg(a) phải là một số chẵn. Đỉnh khác a cũng có bậc chẵn vì chu trình góp 2 đơn vị vào bậc của nó mỗi lần đi qua đỉnh này. Vậy, mỗi đỉnh của G đều có bậc chẵn.

(⇐) Giả sử mọi đỉnh của đa đồ thị liên thông G đều có bậc chẵn. Ta sẽ chứng minh tồn tại một chu trình Euler trong G.

Thật vậy, ta sẽ xây dựng một chu trình đơn bắt đầu từ đỉnh a tùy ý của G. Gọi x_o = a; Trước tiên, ta chọn tùy ý cạnh x_ox_1, x_1x_2, ..., $x_{n-1}x_n$ càng dài càng tốt. Ví dụ, trong đồ thị G sau:



Ta bắt đầu tại a và chọn các cạnh liên tiếp ab, bc, cf, fa. Đường đi mà ta chọn sẽ kết thúc vì đồ thị có hữu hạn đỉnh. Đường đi này bắt đầu tại a với cạnh có dạng ax và kết thúc tại a với cạnh có dạng ya.

Điều này luôn xảy ra vì mỗi lần đường đi qua một đỉnh bậc chẵn, nó chỉ dùng một cạnh để vào đỉnh này và còn ít nhất một đỉnh để ra khỏi đỉnh này. Đường đi vừa nói trên có thể đi qua tất cả các cạnh hoặc có thể không. Nếu tất cả các cạnh được sử dụng thì ta nhận được chu trình Euler. Nếu không, ta gọi H là đồ thị con nhận được từ G bằng cách xóa các cạnh đã dùng và các đỉnh không liên thuộc với các cạnh còn lại. Chẳng hạn với đồ thị trên, khi xóa đi chu trình a, b, c, f, a khỏi đồ thị trên, ta nhận được đồ thị con H.

Vì G là liên thông ⇒ H có ít nhất có một đỉnh chung với chu trình vừa bị xóa. Gọi w là đỉnh đó (trong ví dụ trên là đỉnh c). Mỗi đỉnh của H có bậc chẵn vì tất cả các đỉnh của G có bậc chẵn và với mỗi đỉnh ta đã xóa đi từng cặp liên thuộc để tạo ra H. Lưu ý rằng H có thể không liên thông. Bắt đầu từ đỉnh w ta xây dựng một đường đi đơn bằng cách chọn càng nhiều càng tốt như ta đã làm trong G. Đường này phải kết thúc tại w. Ví dụ trong đồ thị H nêu trên ta có chu trình con: c, d, e, c. Sau đó, ta tạo một chu trình trong G bằng cách ghép chu trình trong H và chu trình ban đầu trong G (điều này thực hiện được vì 2 chu trình có chung đỉnh w). Tiếp tục quá trình này cho đến khi tất cả các đỉnh được sử dụng. Quá trình này phải kết thúc vì đồ thị có hữu hạn đỉnh. Do đó, ta đã xây dựng được một chu trình Euler.

Bây giờ, trở lại bài toán 7 cây cầu ở Königsberg: có thể xuất phát từ một địa điểm nào đó trong thành phố, đi qua tất cả các cầu (mỗi cầu đi qua đúng một lần) và trở về điểm xuất phát? Ta đã thấy đồ thị biểu diễn các cầu ở Königsberg có 4 đỉnh bậc lẻ. Do đó, theo định lý trên sẽ không có chu trình Euler trong đồ thị này. Điều này cũng có nghĩa là bài toán 7 cây cầu ở Königsberg không có lời giải. Hay nói cách khác, không có chu trình nào thỏa yêu cầu đặt ra.

1.2.2 Định lý về đường đi Euler

Đa đồ thị liên thông G =(V, E) có đường đi Euler nhưng không có chu trình Euler khi và chỉ khi nó có **đúng hai đỉnh bậc lẻ**.

Chứng minh:

(⇒) Giả sử đa đồ thị G có đường đi Euler nhưng không có chu trình Euler. Ta sẽ chứng minh G có đúng 2 đỉnh bậc lẻ.

Thật vậy, giả sử G có đường đi Euler từ a đến b, nhưng không có chu trình Euler. Cạnh đầu tiên của đường đi góp một đơn vị vào deg(a). Sau đó mỗi lần đường đi qua a lại góp thêm 2 đơn vị vào deg(a).

Chắc chắn đường đi không thể kết thúc tại a, cho nên deg(a) là số lẻ. Cạnh cuối cùng của đường đi góp một đơn vị vào deg(a) và mỗi lần đi qua b, nó cũng góp 2 đơn vị vào deg(b). Do đó, deg(b) là số lẻ. Các đỉnh trung gian đều có bậc chẵn vì mỗi lần đường đi tới rồi lại đi nên tăng hai đơn vị cho bậc của đỉnh đó. Vậy, đồ thị đã cho có đúng 2 đỉnh bậc lẻ.

(⇐) Giả sử đa đồ thị liên thông G có đúng 2 đỉnh bậc lẻ. Ta sẽ chứng minh G có đường đi Euler.

Thật vậy, giả sử G có đúng 2 đỉnh bậc lẻ là a và b. Khi đó trong đồ thị mới G' = G ∪ ab, tất cả các đỉnh đều có bậc chẵn. Do đó, theo định lý Euler, tồn tại một chu trình Euler trong G'. Trong chu trình này bỏ cạnh ab, ta được đường đi Euler trong G.

Như vậy, trong một đa đồ thị liên thông có 2 đỉnh bậc lẻ thì đường đi Euler trong đồ thị đó sẽ nhận 2 đỉnh bậc lẻ làm các điểm đầu mút.

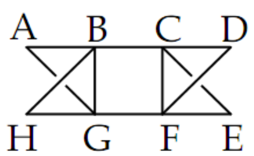
1.2.3 Thuật toán Fleury tìm chu trình Euler

Để tìm một chu trình Euler trong một đa đồ thị có tất cả các đỉnh đều bậc chẵn, ta có thể sử dụng thuật toán Fleury như sau:

Xuất phát từ một đỉnh bất kỳ của đồ thị G và tuân theo hai qui tắc sau:

- *Qui tắc 1: Mỗi khi đi qua một cạnh nào thì xóa cạnh đó đi, sau đó xóa đỉnh cô lập (nếu có).*
- *Qui tắc 2: Không bao giờ đi qua một cầu, trừ khi không còn cách đi nào khác để di chuyển.*

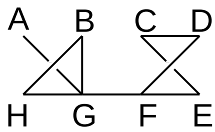
1.2.4 Các ví dụ



Hình 3. Đồ thị có chứa chu trình Euler

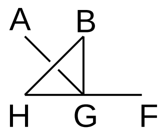
Ví dụ 1: Tìm một chu trình Euler trong đồ thị ở hình 3 bằng *thuật toán Fleury*

Xuất phát từ đỉnh A, giả sử ta chọn cạnh AB, BC, CF. Sau đó xóa 3 cạnh này, ta được đồ thị:



Đến đây, ta không thể chọn FG vì GF là một cầu, cho nên ta chọn FD, DC, CE, EF.

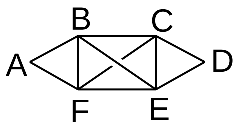
Đến đây, ta được đồ thị sau:



Ta không còn cách chọn nào khác, nên phải chọn FG, GH, HB, BG, GA.

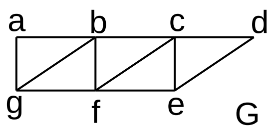
Như vậy, ta có chu trình Euler sau: A, B, C, F, D, C, E, F, G, H, B, G, A.

Ví dụ 2: Tìm một chu trình Euler của đồ thị sau:



Dễ thấy một chu trình Euler: A, B, C, D, E, C, F, B, E, F, A.

Ví dụ 3: Tìm đường đi Euler của đồ thị sau nếu có:



1.3 Chu trình và đường đi Euler đối với đồ thị có hướng

1.3.1 Định lý về chu trình Euler

Đồ thị có hướng G=(V, E) có chứa một chu trình Euler khi và chỉ khi G là liên thông yếu, đồng thời bậc vào và bậc ra của mỗi đỉnh là bằng nhau.

Chứng minh:

Tương tự định lý Euler đối với đồ thị vô hướng (*Xem như bài tập - Sinh viên tự chứng minh*).

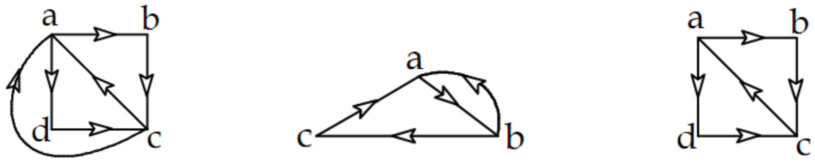
1.3.2 Định lý về đường đi Euler

Cho G =(V,E) là một đa đồ thị có hướng. G có đường đi Euler nhưng không có chu trình Euler ⇔ G là liên thông yếu, đồng thời bậc vào và bậc ra của các đỉnh là bằng nhau, trừ 2 đỉnh, một đỉnh có bậc vào lớn hơn bậc ra một đơn vị, còn đỉnh kia có bậc vào nhỏ hơn bậc ra một đơn vị.

Chứng minh:

Tương tự định lý Euler đối với đồ thị vô hướng

Ví dụ: Tìm chu trình và đường đi Euler nếu có của các đồ thị sau:

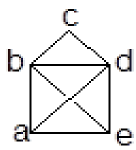


2. Đồ thị Hamilton

2.1 Chu trình Hamilton

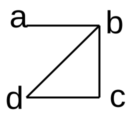
Một chu trình sơ cấp đi qua tất cả các đỉnh của đồ thị G=(V,E) (đi qua mỗi đỉnh đúng một lần) được gọi là chu trình Hamilton. Đồ thị G=(V,E) có chứa chu trình Hamilton được gọi là đồ thị Hamilton.

Ví dụ 1: Đồ thị sau có chu trình Hamilton là: a, b, c, d, e, a.



Hình 4. Đồ thị chứa chu trình Hamilton

Ví dụ 2: Đồ thị sau đây không có chu trình Hamilton vì mọi chu trình chứa mọi đỉnh của đồ thị đều phải đi qua cạnh ab 2 lần.



2.1.1 Điều kiện đủ để tồn tại chu trình Hamilton

Định lý Dirac:

Đơn đồ thị vô hướng liên thông G=(V,E) có bậc ở mỗi đỉnh không nhỏ hơn nửa số đỉnh của đồ thị $\deg(v_i) \geq \frac{|V|}{2}, \forall v_i \in V$ thì đồ thị luôn tồn tại chu trình Hamilton (giả thiết rằng $|V| > 2$)

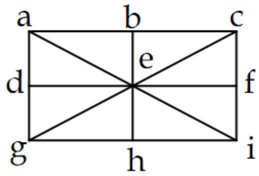
2.1.2 Phương pháp tìm chu trình Hamilton

Cho một đồ thị G =(V,E). Để tìm một chu trình Hamilton trong đồ thị G, ta thực hiện theo 4 qui tắc sau:

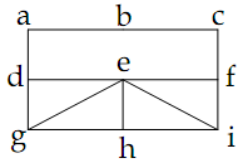
- *Qui tắc 1: Nếu tồn tại một đỉnh v của G có* $\deg(v) \leq 1$ *thì đồ thị G không có chu trình Hamilton.*
- *Qui tắc 2: Nếu đỉnh v có bậc là 2 thì cả 2 cạnh tới v đều phải thuộc chu trình Hamilton.*
- *Qui tắc 3: Chu trình Hamilton không chứa bất kỳ chu trình con thực sự nào.*
- *Qui tắc 4: Trong quá trình xây dựng chu trình Hamilton, sau khi đã lấy 2 cạnh tới một đỉnh v đặt vào chu trình Hamilton rồi thì không thể lấy thêm cạnh nào*

tới v nữa, do đó có thể xóa mọi cạnh còn lại tới v.

Ví dụ 1: Tìm một chu trình Hamilton của đồ thị sau:

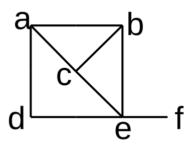


Xuất phát từ đỉnh a. Ta có deg(a) = 3, cho nên ta chỉ giữ lại 2 cạnh liên thuộc với a: ta chọn ab và ad, do đó ta bỏ cạnh ae (theo quy tắc 4). Tương tự tại b, ta bỏ cạnh be tại c ta bỏ cạnh ce (theo quy tắc 4). Khi đó ta có đồ thị:

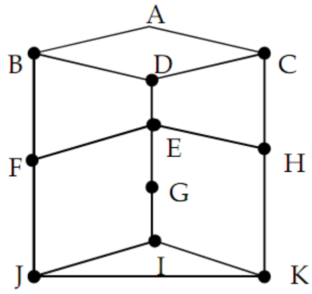


Tại đỉnh f, ta bỏ cạnh fe. Tại đỉnh i ta bỏ cạnh ie, tại đỉnh h ta bỏ cạnh hg (theo quy tắc 4) tại đỉnh e, ta bắt buộc đi theo eg, gd (theo quy tắc 2). Tại đỉnh a ta phải chọn cạnh da (theo quy tắc 2). Vậy, ta có chu trình Hamilton: **a, b, c, f, i, h, e, g, d, a**.

Ví dụ 2: Đồ thị sau đây không có chu trình Hamilton vì: deg(f) = 1.



Ví dụ 3: Chứng minh rằng đồ thị sau không có chu trình Hamilton:

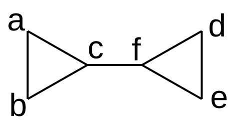


Hình 5. Ví dụ về đồ thị không có chu trình Hamilton

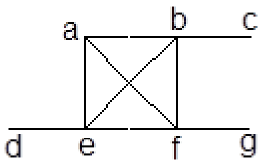
2.2 Đường đi Hamilton

Đường đi sơ cấp đi qua tất cả các đỉnh của đồ thị G=(V,E) (đi qua mỗi đỉnh đúng một lần) được gọi là đường đi Hamilton. Đồ thị có đường đi Hamilton còn gọi là đồ thị nửa Hamilton

Ví dụ 1: Đồ thị sau có đường đi Hamilton là: a, b, c, f, d, e



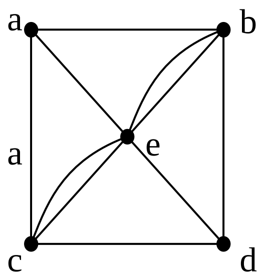
Ví dụ 2: Tìm đường đi Hamilton nếu có của đồ thị:



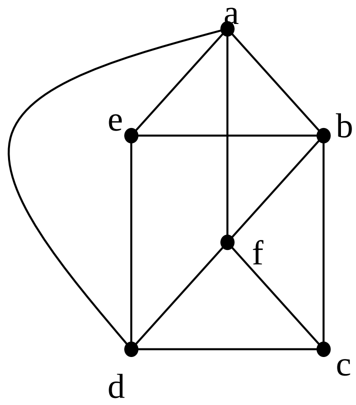
❖ **Bài tập củng cố:**

Bài 1. Các đồ thị sau có chu trình Euler, đường đi Euler hay không? Nếu có hãy xây dựng chu trình, đường đi đó.

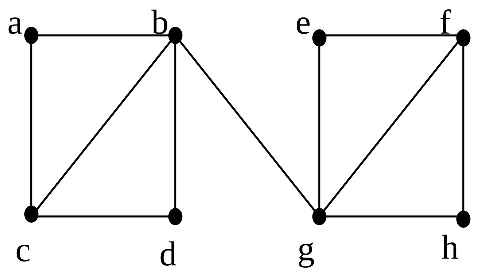
a.



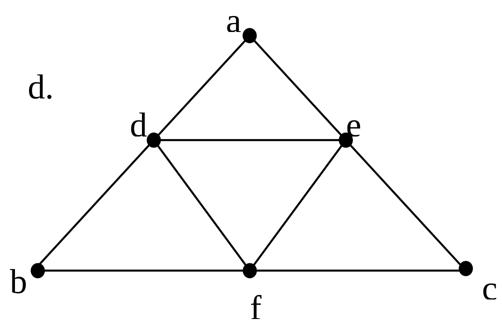
b.



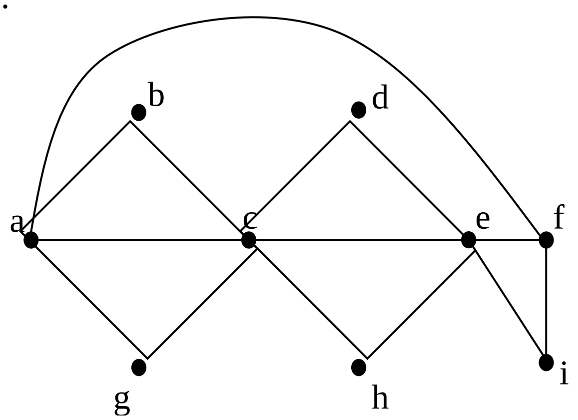
c.



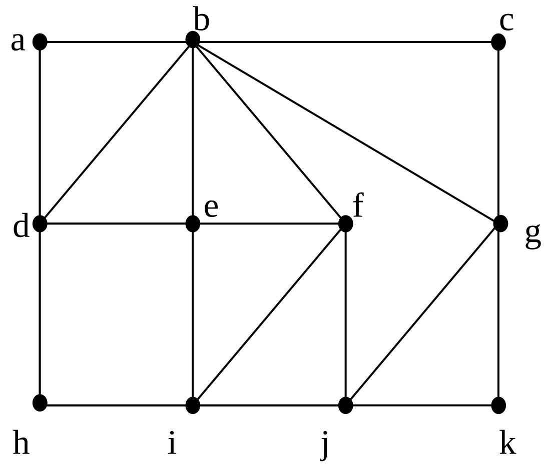
d.



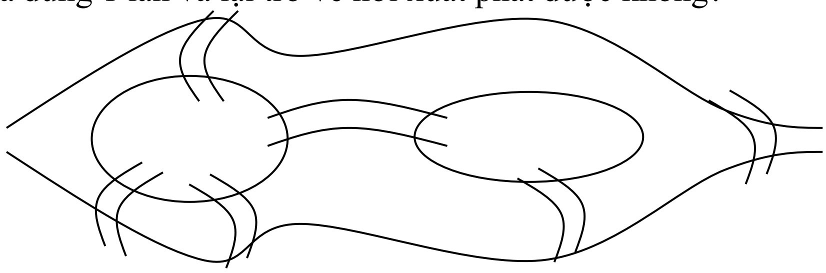
e.



f.

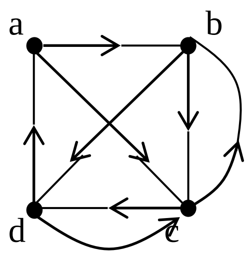


Bài 2. Một người nào đó có thể đi qua những chiếc cầu như trên hình vẽ sau, mỗi chiếc cầu đi qua đúng 1 lần và lại trở về nơi xuất phát được không?

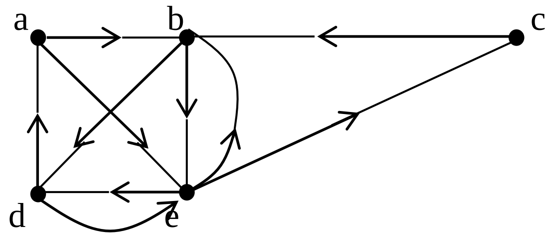


Bài 3. Xem xét các đồ thị có hướng sau, có chu trình hay đường đi Euler hay không? Nếu có, hãy xây dựng chu trình và đường đi đó.

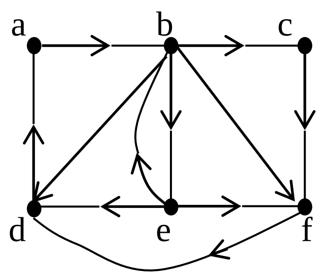
a.



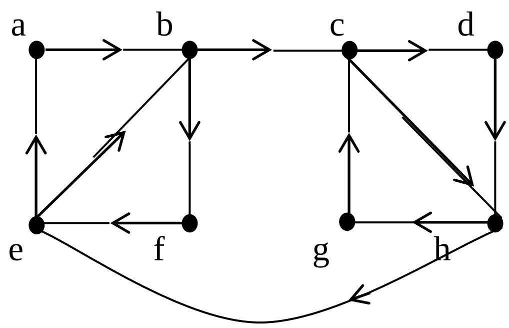
b.



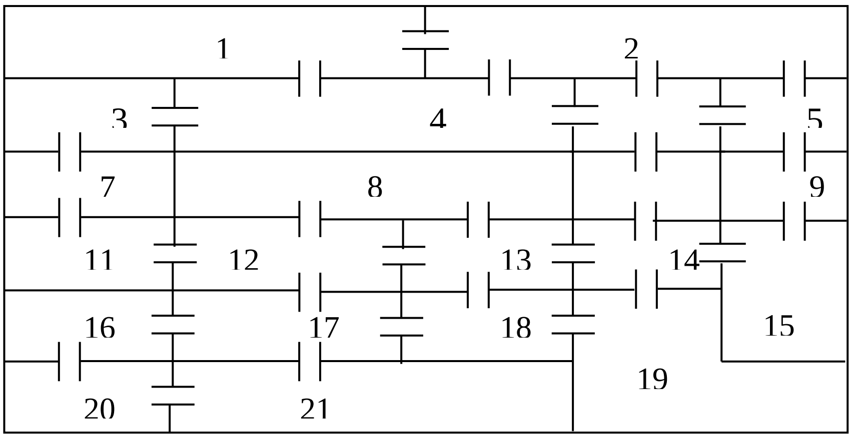
c.



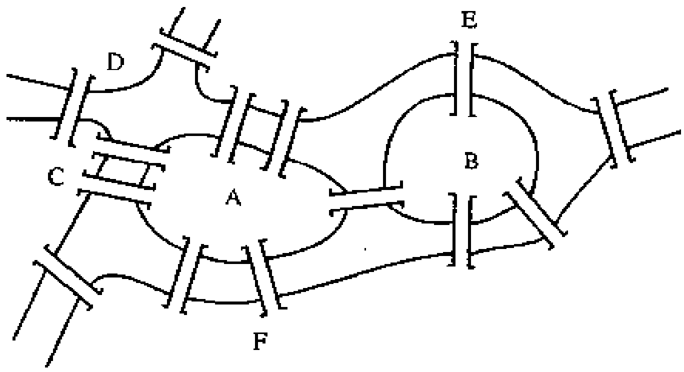
d.



Bài 4. Một ông vua đã xây dựng một lâu đài để cất báu vật. Người ta tìm thấy sơ đồ của lâu đài như sau với lời căn dặn: muốn tìm báu vật, chỉ cần từ một trong các căn phòng bên ngoài cùng (số 1, 2, 6, 10...) đi qua tất cả các cửa phòng, mỗi cửa chỉ một lần. Báu vật được giấu sau cánh cửa cuối cùng. Hãy tìm nơi giấu báu vật.



Bài 5. Cho hình vẽ sau



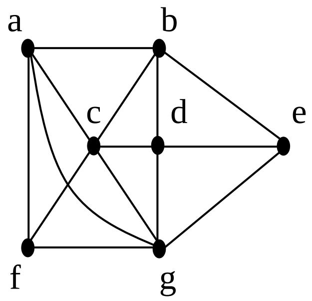
a. Vẽ đồ thị G tương ứng

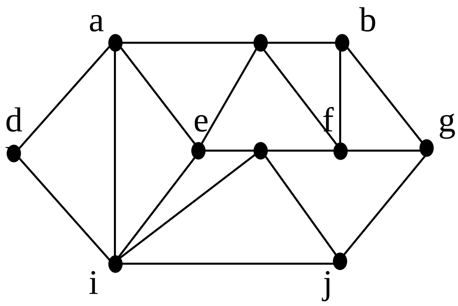
b. G có chu trình Euler hoặc đường đi Euler không? Tại sao?

Bài 6. Tìm chu trình Euler hoặc đường đi Euler nếu có của đồ thị với ma trận kề sau

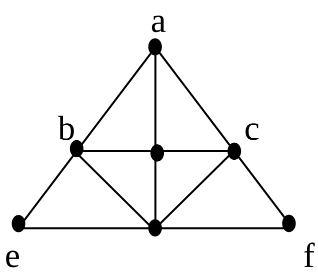
$$a.\ \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{bmatrix} \qquad b.\ \begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 2 & 0 & 1 \\ 1 & 0 & 0 & 2 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Bài 7. Tìm các chu trình Hamilton và đường đi Hamilton (nếu có) của các đồ thị sau:

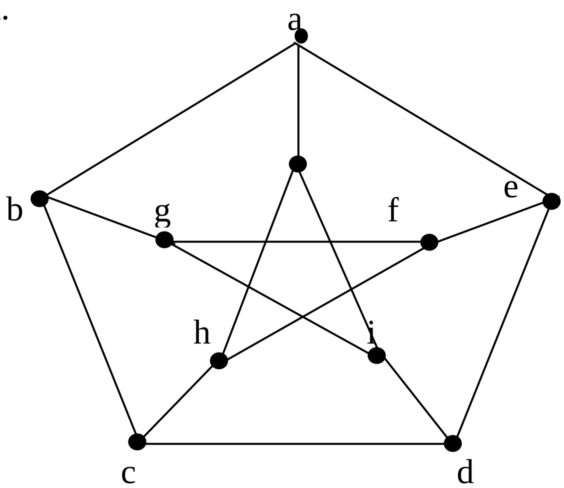




c.



d.



Đồ thị Petersen

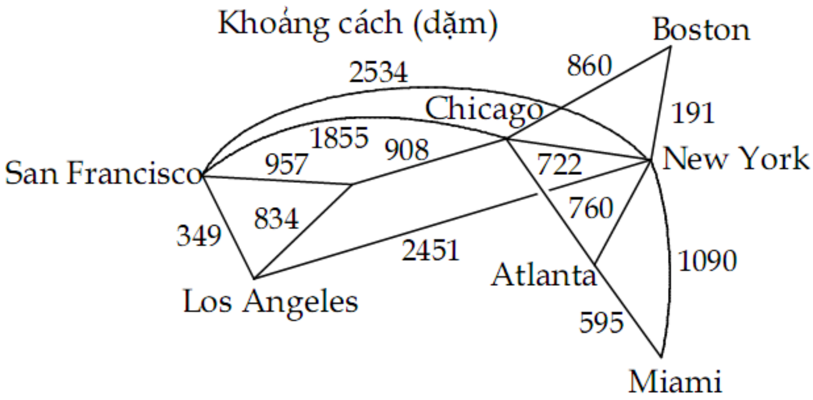
BÀI 3
BÀI TOÁN ĐƯỜNG ĐI NGẮN NHẤT

❖ **Mục tiêu học tập:** Sau khi học xong bài này, người học có thể:
- Tìm đường đi ngắn nhất trên đồ thị có trọng số

1. Mở đầu

Trong thực tế, nhiều bài toán có thể mô hình bằng đồ thị có trọng số. Đó là đồ thị mà mỗi cạnh của nó được gán một con số (nguyên hoặc thực) gọi là trọng số ứng với cạnh đó. Ví dụ ta cần mô hình một hệ thống đường hàng không. Mỗi thành phố được biểu diễn bằng một đỉnh, mỗi chuyến bay là một cạnh nối 2 đỉnh tương ứng. Nếu trong bài toán đang xét ta cần tính đến khoảng cách giữa các thành phố thì ta cần gán cho mỗi cạnh của đồ thị cơ sở trên khoảng cách giữa các thành phố tương ứng. Nếu ta quan tâm đến thời gian của mỗi chuyến bay thì ta sẽ gán các thời lượng này cho mỗi cạnh của đồ thị cơ sở

Đồ thị biểu diễn khoảng cách giữa một số thành phố của nước Mỹ.



Hình 1. Khoảng cách giữa các thành phố

Bài toán đặt ra là *tìm đường đi ngắn nhất từ thành phố này đến thành phố khác*. Hay nói theo ngôn ngữ của lý thuyết đồ thị: *ta cần tìm đường đi có tổng trọng số (ngắn) nhỏ nhất từ đỉnh này đến một đỉnh khác của đồ thị.*

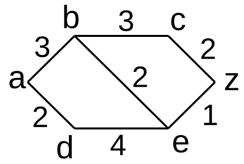
2. Thuật toán tìm đường đi ngắn nhất

Thuật toán được áp dụng cho đồ thị không chứa chu trình và khuyên có độ dài âm.

2.1 Thuật toán Dijkstra tìm đường đi ngắn nhất từ một đỉnh

Có một số thuật toán tìm đường đi ngắn nhất giữa 2 đỉnh trên một đồ thị có trọng số. Ở đây ta sẽ sử dụng thuật toán Dijkstra, do nhà toán học người Hà Lan: E.Dijkstra đề xuất năm 1959. Chúng ta sẽ áp dụng thuật toán Dijkstra đối với đồ thị vô hướng. Đối với đồ thị có hướng, ta chỉ cần thay đổi một chút. Trước khi giới thiệu thuật toán, ta xét ví dụ sau:

Tính độ dài của đường đi ngắn nhất giữa 2 đỉnh a và z trong đồ thị có trọng số sau:



Đối với đồ thị này, ta dễ dàng tìm được đường đi ngắn nhất từ a đến z bằng cách thử trực tiếp. Tuy nhiên, ta sẽ phát triển một số ý tưởng giúp ta hiểu thuật toán Dijkstra dễ dàng hơn. Ta sẽ tìm độ dài của đường đi ngắn nhất từ a đến các đỉnh kế tiếp cho đến khi đạt tới đỉnh z. Xuất phát từ đỉnh a, ta thấy chỉ có 2 đỉnh b và d liên thuộc với a. Nên chỉ có hai đường đi xuất phát từ a đến b và d là ab và ad với độ dài tương ứng là 3 và 2. Do đó, d là đỉnh gần a nhất.

Bây giờ, ta tìm đỉnh tiếp theo gần a nhất trong tất cả các đường đi qua a và d. Đường đi ngắn nhất từ a tới b là ab với độ dài 3. Đường đi ngắn nhất từ a đến e là a, b, e với độ dài 5. Đường đi ngắn nhất từ a đến c là a, b, c với độ dài 6. Khi đó ta có 2 đường đi từ a đến z qua c và e là a, b, c, z với độ dài 8; a, b, e, z với độ dài 6. Vậy, đường đi ngắn nhất từ a đến z là: a, b, e, z với độ dài 6.

Ví dụ trên đã minh họa những nguyên tắc chung dùng trong thuật toán Dijkstra. Đường đi ngắn nhất từ đỉnh a đến z có thể tìm được bằng cách thử trực tiếp. Nhưng phương pháp này không áp dụng được đối với đồ thị có nhiều cạnh.

Bây giờ, ta nghiên cứu bài toán tìm độ dài của đường đi ngắn nhất giữa a và z trong đơn đồ thị liên thông, vô hướng và có trọng số.

Thuật toán Dijkstra được thực hiện bằng cách tìm độ dài của đường đi ngắn nhất từ a đến đỉnh đầu tiên, từ a đến đỉnh thứ hai,... cho đến khi tìm được độ dài ngắn nhất từ a đến z.

Thuật toán này dựa trên một dãy các bước lặp. Một tập đặc biệt các đỉnh được xây dựng bằng cách cộng thêm một đỉnh trong mỗi bước lặp. Thủ tục gán nhãn được thực hiện trong mỗi lần lặp đó.

Trong thủ tục gán nhãn này, đỉnh w được gán nhãn bằng độ dài đường đi ngắn nhất từ a đến w và chỉ đi qua các đỉnh thuộc tập đặc biệt. Một đỉnh được thêm vào tập này là đỉnh có nhãn nhỏ nhất so với các đỉnh chưa có trong tập đó.

Cụ thể của thuật toán Dijkstra như sau:

- Gán cho đỉnh a nhãn bằng 0; còn các đỉnh khác bằng ∞. Ta ký hiệu:
  $L_0(a) = 0$; $L_0(v) = \infty$; $\forall\ v \neq a$ (đây là bước lặp thứ 0).

- Gọi S_k là tập đặc biệt các đỉnh sau bước lặp thứ k của thủ tục gán nhãn. Chúng ta bắt đầu bằng S_o = {a}. Tập S_k được tạo thành từ S_{k-1} bằng cách thêm vào đỉnh u $\notin S_{k-1}$ mà có nhãn nhỏ nhất.
- Sau khi đỉnh u được ghép vào S_k, ta sửa đổi nhãn của các đỉnh không thuộc S_k sao cho $L_k(v)$ (nhãn của đỉnh v tại bước k) là độ dài của đường đi ngắn nhất từ a đến v mà đường đi này chỉ chứa các đỉnh thuộc S_k (tức là các đỉnh đã thuộc tập đặc biệt cùng với đỉnh u).
- Giả sử v là một đỉnh không thuộc S_k. Để sửa nhãn của v, ta chọn $L_k(v)$ là độ dài của đường đi ngắn nhất từ a đến v và chỉ chứa các đỉnh thuộc S_k. Để ý rằng đường đi ngắn nhất từ a đến v chỉ chứa các phần tử của S_k hoặc là đường đi ngắn nhất từ a đến v chỉ chứa các phần tử của S_{k-1} hoặc là đường đi ngắn nhất từ a đến u trong bước (k−1) cộng với độ dài cạnh uv. Tức là:

  $$\mathbf{L_k(a,v) = \min\{L_{k-1}(a,v);\ L_{k-1}(a,u) + w(uv)\}}$$

  với w(uv) là trọng số của cạnh uv.
- Thủ tục này được lặp bằng cách liên tiếp thêm các đỉnh vào tập đặc biệt các đỉnh cho tới khi đạt tới đỉnh z. Khi thêm z vào tập đặc biệt các đỉnh thì nhãn của nó bằng độ dài của đường đi ngắn nhất từ a đến z.

Thuật toán Dijkstra có thể được trình bày dưới dạng giải mã (pseudo - code) như sau:

THUẬT TOÁN DIJKSTRA

Procedure Dijkstra (G: đơn đồ thị liên thông có trọng số dương);

{G có các đỉnh $a=v_o, v_1, v_2, ..., v_n=z$ và trọng số $w(v_iv_j)$ với $w(v_iv_j)=\infty$ nếu v_iv_j không là một cạnh của G}.

for i = 1 to n

$L(v_i)$: = ∞

L(a): = 0

S: = {a}

{Ban đầu các nhãn được khởi tạo sao cho nhãn của a bằng không, các đỉnh khác bằng ∞; S = {a}}

While z $\notin$ S

begin

u: đỉnh không thuộc S có L(u) nhỏ nhất

S: = S ∪ {u}

for tất cả các đỉnh v không thuộc S

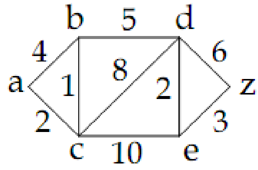
if L(u) + w(uv) < L(v) then

L(v): = L(u) + w(uv)

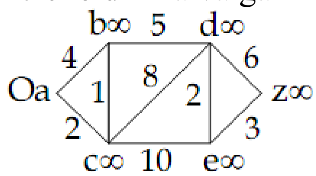
{thêm vào S đỉnh có nhãn nhỏ nhất, và sửa đổi nhãn của các đỉnh không thuộc S}

end {L(z) = Độ dài đường đi ngắn nhất từ a tới z}.

Ví dụ: Dùng thuật toán Dijkstra, tìm đường đi ngắn nhất từ a đến z trong đồ thị sau:



- Ta có: V={a,b,c,d,e,z}, S = {a}.
- Tại bước lặp thứ 1: ta gán 0 cho đỉnh a và gán ∞ cho các đỉnh còn lại. L(a)=0.

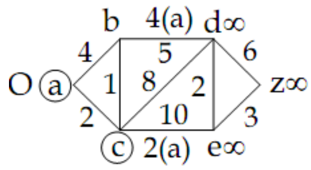


- Trong các đỉnh không thuộc S = {a} và kề với a có 2 đỉnh b và c. Ta có:

  L(b) = min {∞, L(a) + w(ab)} = min {∞, 0 + 4} = 4.

  L(c) = min {∞, L(a) + w(ac)} = min {∞, 0 + 2} = 2.

  Ta có: L(c) nhỏ nhất nên c ∈ S (thêm c vào S) . ⇒ S = {a,c}.



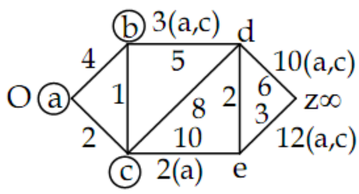
- Trong các đỉnh không thuộc S mà kề với c có 3 đỉnh là b, d, e.

  L(b) = min {4, L(c) + w(cb)} = min{4,2+1} = 3.

  L(e) = min {∞, L(c) + w(ce)} = min{∞,12} = 12.

  L(d) = min{∞,L(c) + w(cd)} = min{∞,2 + 8} = 10.

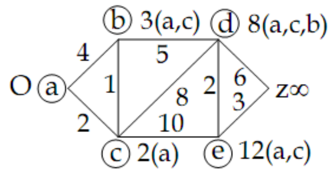
  Ta có: L(b) nhỏ nhất nên b ∈ S ⇒ S = {a,b,c}.



- Trong các đỉnh không thuộc S mà kề với b là d.

  L(d) = min{10,L(b) + w(bd)} = min{10,3 + 5} = 8

  ⇒d ∈ S: S = {a,b,c,d}

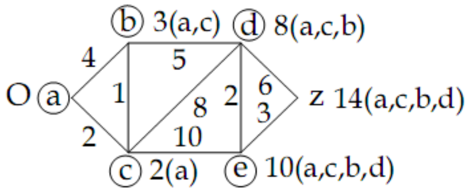


- Trong các đỉnh kề với d mà không thuộc S, có: e,z.

  L(e) = min{12, L(d) + w(de)} = min{12, 8+2} = 10

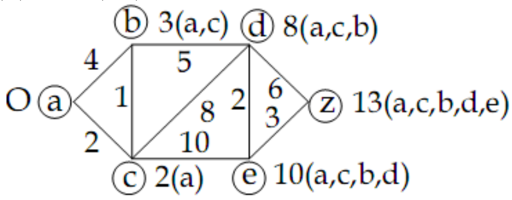
  L(z) = min{∞, d(d) + w(dz)} = min{∞, 8+6} = 14

  ⇒e ∈ S: S = {a,b,c,d,e}.



- Các đỉnh kề với e mà không thuộc S: z.

  L(z) = min{14, L(e) + w(ez)} = min{14, 10+3} = 13.



Vậy, đường đi ngắn nhất từ a đến z là: a, c, b, d, e với độ dài 13.

2.2 Thuật toán Floyd tìm đường đi ngắn nhất giữa tất cả các cặp đỉnh

Ta sẽ tính một ma trận khoảng cách là ma trận vuông cấp n. Nếu tất cả chiều dài các cạnh đều không âm hay $l(u) \geq 0$ ta có thể áp dụng n lần thuật toán Dijktra cho mỗi đỉnh i. Nếu đồ thị có chứa chiều dài âm $(l(u) < 0)$ ta có thể áp dụng thuật toán Floyd như sau.

Ta định nghĩa ***ma trận trọng lượng A*** và ***ma trận đỉnh trước P*** như sau:

$$A_0[i,j] = \begin{cases} 0 & \text{nếu } i = j \\ l(i,j) & \text{nếu } (i,j) \in U \\ \infty & \text{ngược lại} \end{cases}$$

với U là tập cạnh của đồ thị, l(i,j) là trọng số của cạnh (i,j)

Ma trận P: được gán giá trị ban đầu như sau:

$P_0[i,j] = j$ nếu có cạnh nối đỉnh i với đỉnh j và $P_0[i,j] = 0$ nếu đỉnh i và đỉnh j không có cạnh nối.

Ta xác định A_k và P_k dựa vào A_{k-1} và P_{k-1} như sau:

Với i=1 đến n

  Với j=1 đến n

    Nếu $A_{k-1}[i,j] > A_{k-1}[i,k]+A_{k-1}[k,j]$ thì đặt:

$A_k[i,j] = A_{k-1}[i,k]+A_{k-1}[k,j]$ và $P_k[i,j] = P_{k-1}[i,k]$

Nếu không, $A_k[i,j] = A_{k-1}[i,j]$ và $P_k[i,j] = P_{k-1}[i,j]$

Dễ thấy rằng khi thuật toán kết thúc, ngoài ma trận khoảng cách ngắn nhất $A^* = A_n$ cho ta biết chiều dài đường đi ngắn nhất nối hai đỉnh của đồ thị, ta còn có thêm ma trận $P^* = P_n$ cho ta xác định đường đi ngắn nhất này: đường đi ngắn nhất từ i đến j được xác định bởi dãy:

i, $P^*[i,j]$, $P^*[P^*[i,j],j]$, $P^*[P^*[P^*[i,j],j],j]$,…,j

THỦ TỤC FLOYD(L, P)

```
for (k = 1;k≤ n;k++)
    for (i = 1;i≤ n; i++)
        for (j = 1;j≤ n; j++)
            if(a[i,k] + a[k,j] < a[i,j])
                {
                    a[i,j] := a[i,k] + a[k,j];
                    p[i,j] := p[k,j];
                 }
```

Ví dụ:

Xét đồ thị có hướng G xác định bởi ma trận trọng số W như sau:

$$W = \begin{bmatrix} 0 & 2 & 2 & 6 \\ 0 & 0 & -2 & 0 \\ 3 & 5 & 0 & 5 \\ -4 & -1 & 1 & 0 \end{bmatrix}$$

Hãy xác định ma trận A^* và P^*, sau đó chỉ ra đường đi ngắn nhất từ đỉnh 4 đến đỉnh 3.

$$A_0 = \begin{bmatrix} 0 & 2 & 2 & 6 \\ \infty & 0 & -2 & \infty \\ 3 & 5 & 0 & 5 \\ -4 & -1 & 1 & 0 \end{bmatrix} \qquad P_0 = \begin{bmatrix} 0 & 2 & 3 & 4 \\ 0 & 0 & 3 & 0 \\ 1 & 2 & 0 & 4 \\ 1 & 2 & 3 & 0 \end{bmatrix}$$

$$A_1 = \begin{bmatrix} 0 & 2 & 2 & 6 \\ \infty & 0 & -2 & \infty \\ 3 & 5 & 0 & 5 \\ -4 & -2 & -2 & 0 \end{bmatrix} \qquad P_1 = \begin{bmatrix} 0 & 2 & 3 & 4 \\ 0 & 0 & 3 & 0 \\ 1 & 2 & 0 & 4 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} 0 & 2 & 0 & 6 \\ \infty & 0 & -2 & \infty \\ 3 & 5 & 0 & 5 \\ -4 & -2 & -4 & 0 \end{bmatrix} \qquad P_2 = \begin{bmatrix} 0 & 2 & 2 & 4 \\ 0 & 0 & 3 & 0 \\ 1 & 2 & 0 & 4 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

$$A_3 = \begin{bmatrix} 0 & 2 & 0 & 5 \\ 1 & 0 & -2 & 3 \\ 3 & 5 & 0 & 5 \\ -4 & -2 & -4 & 0 \end{bmatrix} \qquad P_3 = \begin{bmatrix} 0 & 2 & 2 & 2 \\ 3 & 0 & 3 & 3 \\ 1 & 2 & 0 & 4 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

$$A_4 = A^* = \begin{bmatrix} 0 & 2 & 0 & 5 \\ -1 & 0 & -2 & 3 \\ 1 & 3 & 0 & 5 \\ -4 & -2 & -4 & 0 \end{bmatrix} \qquad P_4 = P^* = \begin{bmatrix} 0 & 2 & 2 & 2 \\ 3 & 0 & 3 & 3 \\ 4 & 4 & 0 & 4 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

Ta có: P[4,3] = 1

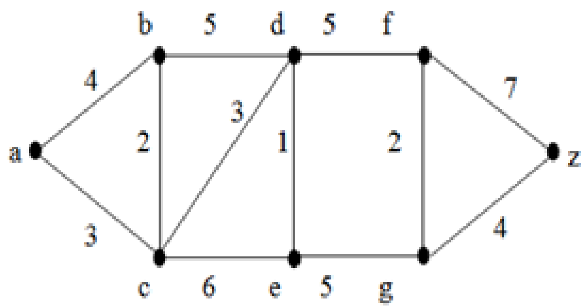
P[1,3] = 2

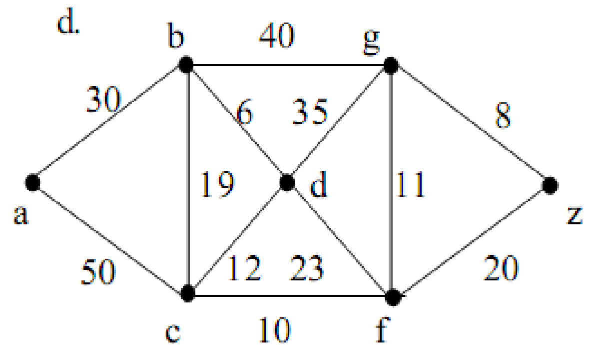
P[2,3] = 3

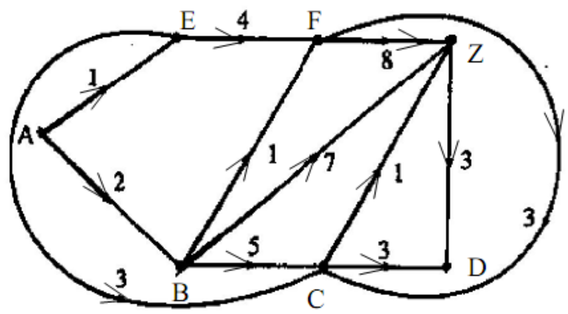
Vậy ĐĐNN từ 4 đến 3 là: 4 – 1 – 2 – 3 có độ dài là -4

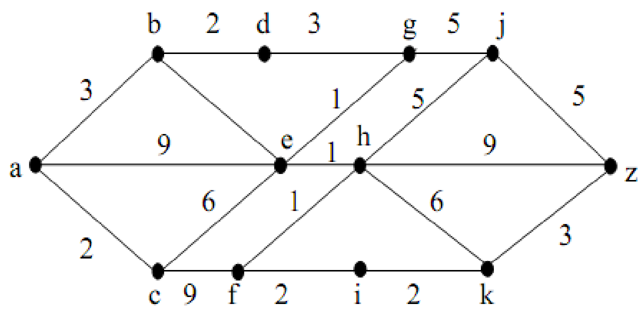
❖ Bài tập củng cố:

Bài 1. Tìm độ dài đường đi ngắn nhất giữa **a** và **z** trong các đồ thị có trọng số sau:

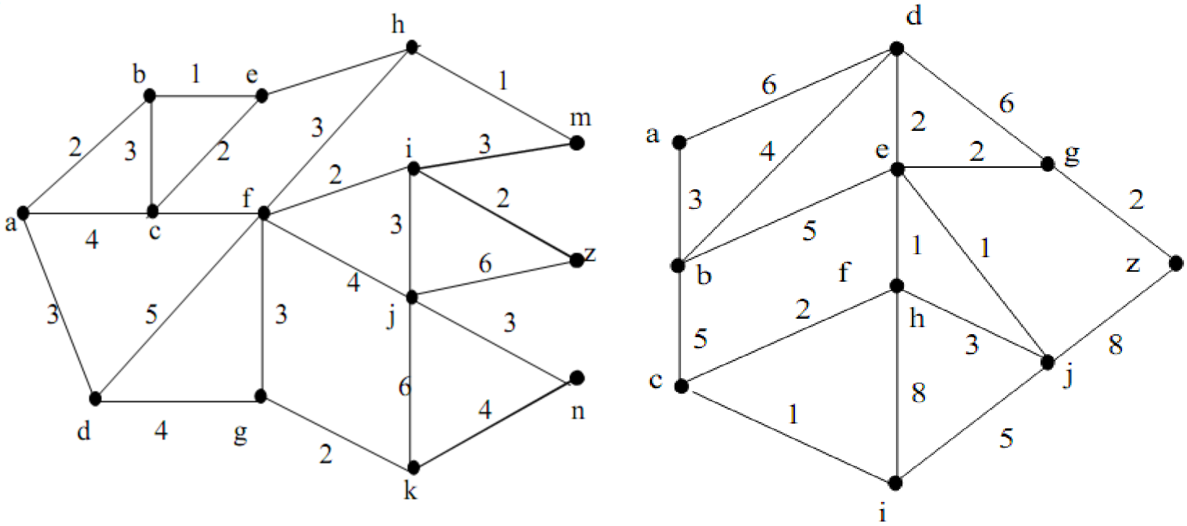




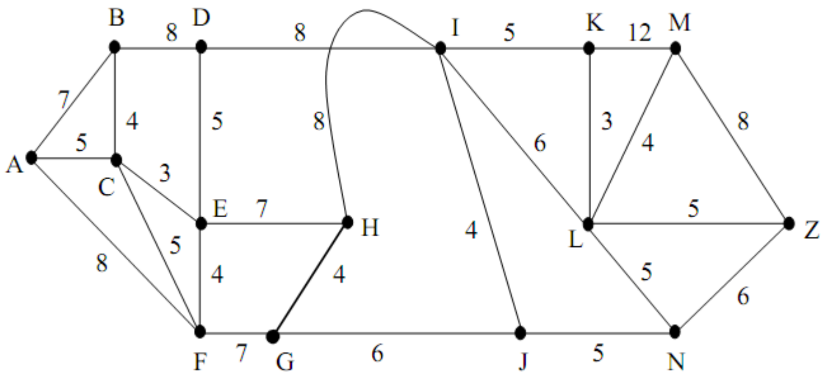




f.



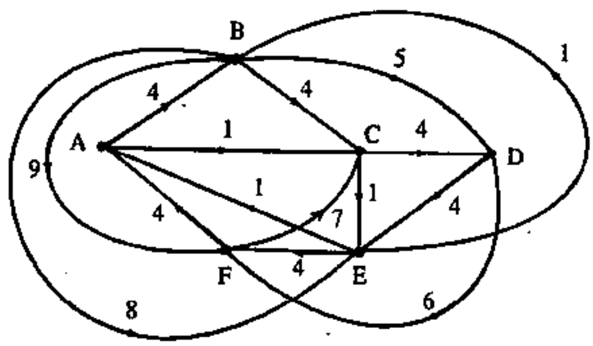
Bài 2. Tìm đường đi ngắn nhất giữa **a** và **z** của đồ thị sau, với điều kiện

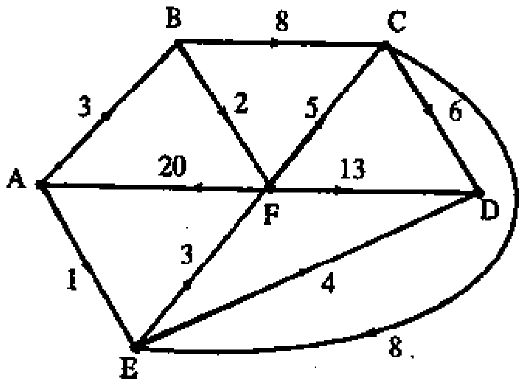


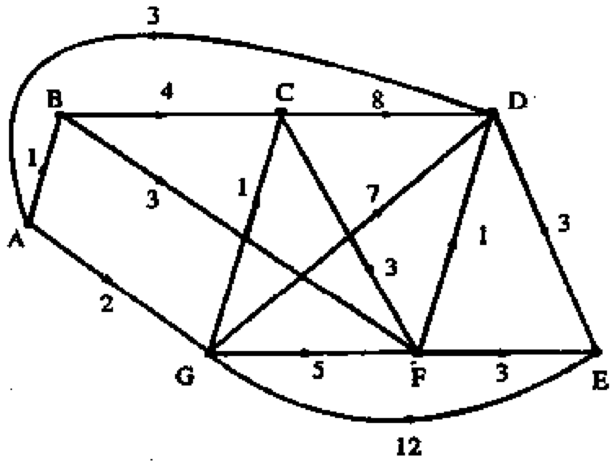
a. Đi qua đỉnh **H.**

b. Chứa cạnh **IJ.**

Bài 3. Áp dụng thuật toán Floyd để giải các bài toán sau







$$
\begin{array}{c|c}
 & \begin{array}{ccccc} A & B & C & D & E \end{array} \\
\begin{array}{c} A \\ B \\ C \\ D \\ E \end{array} &
\begin{bmatrix}
- & 5 & 8 & 13 & 6 \\
4 & - & 7 & 10 & 9 \\
6 & 3 & - & 2 & 7 \\
8 & 4 & 5 & - & 4 \\
12 & 8 & 13 & 3 & -
\end{bmatrix}
\end{array}
$$

CHƯƠNG 3
ĐỒ THỊ PHẲNG VÀ TÔ MÀU ĐỒ THỊ

<u>BÀI 1</u>
ĐỒ THỊ PHẲNG

❖ **Mục tiêu học tập:** Sau khi học xong bài này, người học có thể:
- Xác định đồ thị phẳng

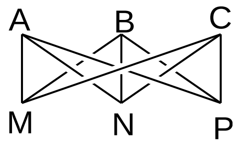
1. Bài toán mở đầu

Để nghiên cứu về đồ thị phẳng, ta bắt đầu bằng việc xét bài toán "***Ba nhà ba giếng***" như sau:

Có ba nhà ở gần ba cái giếng, từ mỗi nhà có đường đi thẳng đến từng giếng, nhưng không có đường nối thẳng các nhà với nhau, cũng như không có đường nối thẳng các giếng với nhau. Có lần bất hòa với nhau, họ tìm cách làm các đường khác đến giếng sao cho các đường này đôi một không giao nhau. Họ có thực hiện được ý định đó không?

Ta xây dựng đồ thị G=(V,E) mô tả đầy đủ các thông tin của bài toán:

- Đỉnh: Lấy các điểm trên mặt phẳng hay trong không gian tương ứng với các gia đình và các giếng nước. Đối tượng của bài toán ở đây được chia làm hai loại là gia đình và giếng nước. Vậy, mỗi đỉnh $v \in V$ biểu diễn cho một gia đình hoặc một giếng nước.
- Cạnh: Trong đồ thị G các đỉnh v_i và v_j được nối với nhau bằng một cạnh nếu có đường nối thẳng (trực tiếp) từ một gia đình đến một giếng nước. Vậy, mối quan hệ giữa 02 đối tượng ở đây là mối quan hệ đường đi. Mỗi cạnh $e \in E$ nối 2 đỉnh v_i (đại diện cho 01 gia đình) và v_j (đại diện cho một giếng nước) trong G nếu có đường đi trực tiếp từ gia đình v_i đến giếng nước v_j. Ta có đồ thị G như sau:



Hình 1. Mô hình hóa bài toán 3 nhà 3 giếng

Khi giải quyết bài toán trên ta cần đến khái niệm đồ thị phẳng như sau:

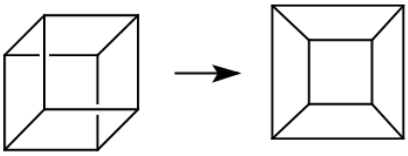
2. Đồ thị phẳng

Định nghĩa

Một đồ thị được gọi là phẳng nếu nó có thể vẽ được trên một mặt phẳng mà không có các cạnh nào cắt nhau ở điểm không phải là điểm mút của mỗi cạnh. Hình vẽ như vậy được gọi là một biểu diễn phẳng của đồ thị.

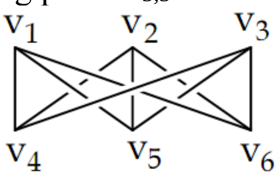
Ví dụ 1: K_4 là đồ thị phẳng vì có thể vẽ lại như sau:

Ví dụ 2: Q_3 cũng là đồ thị phẳng vì:

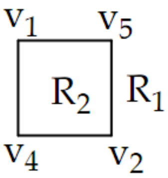


Hình 2. Đồ thị phẳng K_4 (ví dụ 1) và Q_3 (ví dụ 2).

Ví dụ 3: Ta sẽ xét xem đồ thị lưỡng phân $K_{3,3}$ có là đồ thị phẳng không?



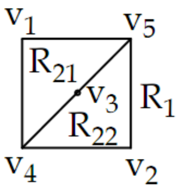
Ta sẽ chứng minh $K_{3,3}$ là đồ thị không phẳng. Thật vậy, ta thấy trong một biểu diễn phẳng bất kỳ của $K_{3,3}$ thì v_1 và v_2 đều nối với v_4 và v_5. Bốn đỉnh này tạo thành một đường khép kín chia mặt phẳng ra làm hai miền R_1 và R_2 như sau:



Ta còn lại đỉnh v_6. Có 3 trường hợp đối với v_6:

Nếu v_6 nằm trong R_1 thì cạnh nối v_3 với v_6 sẽ cắt ít nhất một cạnh khác.

Nếu v_6 nằm trong R_{21}: cạnh nối v_6 với v_5 sẽ cắt ít nhất một cạnh khác.



Nếu v_6 nằm trong R_{22}: cạnh nối v_6 với v_1 sẽ cắt ít nhất một cạnh khác.

$\Rightarrow$ v_3 không thể nằm trong R_2.

Tương tự ta cũng chứng minh được nếu v_3 nằm trong R_1 thì đồ thị cũng không phẳng.

Vậy, $K_{3,3}$ là đồ thị không phẳng.

Khi ta kết luận $K_{3,3}$ là đồ thị không phẳng, ta cũng đã giải quyết được bài toán ***"ba nhà ba giếng"***. Không có đường đi nào nối mỗi nhà với 3 giếng mà không cắt nhau. Hay nói cách khác, ba nhà ba giếng nói trên không thể nối với nhau trên một mặt phẳng mà không cắt nhau.

3. Công thức Euler

Ta thấy khi biểu diễn phẳng một đồ thị, ta chia mặt phẳng thành các miền, kể cả miền vô hạn. Euler đã chứng minh rằng tất cả các biểu diễn phẳng của một đồ thị đều chia mặt phẳng thành cùng một số miền như nhau. Euler cũng đã tìm ra mối liên hệ giữa số miền; số đỉnh và số cạnh của một đồ thị phẳng.

3.1 Định lý 1 (Công thức Euler về đồ thị phẳng - liên thông)

Cho G là một đơn đồ thị phẳng liên thông với e cạnh, v đỉnh. Gọi r là số miền (regions) trong biểu diễn phẳng của G. Khi đó:

$$\mathbf{r = e - v + 2} \text{ hay } \quad \mathbf{v - e + r = 2.}$$

Chứng minh:

Giả sử ta đã có biểu diễn phẳng của G. Ta sẽ chứng minh bằng cách xây dựng một dãy các đồ thị con của G: $G_1, G_2, ..., G_e = G$ bằng cách ghép thêm một cạnh vào đồ thị ở bước trước. Điều này là làm được vì ta có thể lấy bất kỳ một cạnh của G để được G_1. Sau đó, ta nhận được G_{n+1} từ G_n bằng cách thêm vào G_n một cạnh có một đỉnh liên thuộc với G_n và một đỉnh khác liên thuộc với cạnh mới đó. Điều này luôn làm được do G là liên thông. Ta sẽ nhận được đồ thị G sau e bước (vì G có e cạnh).

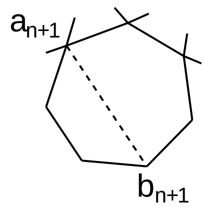
Gọi r_n, e_n và v_n tương ứng là số miền, số cạnh và số đỉnh của biểu diễn phẳng của G_n; n = 1, ..., e.

Ta sẽ chứng minh bằng qui nạp:

Đối với G_1 ta có: $e_1 = 1$; $v_1 = 2$; $r_1 = 1$. Do đó: $r_1 = e_1 - v_1 + 2$.

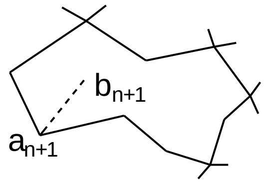
Giả sử đã có $r_n = e_n - v_n + 2$. Gọi $a_{n+1}b_{n+1}$ là cạnh ghép thêm vào G_n để có G_{n+1}. Khi đó có hai trường hợp có thể xảy ra:

Nếu cả hai đỉnh a_{n+1} và b_{n+1} đều thuộc G_n:



Do G_{n+1} là đồ thị phẳng nên $a_{n+1}b_{n+1}$ không thể cắt bất cứ cạnh nào của $G_{n+1} \Rightarrow a_{n+1}$ và

b_{n+1} phải nằm trên biên của miền chung R $\Rightarrow$ cạnh $a_{n+1}b_{n+1}$ chia R ra làm hai miền con.
Khi đó: $r_{n+1} = r_n + 1$; $e_{n+1} = e_n + 1$.
$V_{n+1} = v_n \Rightarrow r_{n+1} = e_{n+1} - v_{n+1} + 2$.
+ Nếu một trong a_{n+1} hoặc b_{n+1} không phụ thuộc G_n: Không mất tính tổng quát, ta có thể giả sử $a_{n+1} \in G_n$ và $b_{n+1} \notin G_n$. Khi đó ta thấy khi thêm cạnh $a_{n+1}b_{n+1}$ vào G_n để được G_{n+1} sẽ không thêm miền mới nào vì a_{n+1} phải ở trên miền có a_{n+1} ở trên biên của nó.



Do đó: $r_{n+1} = r_n$; $v_{n+1} = v_n + 1$; $e_{n+1} = e_n + 1 \Rightarrow r_{n+1} = e_{n+1} - v_{n+1} + 2$.
Vậy, $\forall n = 1, 2, ..., e$ ta luôn có: $r_n = e_n - v_n + 2$
hay $r = e - v + 2$.
Ví dụ: Cho G là một đơn đồ thị liên thông có 8 đỉnh và mỗi đỉnh đều có bậc là 3. Khi đó biểu diễn phẳng của đồ thị sẽ chia mặt phẳng thành bao nhiêu miền?
Giải:
Ta có: v = 8, theo giả thiết mỗi bậc có đỉnh bằng 3 nên ta có tổng số bậc của đỉnh:

$$3v = 8.3 = 24$$

mà tổng số bậc của đỉnh = 2e
$\Rightarrow 2e = 24 \Rightarrow e = 12$
$\Rightarrow$ số miền của G: $r = e - v + 2 = 12 - 8 + 2 = 6$.

3.1.1 Hệ quả 1

Cho G là một đơn đồ thị phẳng liên thông với e cạnh và v đỉnh; $v \geq 3$. Khi đó:

$$e \leq 3v - 6.$$

Chứng minh:
Trong một đồ thị phẳng, liên thông thì:
+ Mỗi miền r được bao bởi ít nhất là 3 cạnh,
+ Mỗi cạnh nằm trên nhiều nhất 2 miền.
Vậy trong một đồ thị phẳng, ta có: $3r \leq 2e$ hay: $r \leq \frac{2}{3}e$ (1)

Mặt khác, theo định lý Euler về đồ thị phẳng liên thông, ta có:
$v - e + r = 2 \quad \Rightarrow \quad r = e - v + 2 \quad (2)$
Thay (1) vào (2), ta được: $\frac{2}{3}e \geq e - v + 2 \Rightarrow 2e \geq 3e - 3v + 6 \Rightarrow e \leq 3v - 6$

Ví dụ: Chứng minh rằng đồ thị K_5 là không phẳng.

Giải: Ta có đồ thị K_5 có 5 đỉnh và 10 cạnh $\Rightarrow$ bất đẳng thức $e \le 3v - 6$ không đúng với K_5. Do đó, K_5 là đồ thị không phẳng.

3.1.2 Hệ quả 2

Cho G là một đơn đồ thị phẳng liên thông có e cạnh, v đỉnh; $v \ge 3$ và không có chu trình độ dài 3. Khi đó $e \le 2v - 4$.

Chứng minh

Trong một đồ thị phẳng, liên thông e cạnh, v đỉnh; $v \ge 3$ và không có chu trình độ dài 3 thì:

+ Mỗi miền r được bao bởi ít nhất là 4 cạnh,

+ Mỗi cạnh nằm trên nhiều nhất 2 miền.

Vậy trong một đồ thị phẳng, ta có: $4r \le 2e$ hay: $r \le \frac{1}{2}e$ (1)

Mặt khác, theo định lý Euler về đồ thị phẳng liên thông, ta có:

$v - e + r = 2 \quad \Rightarrow \quad r = e - v + 2 \quad (2)$

Thay (1) vào (2), ta được:

$$\frac{1}{2}e \ge e - v + 2 \quad \Rightarrow \quad e \ge 2e - 2v + 4 \Rightarrow e \le 2v - 4$$

Ví dụ: Chứng minh rằng $K_{3,3}$ là không phẳng.

Giải: Ta có $K_{3,3}$ không có chu trình độ dài 3. Hơn nữa, $K_{3,3}$ có 6 đỉnh và 9 cạnh $\Rightarrow$ e = 9 và 2v – 4 = 8. Điều này không thỏa hệ quả 2 $\Rightarrow$ $K_{3,3}$ không phẳng.

3.1.3 Hệ quả 3

Cho G là một đơn đồ thị phẳng liên thông có e cạnh, v đỉnh thì G phải có ít nhất một đỉnh w có $\deg(w) \le 5$

Chứng minh

Theo nhận xét của hệ quả 1, chúng ta có trọng một đồ thị phẳng liên thông thì ta có bất đẳng thức:

$$3r \le 2e \quad (1)$$

Giả sử G là một đồ thị phẳng, liên thông có e cạnh, v đỉnh mà $\forall w \in W$ đều có $\deg(w) \ge 6$. Điều này có nghĩa là mỗi đỉnh của G sẽ là đầu mút của ít nhất 6 cạnh và mỗi cạnh liên thuộc với 2 đỉnh. Từ đó, ta có:

$$6v \le 2e \text{ hay } 3v \le e \quad (2)$$

Cộng (1) và (2) vế theo vế, chúng ta có:

$$3r + 3v \le 3e \quad \Rightarrow \quad r + v \le e \quad \Rightarrow \quad r + v - e \le 0\,(!)$$

Vậy trong một đồ thị phẳng, liên thông có e cạnh, v đỉnh thì phải có ít nhất một đỉnh w có $\deg(w) \le 5$

3.2 Định lý 2 (Công thức Euler về đồ thị phẳng bất kỳ)

Cho G là một đơn đồ thị phẳng với e cạnh, v đỉnh và có $k \ge 1$ thành phần liên thông. Gọi r là số miền (regions) trong biểu diễn phẳng của G. Khi đó:

$$\mathbf{v - e + r = k + 1.}$$

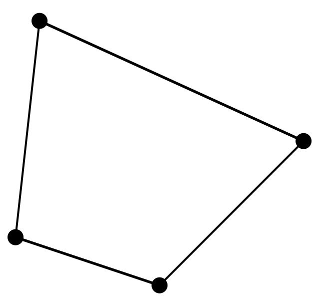
Đây chính là công thức Euler cho một đồ thị phẳng bất kỳ.

3.3 Định lý Kuratowski

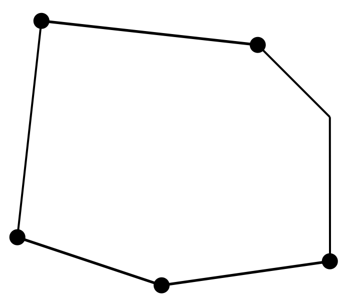
Ta đã biết $K_{3,3}$ và K_5 không phẳng nên một đồ thị là không phẳng nếu nó chứa một trong hai đồ thị này như là các đồ thị con. Hơn thế nữa, tất cả các đồ thị không phẳng cần phải chứa đồ thị con nhận được từ $K_{3,3}$ hoặc K_5 bằng một số phép toán nào đó.

Nếu một đồ thị G là phẳng, nếu bỏ đi cạnh ab của G và thêm vào đỉnh mới c cùng với 2 cạnh ac và cb thì đồ thị nhận được cũng là phẳng. Ta gọi phép toán như trên là ***phép phân chia sơ cấp***. Hai đồ thị nhận được từ cùng một đồ thị bằng một dãy các phép phân chia sơ cấp được gọi là ***đồng phôi*** với nhau.

Ví dụ: Hai đồ thị G_1 và G_2 sau đây là đồng phôi với nhau vì cùng được tạo ra từ K_3 bằng một số phép phân chia sơ cấp



G_1

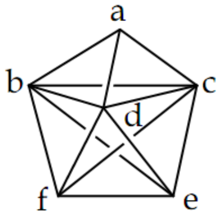


G_2

Định lý Kuratowski:

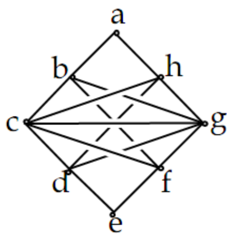
Đồ thị G là không phẳng khi và chỉ khi G chứa một đồ thị con đồng phôi với $K_{3,3}$ hoặc K_5.

Ví dụ 1: Xét đồ thị G sau:



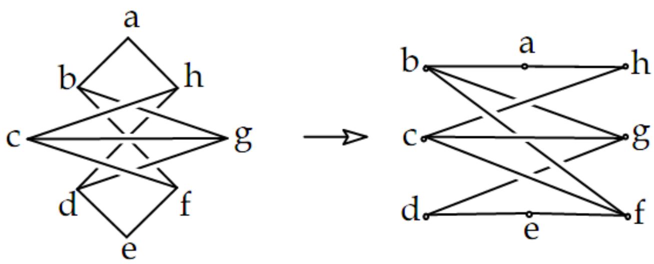
Nếu ta xóa đỉnh a của G, đồng thời xóa đi cạnh ab, ac, ad thì ta được một đồ thị con của G là đồ thị $K_5 \Rightarrow$ G không phẳng.

Ví dụ 2: Cho đồ thị G như sau:



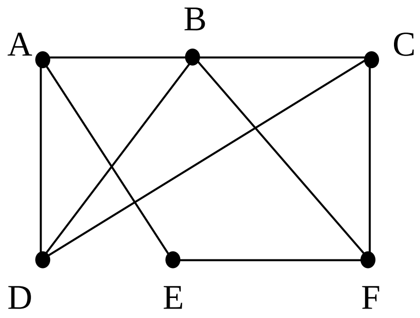
Nếu ta bỏ đi các cạnh bc, cd, fg, gh ta được đồ thị con của G:

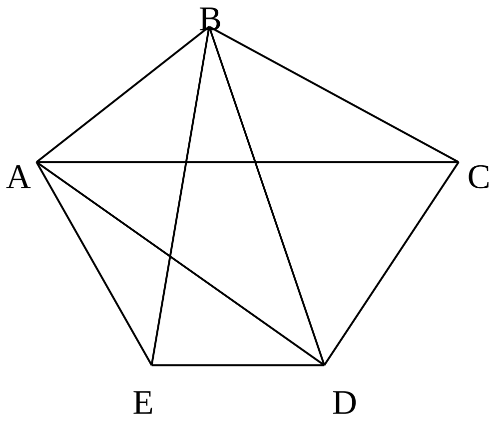
Đồ thị này đồng phôi với $K_{3,3} \Rightarrow$ G không phẳng.

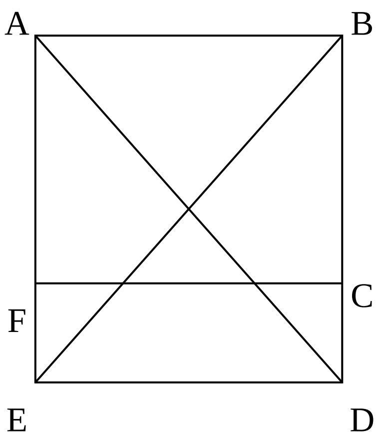


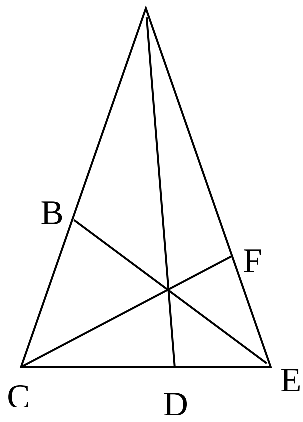
❖ Bài tập củng cố:

Bài 1. Các đồ thị sau có là phẳng hay không? Nếu có hãy vẽ nó không có cạnh cắt nhau:







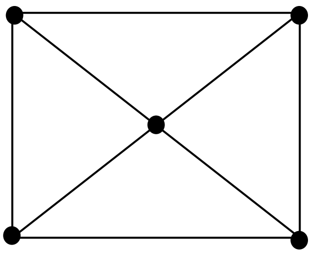


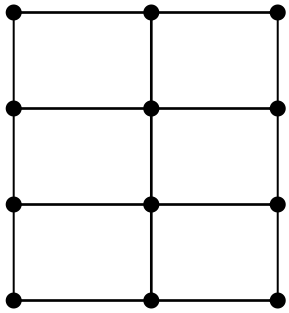
Bài 2. Các đồ thị sau có là phẳng hay không

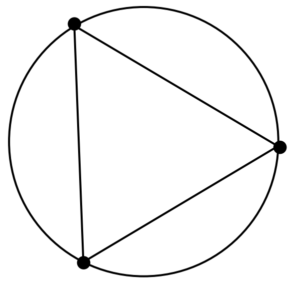
a. $\begin{bmatrix} & & 1 & 1 & 1 & \\ & & & 1 & 1 & 1 \\ 1 & & & & 1 & 1 \\ 1 & 1 & & & & 1 \\ 1 & 1 & 1 & & & \\ & 1 & 1 & 1 & & \end{bmatrix}$

b. $\begin{bmatrix} & 1 & & & & 1 & 1 \\ 1 & & 1 & & 1 & & \\ & 1 & & 1 & & & 1 \\ & & 1 & & 1 & 1 & \\ & 1 & & 1 & & 1 & 1 \\ 1 & & & 1 & 1 & & \\ 1 & & 1 & & 1 & & \end{bmatrix}$

Bài 3. Tìm số đỉnh, cạnh và miền của các đồ thị sau:

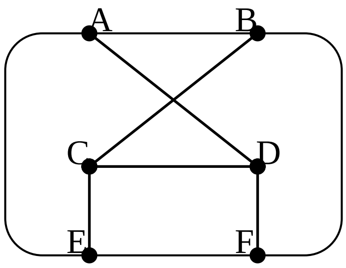


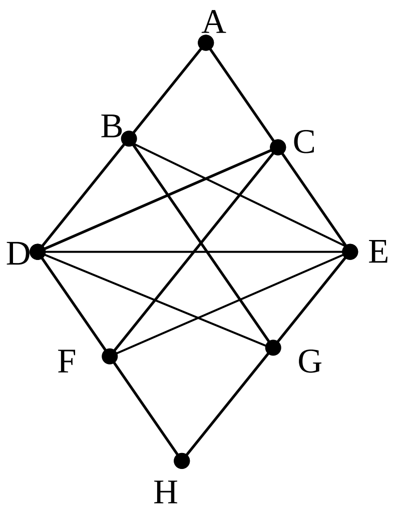


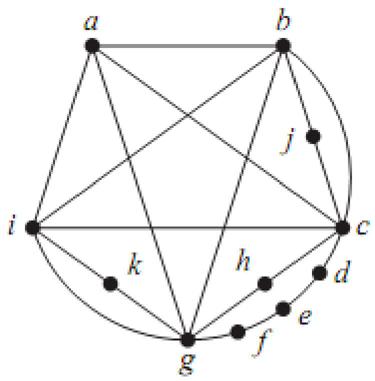


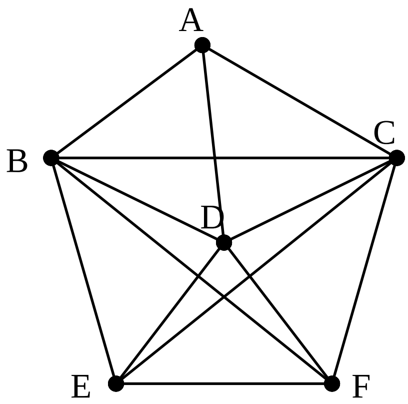
Bài 4. Có năm ngôi nhà nối với hai thiết bị sinh hoạt: Gas và Điện. Hỏi có cách nối để các dây nối không cắt nhau hay không? Tại sao?

Bài 5. Chứng minh các đồ thị sau là không phẳng:









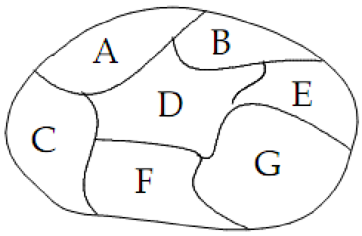
BÀI 2
TÔ MÀU ĐỒ THỊ

❖ **Mục tiêu học tập:** Sau khi học xong bài này, người học có thể:
- Xác định bài toán quy về bài toán tô màu
- Tô màu đồ thị

1. Bài toán mở đầu

Khi tô màu một bản đồ, hai miền có chung biên giới được tô bằng hai màu tùy ý, miễn là khác nhau. Để đảm bảo chắc chắn hai miền kề nhau không bao giờ có màu trùng nhau, ta có thể tô mỗi miền bằng một màu khác nhau. Rõ ràng, điều này là không cần thiết vì đối với nhiều bản đồ, ta có thể dùng số màu ít hơn số miền nhưng vẫn đảm bảo hai miền kề nhau có màu khác nhau. Do đó bài toán đặt ra là "**Xác định số màu tối thiểu cần có để tô màu một bản đồ sao cho hai miền kề nhau có màu khác nhau**".

Ví dụ: Ta có bản đồ:



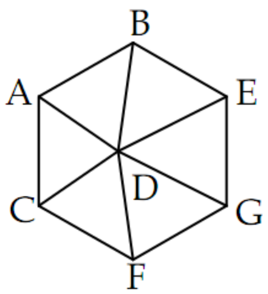
Hình 1. Bản đồ các miền

Đối với bản đồ này, ta cần ba màu là đủ (hai màu là không đủ).

Để tô màu một bản đồ, chúng ta sẽ chuyển bản đồ về dưới dạng đồ thị. Mỗi bản đồ trên mặt phẳng có thể biểu diễn bằng một đồ thị. Trong đó:

- Mỗi miền của bản đồ được biểu diễn bằng một đỉnh của đồ thị.
- Các cạnh nối hai đỉnh nếu các miền được biểu diễn bằng hai đỉnh này có biên giới chung. Hai miền chung nhau chỉ một điểm, không được coi là kề nhau.

Đồ thị nhận được từ bản đồ bằng cách xây dựng trên được gọi là **đồ thị đối ngẫu** của bản đồ đang xét. Ví dụ bản đồ nêu trên có đồ thị đối ngẫu như sau:



Hình 2. Bản đồ đối ngẫu của bản đồ ở hình 1

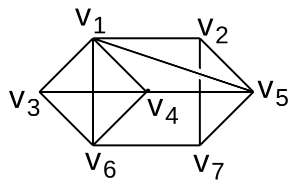
2. Tô màu đồ thị

Định nghĩa

Tô màu một đơn đồ thị là việc gán màu cho các đỉnh của nó sao cho hai đỉnh liền kề có màu khác nhau. Mỗi đồ thị có thể có nhiều cách tô màu khác nhau.

Số màu hay sắc số (Chromatic number) của một đồ thị G là số màu tối thiểu cần thiết để tô màu G. Ký hiệu: $\chi(G)$

Ví dụ: Xét đồ thị G:



Để tô màu đồ thị G, trước hết ta thấy v_1 có bậc cao nhất, $deg(v_1) = 5$ cho nên ta cho v_1 có màu a. $deg(v_5) = 4 \Rightarrow$ cho v_5 có màu b. Cho v_4 màu c. Vì $deg(v_4) = 4$ và v_4, v_5 kề nhau $\Rightarrow$ v_6 có màu b. Còn v_3 có màu d. Còn lại 2 đỉnh v_2 và v_7. Ta thấy v_2 kề với v_1 và $v_5 \Rightarrow$ cho v_2 màu c; v_7 không kề với v_1 nên ta cho v_7 màu a.

Vậy ta có đỉnh:	v_1	v_2	v_3	v_4	v_5	v_6	v_7
màu:	a	c	d	c	b	b	a

$\Rightarrow$ ta sử dụng 4 màu để tô màu G.

Ta thấy G có 4 đỉnh v_1, v_3, v_4, v_6 đôi một kề nhau $\Rightarrow \chi(G) \geq 4$. Theo trên ta chỉ dùng 4 màu $\Rightarrow \chi(G) = 4$

3. Một số định lý về tô màu đồ thị

3.1 Định lý 1

Mọi đơn đồ thị đầy đủ K_n đều có: $\chi(K_n) = n$.

Chứng minh

Khẳng định được chứng minh bằng quy nạp theo số đỉnh của đồ thị.

- *Trường hợp cơ sở :* Với n = 1, K_1 có 1 đỉnh nên phải dùng 1 màu để tô.
- *Giả thiết quy nạp*: Giả sử khẳng định đúng với n = k. Nghĩa là mọi đơn đồ thị đủ có k đỉnh đều có $\chi(K_k) = k$. Ta cần khẳng định tính đúng đắn của định lý đối với n = k + 1. Nghĩa là $\chi(K_{k+1}) = k + 1$.

  Giả sử K_{k+1} là một đơn đồ thị đầy đủ với tập đỉnh:

  $$V = \{v_1, v_2, \cdots, v_k, v_{k+1}\}$$

  Ta loại khỏi K_{k+1} một đỉnh tuỳ ý (chẳng hạn đỉnh v_{k+1}) cùng các cạnh liên thuộc với đỉnh này. Đồ thị con nhận được là một đơn đồ thị đầy đủ có k đỉnh. Theo giả thuyết quy nạp chúng ta có $\chi(K_k) = k$.

  Bây giờ ta "khôi phục" lại đỉnh vk_{+1} cùng với các cạnh liên thuộc với nó, tức là “trở lại” đồ thị K_{k+1}. Vì đỉnh v_{k+1} kề với tất cả các đỉnh còn lại nên để tô màu cho v_{k+1} ta phải sử dụng một màu mới. Do đó: $\chi(K_{k+1}) = k + 1$.

3.2 Định lý 2

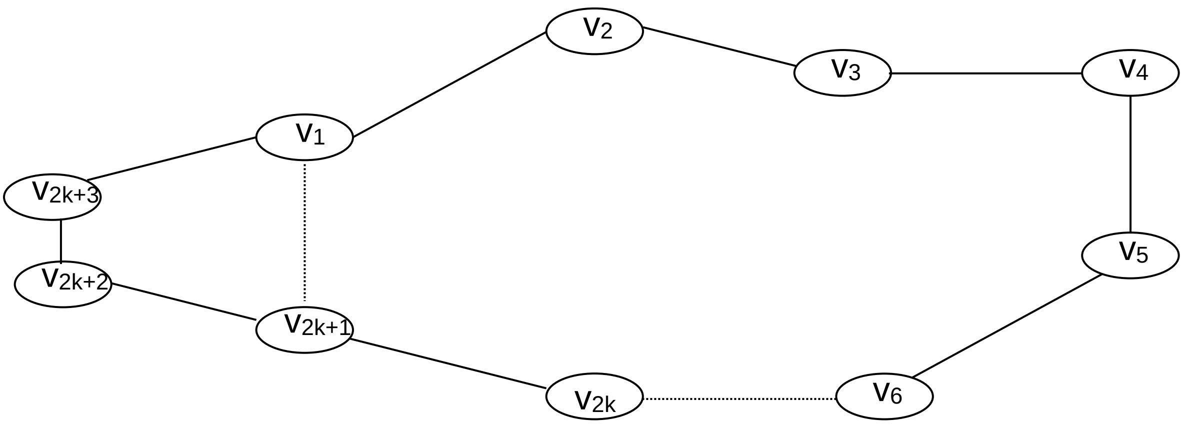
Mọi chu trình độ dài lẻ đều có sắc số là 3.

Chứng minh

Giả sử α là một chu trình độ dài lẻ tuỳ ý. Khi đó, tồn tại một số tự nhiên n để $|\alpha| = 2n+1$. Giả sử dãy các đỉnh của α là: $V = \{v_1, v_2, ..., v_{2n}, v_{2n+!}\}$. Ta sẽ chứng minh khẳng định trên bằng quy nạp theo n.

- *Trường hợp cơ sở:* Với n = 1. Chu trình α gồm 3 đỉnh v_1, v_2, v_3. Do mỗi đỉnh v_i ($1 \le i \le 3$) đều kề với 2 đỉnh còn lại, nên ta phải dùng đúng 3 màu khác nhau để tô cho α vì 2 đỉnh kề nhau tuỳ ý đều phải có màu khác nhau.
- *Giả thiết quy nạp*: Giả sử khẳng định đã đúng với $n \le k$, nghĩa là với một chu trình α_1 tuỳ ý với độ dài 2n + 1 ($1 \le n \le k$) đều có sắc số bằng 3. Ta chỉ cần chỉ ra rằng với n = k + 1 khẳng định vẫn đúng. Nghĩa là chu trình α có độ dài 2(k + 1) + 1 cũng có sắc số bằng 3.

  Giả sử α là chu trình có độ dài lẻ tùy ý có độ dài bằng 2(k + 1) + 1 và có tập đỉnh: $V = \{v_1, v_2, ..., v_{2k}, v_{2k+!}, v_{2k+2}, v_{2k+3}\}$ ta mô tả chu trình α như hình vẽ sau:

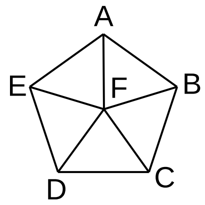


  Nối đỉnh v_1 với đỉnh v_{2k+1} ta được chu trình α_1 với độ dài lẻ là 2k+1. Theo giả thuyết quy nạp thì α_1 có sắc số là 3 và hai đỉnh v_1 và v_{2k+1} phải có màu khác nhau. Chẳng hạn v_1 được tô bằng màu M_1 và đỉnh v_{2k+1} được tô bằng màu M_2. Khi đó, để tô đỉnh v_{2k+2} ta có thể dùng lại màu M_1 và tô đỉnh v_{2k+3} ta có thể dùng lại màu M_2. Nghĩa là không cần dùng thêm màu mới. Vậy sắc số của α là 3 và định lý đã được chứng minh.

3.3 Định lý 3

Nếu G có chứa một đồ thị con đẳng cấu với Kn thì $\chi(G) \geq n$.

Ví dụ: Tìm sắc số của đồ thị G:



Ta có: G chứa K_3 nên $\chi(G) \geq 3$.

Ta lại có F có bậc lớn nhất nên ta tô F màu 1, A màu 2, B màu 3. Khi đó C phải tô màu 2 và D tô màu 3. Còn lại đỉnh E kề với A, F, D đã có đủ 3 màu 1, 2, 3. Do đó, E phải có màu 4.

Ta tìm sắc số của G: Do G có chứa chu trình lẻ (ABCDEA) nên ta có các đỉnh A, B, C, D, E phải được tô bằng 3 màu. Mặt khác, đỉnh F kề với tất cả các đỉnh A, B, C, D, E nên ta phải dùng màu thứ 4 để tô cho F. Vậy: $\chi(G) = 4$.

☞ ***Chú ý:***

➢ Nếu G' là một đồ thị con của G thì $\chi(G) \geq \chi(G')$.

➢ Nếu dùng k màu để tô màu G thì không cần quan tâm đến những đỉnh có bậc nhỏ hơn k.

3.4 Định lý 4

Một đơn đồ thị G=(V,E) có thể tô bằng 2 màu khi và chỉ khi nó không có chu trình độ dài lẻ.

Chứng minh

Điều kiện cần: Giả sử G là đồ thị 2 sắc (có thể tô tất cả các đỉnh của G bằng 2 màu) ngưng trong G lại có một chu trình lẻ α. Khi đó theo định lý 2, sắc số của α phải là 3. Mặt khác theo định lý 2 ta có $\chi(G) \geq \chi(\alpha) = 3$. Do đó, sắc số của G ít nhất phải là 3. Ta suy ra điều mâu thuẫn với giả thuyết, nên G không có chu trình độ dài lẻ.

Điều kiện đủ: Giả sử đồ thị G không có chu trình độ dài lẻ. Ta cần chứng minh rằng G là đồ thị 2 sắc. Ta bắt đầu tô dần các đỉnh của G theo quy tắc sau:

+ Tô M_1 cho đỉnh $w \in V$ bất kỳ.

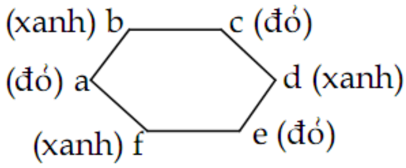
+ Nếu một đỉnh $u \in V$ nào đó đã được tô bằng M_1, ta sẽ dùng màu M_2 để tô cho tất cả các đỉnh kề với u và ngược lại nếu đỉnh $u \in V$ nào đó đã được tô bằng M_2, ta sẽ dùng màu M_1 để tô cho tất cả các đỉnh kề với u.

Vì G là đồ thị hữu hạn nên đến một lúc nào đó tất cả các đỉnh của G sẽ phải được tô màu và mỗi đỉnh của G không thể cùng lúc vừa được tô bằng M_1 vừa được tô bằng M_2. Thật vậy, giả sử trong G tồn tại đỉnh v mà theo nguyên tắc nó vừa được tô bằng

M_1 đồng thời cũng được tô bằng M_2. Khi đó v phải kề với đỉnh s được tô bằng M_1 và đỉnh t được tô bằng M_2, nên khi đó các đỉnh v, s, t phải nằm trên một chu trình độ dài lẻ. Như vậy mâu thuẫn với giả thuyết nên đỉnh v không tồn tại.

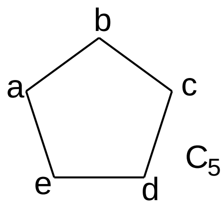
Ví dụ: Xét đồ thị vòng C_n ($n \geq 3$).

➢ Khi n chẵn, chẳng hạn n = 6. Ta có:



Dễ thấy khi đó $\chi(C_6) = 2$ vì ta có thể tô màu như trên. Rõ ràng C_6 không có chu trình lẻ.

➢ Khi n lẻ, chẳng hạn n = 5. Ta có:



C_5 có chứa chu trình lẻ $\Rightarrow \chi(C_5) \geq 3$. Dễ thấy $\chi(C_5) = 3$.

Tổng quát:

$$\begin{cases} \chi(C_n) = 2 & \text{nếu n chẵn } (n \geq 3) \\ \chi(C_n) = 3 & \text{nếu n lẻ } (n \geq 3) \end{cases}$$

3.5 Định lý 5 (Định lý bốn màu) (định lý Appel-Haken, 1976)

Mọi đồ thị phẳng đều có sắc số không lớn hơn 4.

(Định lý này là định lý đầu tiên được chứng minh với sự hỗ trợ của máy vi tính).

3.6 Thuật toán Welch-Powell về tô màu đồ thị

Để tô màu một đồ thị G, ta có thể sử dụng thuật toán Welch-Powell như sau:

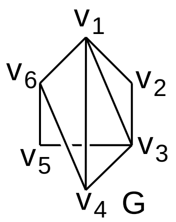
➢ Sắp xếp các đỉnh G theo bậc giảm dần.

➢ Dùng một màu để tô đỉnh đầu tiên và cũng dùng màu này để tô màu các đỉnh liên tiếp trong danh sách mà không kề với đỉnh đầu tiên.

➢ Bắt đầu trở lại đầu danh sách, tô màu thứ hai cho đỉnh chưa được tô và lập lại quá trình trên cho đến khi tất cả các đỉnh đều được tô màu.

☞ ***Chú ý:*** Thuật toán Welch-Powell chưa cho ta sắc số của một đồ thị G, nó chỉ giúp ta một cách tiếp cận để tìm sắc số của một đồ thị. Để tìm sắc số của một đồ thị thì sau khi tô màu xong ta phải sử dụng các định lý, các tính chất đã học của lý thuyết đồ thị để khẳng định số màu được dùng là ít nhất và từ đó suy ra sắc số của đồ thị. Bài toán tìm sắc số của một đô thị là một bài toán khó và không phải đồ thị nào cũng tìm được sắc số của nó một cách dễ dàng.

Ví dụ: Dùng thuật toán Welch-Powell để tô màu và tìm sắc số của đồ thị sau:



Ta có đỉnh:	v_1	v_3	v_4	v_6	v_2	v_5
Bậc:	4	4	3	3	2	2
Màu	a	b	c	b	c	a

Ta lại có G chứa đồ thị con đẳng cấu với K_3 bao gồm các đỉnh v_1, v_2, v_3 $\Rightarrow$ $\chi(G) \geq 3$.

Do G có chứa đồ thị con là K_3 nên theo định lý 2 ta có $\chi(G) \geq 3$. Ta có thể dùng 3 màu để tô G là ít nhất $\Rightarrow$ $\chi(G) = 3$.

4. Ứng dụng của bài toán tô màu

Bài toán tô màu có nhiều ứng dụng trong thực tế. Chẳng hạn: việc xếp lịch thi cho sinh viên, phân chia tần số phát sóng của các đài phát thanh truyền hình, bố trí các con vật trong sở thú,…

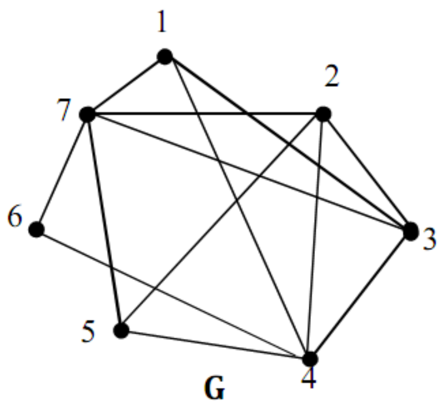
Ví dụ: Hãy lập lịch thi cho các sinh viên trong một trường đại học sao cho không có sinh viên nào phải thi 02 môn trong cùng một buổi thi và số buổi thi là ít nhất. Chẳng hạn, có 7 môn cần xếp lịch thi được đánh số lần lượt từ 1 đến 7 và các cặp môn sau có chung sinh viên dự thi:

+ 1 và 3, 1 và 4, 1 và 7.	+ 4 và 5, 4 và 6.
+ 2 và 3, 2 và 4, 2 và 5, 2 và 7.	+ 5 và 7.
+ 3 và 4, 3 và 7.	+ 6 và 7.

Giải Xây dựng đồ thị G = (V, E) mô tả bài toán:

- Mỗi đỉnh $v_i \in V$, $i = \overline{1,7}$ đại diện cho môn thi thứ i.
- Mỗi cạnh $e \in E$ nối 2 đỉnh v_i và v_j nếu có sinh viên phải thi cả hai môn được biểu diễn bằng 2 đỉnh này.

Ta có đồ thị G=(V,E) như sau:



G

Để lập lịch thi ta sẽ tiến hành tô màu cho G theo thuật toán Welch-Powell như sau:

Đỉnh:	4	7	2	3	1	5	6
Bậc:	5	5	4	4	3	3	2
Màu:	M1	M1	M2	M3	M2	M3	M2

Ta thấy G có chứa đồ thị con K_3 nên dùng 3 màu để tô cho G là ít nhất.

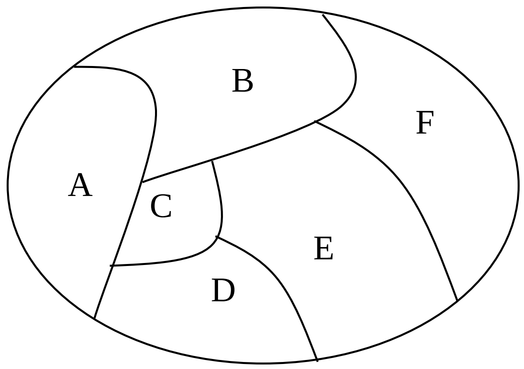
Vậy: $\chi(G) = 3$

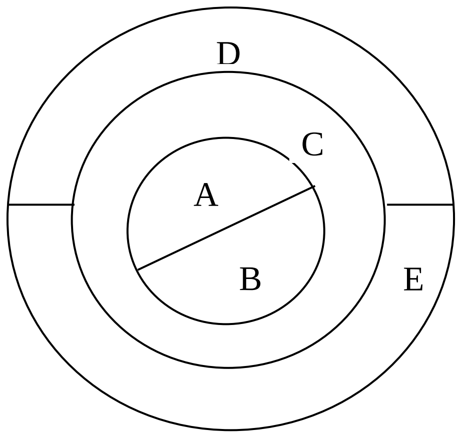
Tóm lại, ta có lịch thi như sau:

Buổi thi	**Môn thi**
I	4, 7
II	1, 2, 6
III	3, 5

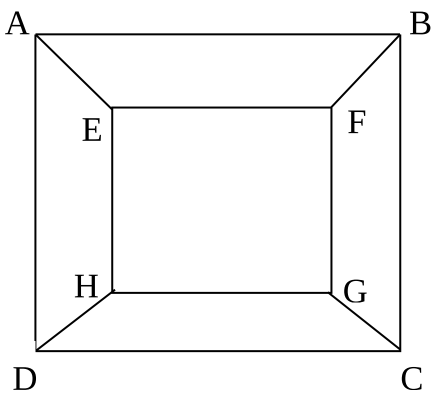
❖ **Bài tập củng cố:**

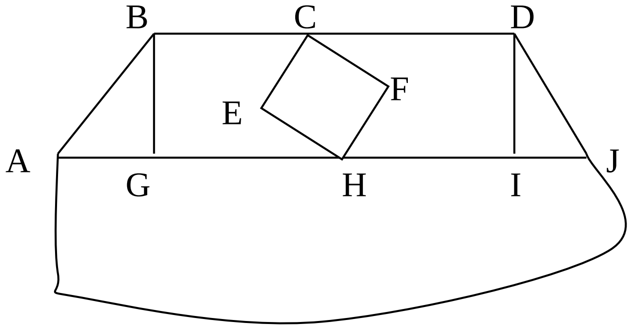
Bài 1. Xây dựng đồ thị đối ngẫu và tô màu các bản đồ sau:

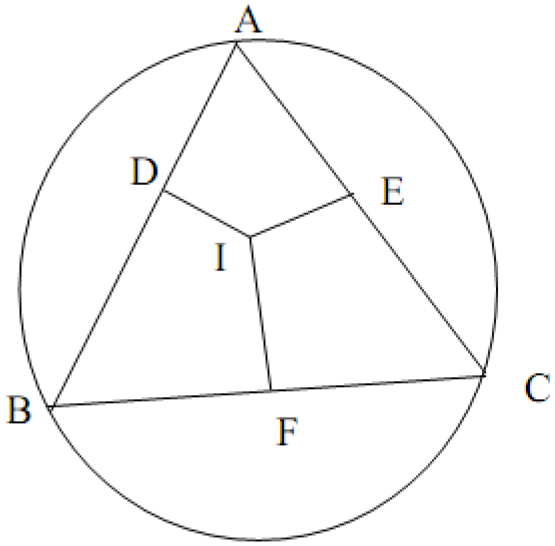


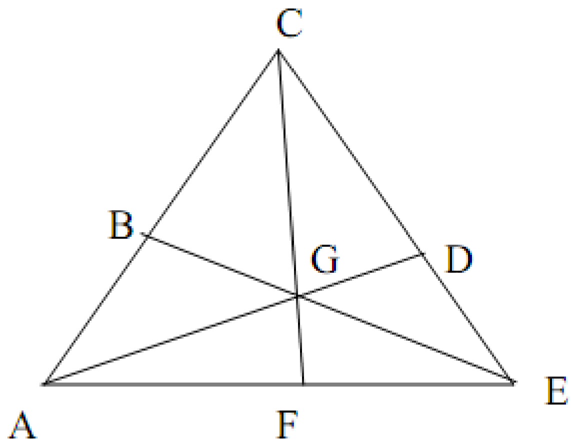


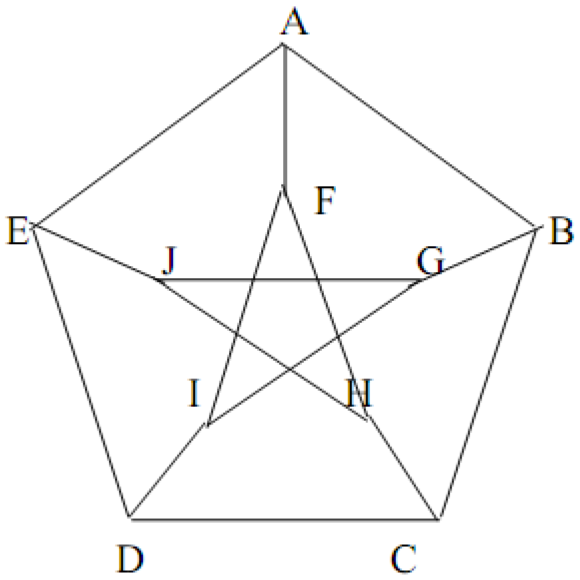
Bài 2. Tìm sắc số của các đồ thị sau:

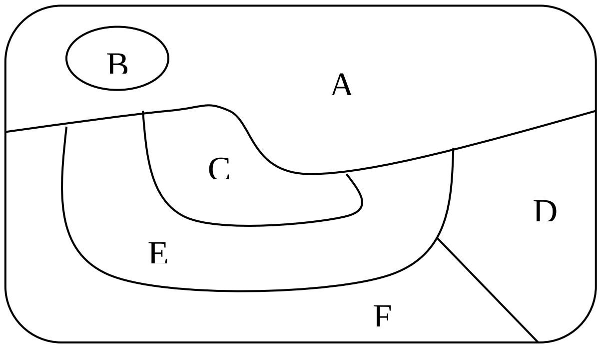












Bài 3. Một người giữ thảo cầm viên muốn sắp xếp các con vật sống theo thói quen tự nhiên của chúng. Nhưng ông ta không thể cho tất cả các con vật sống chung một chỗ vì chúng có thể ăn thịt lẫn nhau. Dấu chấm trong bảng sau chỉ ra những con vật có thể ăn thịt lẫn nhau. Số nơi nhỏ nhất người giữ thảo cầm viên cần để nuôi các con vật là bao nhiêu?

	a	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**	**j**
a		●			●					●
b	●			●			●			
c								●		●
d		●				●				
e	●								●	
f				●						●
g		●								
h			●						●	
i					●			●		●
j	●		●			●			●	

Bài 4. Sáu đài truyền hình ở cách nhau như đã cho trong bảng dưới đây. Mỗi đài sẽ được cấp 1 kênh để phát sóng. Hãy tìm số kênh ít nhất cần phát, biết rằng hai đài phát cách nhau không quá 150 dặm sẽ không được cấp phát chung một kênh.

	1	**2**	**3**	**4**	**5**	**6**
1		85	175	200	20	100
2	85		125	175	100	160
3	175	125		100	200	350
4	200	175	100		210	220
5	20	100	200	210		100
6	100	160	350	220	100	

CHƯƠNG 4
CÂY

BÀI 1
CÁC KHÁI NIỆM CƠ BẢN VỀ CÂY

❖ **Mục tiêu học tập:** Sau khi học xong bài này, người học có thể:
- Trình bày các khái niệm cơ bản và tính chất của cây

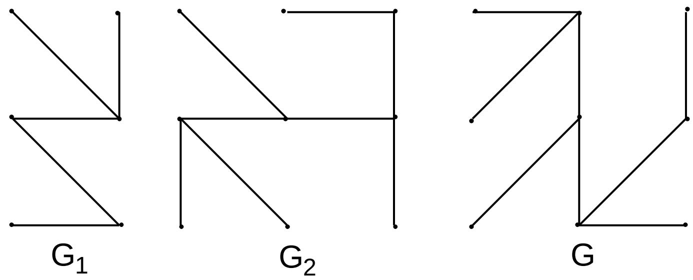
1. Cây (Tree)

Định nghĩa

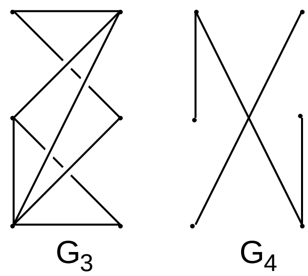
Cây là một đồ thị vô hướng, liên thông và không có chu trình sơ cấp.

Do cây không có chu trình sơ cấp, nên cây không thể có cạnh bội và khuyên. Vậy mọi cây là đơn đồ thị.

Các ví dụ



Hình 1. Các cây G_1 và G_2, G.



Hình 2. G_3 và G_4 không là cây.

+ G_3 có chứa chu trình nên G_3 không là cây,

+ G_4 không liên thông nên G_4 không là cây.

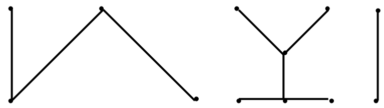
2. Rừng

Định nghĩa

Rừng là đồ thị vô hướng không có chu trình.

Từ định nghĩa, ta thấy rừng là một đồ thị có nhiều thành phần liên thông mà mỗi thành phần liên thông của nó là một cây.

Ví dụ



Hình 3. G là một rừng và G có 3 thành phần liên thông.

Định lý về điều kiện đủ của cây

Nếu trong đồ thị vô hướng G, mọi cặp đỉnh của nó luôn tồn tại một đường đi sơ cấp duy nhất thì G là một cây.

Chứng minh

Giả sử G là một cây ta sẽ chứng minh mọi cặp đỉnh trong G đều tồn tại một đường đi sơ cấp duy nhất. Thật vậy:

➢ Nếu trong G tồn tại một đường đi α giữa hai đỉnh v và w và α không là đường đi sơ cấp. Khi đó trên đường đi α sẽ tồn tại ít nhất một đỉnh u được đi lặp lại. Khi đó trong G sẽ có một chu trình:

$\beta = u..v_j v_{j+1}..u$ (1)

➢ Mặt khác, giả sử trên G có 2 đường sơ cấp khác nhau α_1 và α_2 nối 2 đỉnh v và w. Được liệt kê lần lượt như sau:

$\alpha_1 = vv_1..v_n w$ và $\alpha_2 = vv_1^{'}v_2^{'}..v_n^{'}w$

+ Gọi i là chỉ số bé nhất của các đỉnh trên đường đi sao cho $v_i v_{i+1} \neq v_i^{'} v_{i+1}^{'}$

+ Gọi j là chỉ số bé nhất ($j \geq i$) sao cho tồn tại một chỉ số $k \geq i$ để $v_k^{'} = v_j$

Ta thấy rằng các chỉ số i, j và k như thế là tồn tại và khi đó trong G tồn tại một chu trình:

$\beta^{'} = v_i v_{i+1}..v_j v_{k-1}^{'}..v_i^{'} \equiv v_i$ (2)

Vậy từ (1) và (2), ta có: Nếu trong đồ thị vô hướng G, mọi cặp đỉnh của nó luôn tồn tại một đường đi sơ cấp duy nhất thì G là một cây.

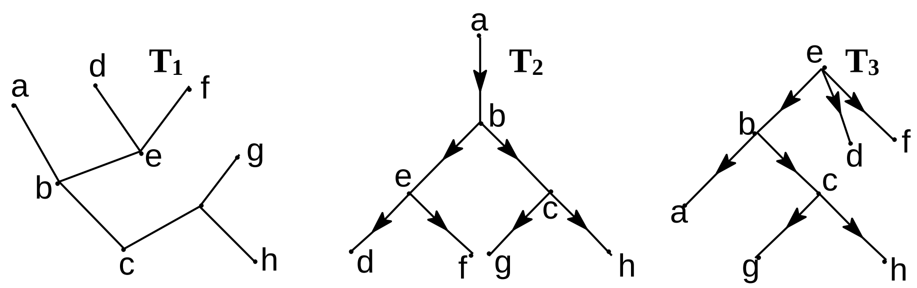
3. Cây có gốc

3.1 Định nghĩa

Trong một cây, nếu ta chọn một đỉnh đặc biệt gọi là gốc của cây và định hướng các cạnh trên cây từ gốc đi ra thì ta được một đồ thị có hướng gọi là cây có gốc.

☞ ***<u>Chú ý:</u>*** Cùng một cây, nếu ta chọn gốc khác nhau thì cây có gốc thu được sẽ khác nhau.

Ví dụ:



Hình 4. Cây và cây có gốc

+ T_1 là một cây,
+ T_2 là cây có gốc a,
+ T_3 là cây có gốc e,
+ T_2 và T_3 là các cây có gốc được sinh ra từ cây T_1.

3.2 Một số khái niệm

Cho T là một cây có gốc, v là một đỉnh khác gốc của T.

- Cha của v là đỉnh u $\in$ T sao cho có một cạnh có hướng duy nhất từ u $\rightarrow$ v. Khi đó, u được gọi là cha của v; v là con của u.
- Các đỉnh có cùng cha được gọi là anh em.
- Tổ tiên của một đỉnh khác gốc là các đỉnh trên đường đi từ gốc đến đỉnh đó.
- Con cháu của v là các đỉnh có v là tổ tiên.
- Các đỉnh của cây không có con được gọi là lá.
- Các đỉnh có con được gọi là đỉnh trong.
- Trong một cây, cho a là một đỉnh. Cây con với gốc a là đồ thì con của cây đang xét, bao gồm a và các con cháu của nó cùng tất cả các cạnh liên thuộc với các con cháu của a.
- Mức của một đỉnh v trong một cây có gốc T là khoảng cách từ gốc đến v.
- Mức lớn nhất của một đỉnh bất kỳ trong cây gọi là chiều cao của cây.

4. Định lý Daisy Chain

Giả sử T là một đồ thị có n đỉnh. Khi đó, 6 mệnh đề sau là tương đương:

(1): T là một cây,

(2): T không có chu trình và có n - 1 cạnh.

(3): T là một đồ thị liên thông và nếu hủy bất kỳ một cạnh nào của nó cũng làm mất tính liên thông.

(4): Giữa 2 đỉnh bất kỳ của T , luôn tồn tại một đường đi sơ cấp duy nhất nối 2 đỉnh này.

(5): T không có chu trình và nếu thêm một cạnh mới nối 2 đỉnh bất kỳ của T thì sẽ tạo ra một chu trình.

(6): T liên thông và có n - 1 cạnh.

Chứng minh

Ta sẽ chứng minh định lý này theo quy trình vòng tròn như sau: $(1)\Rightarrow(2)\Rightarrow(3)\Rightarrow(4)\Rightarrow(5)\Rightarrow(6)\Rightarrow(1)$

➢ $(1)\Rightarrow(2)$**:** Vì T là một cây nên T là đồ thị liên thông. Hơn nữa, ta có thể xem T là một cây có gốc. Khi đó, mọi đỉnh trên cây đều có bậc vào là 1 ngoại trừ đỉnh gốc có bậc vào là 0. T có n đỉnh nghĩa là có n - 1 đỉnh khác gốc. Vậy T liên thông và có n -1 cạnh.

➢ $(2)\Rightarrow(3)$: Vì T không có chu trình nên mỗi thành phần liên thông của T phải là một cây. Giả sử T có $k \geq 1$ thành phần liên thông. Gọi n_i là số đỉnh thuộc thành phần liên thông thứ i, $i = 1..k$, theo mệnh đề (2) ta có số cạnh của thành phần liên thông thứ i là n_i-1.

Theo giả thuyết ta có:

$$\sum_{i=1}^{k} n_i - 1 = n - k = n - 1$$

Đẳng thức này chỉ xảy ra khi k =1, nghĩa là T có 1 thành phần liên thông nghĩa là T liên thông.

Bây giờ, nếu hủy đi một cạnh của T, ta nhận được một đồ thị T' có n - 2 cạnh và T' không có chu trình vậy T' cũng là một cây: vô lý.

➢ $(3)\Rightarrow(4)$: Gọi v và w là 2 đỉnh bất kỳ của T. Vì T liên thông nền tồn tại một đường đi nối chúng. Giả sử có 2 đường khác nhau cùng nối v với w, khi đó nếu ta hủy đi một cạnh bất kỳ thuộc trên đường thứ nhất nhưng không thuộc đường thứ hai thì vẫn không làm mất tính liên thông của T. Điều này mâu thuẩn với tính chất (3).

➢ $(4)\Rightarrow(5)$: Nếu T có một chu trình nối 2 đỉnh phân biệt v và w, khi thêm vào một cạnh mới nối 2 đỉnh này thì rõ ràng có 2 đường đi khác nhau cùng nối 2 đỉnh v và w. Mâu thuẫn với (4). Vậy T không có chu trình.

Bây giờ, thêm một cạnh nối 2 đỉnh v và w của T. Theo giả thuyết trong T có một đường đi trong T (không chứa cạnh mới) nối v và w. Rõ ràng, đường này thêm cạnh mới sẽ tạo thành chu trình.

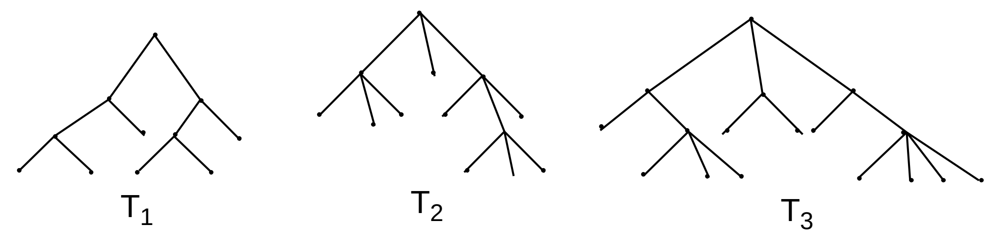
➢ $(5)\Rightarrow(6)$: Xét hai đỉnh bất kỳ v và w của T. Thêm vào T một cạnh mới nối 2 đỉnh này sẽ tạo thành chu trình (giả thuyết). Hủy cạnh mới này khỏi chu trình, ta được một đường trong T nối 2 đỉnh v và w. Vậy T liên thông và không có chu trình. Theo giả thuyết T có n đỉnh nên T có n - 1 cạnh.

➢ $(6) \Rightarrow (1)$: Giả sử T có chu trình và có n - 1 cạnh. Hủy một cạnh khỏi chu trình này vẫn không làm mất tính liên thông của T. Nếu đồ thị nhận được vẫn còn chu trình, ta lại hủy bỏ một cạnh trong chu trình mới,cứ thế tiếp tục cho đến khi nhận được một đồ thị liên thông không có chu trình. Đồ thị này là một cây có n đỉnh nhưng có ít hơn n - 1 cạnh (mâu thuẫn). Vậy T không có chu trình, nghĩa là T là một cây.

5. Cây m-phân

Cây có gốc được gọi là cây m-phân nếu tất cả các đỉnh trong của nó không có hơn m con. Cây được gọi là m-phân đầy đủ nếu mọi đỉnh trong của nó có đúng m con. Cây 2-phân được gọi là cây nhị phân.

Ví dụ:



Hình 5. Cây m-phân

+ T_1: cây nhị phân đầy đủ

+ T_2: cây tam phân đầy đủ

+ T_3: cây tứ phân (không đầy đủ).

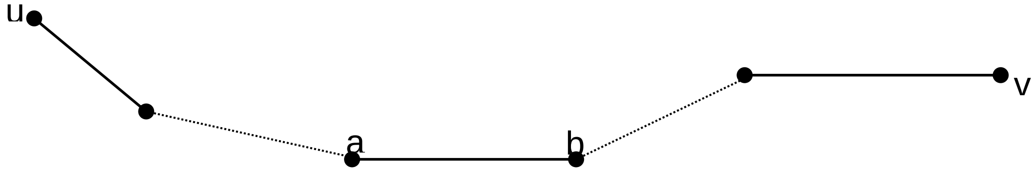
6. Một số tính chất của cây

6.1 Tính chất 1

Nếu T là một cây có n đỉnh ($n \geq 2$) thì T phải có ít nhất 2 đỉnh treo.

Chứng minh

Giả sử a và b là một cạnh của T. Gọi α là đường sơ cấp dài nhất trên T chứa cạnh ab. Chẳng hạn:



Ta thấy u và v phải là 2 đỉnh treo. Thật vậy, nếu u hoặc v không là đỉnh treo thì nó phải là đầu múc của 1 cạnh xu (hoặc vy) nào đó. Khi đó, sẽ tồn tại đường α' chứa cạnh ab có độ dài lớn hơn α (Vô lý). Vậy u và v phải là 2 đỉnh treo.

6.2 Tính chất 2

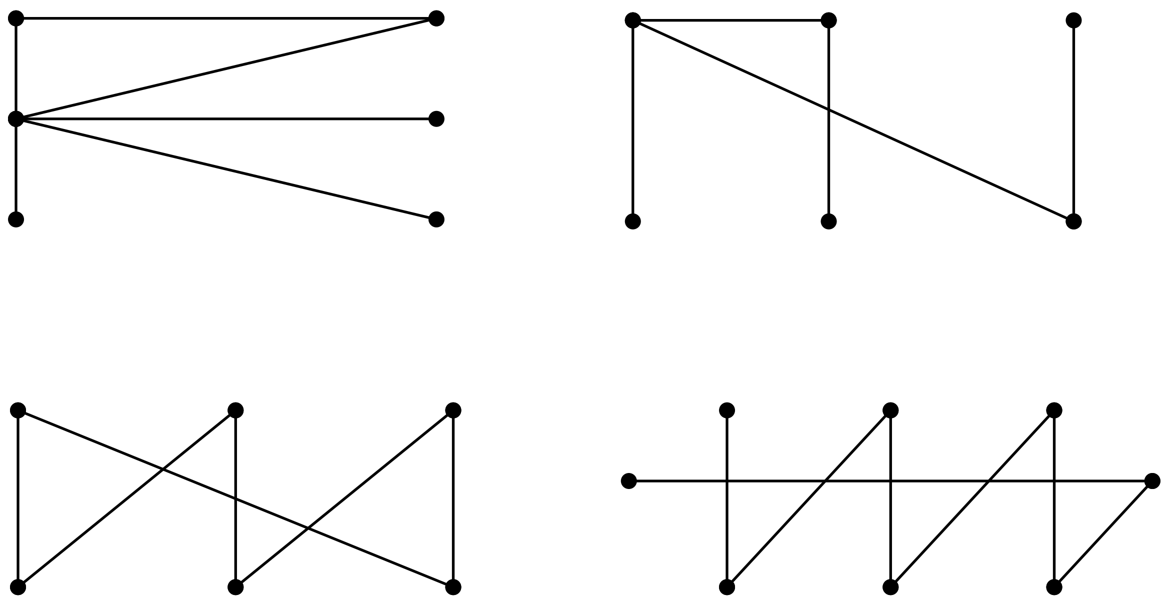
Cây m-phân đầy đủ với i đỉnh trong sẽ có tất cả n = m.i + 1 đỉnh.

Chứng minh

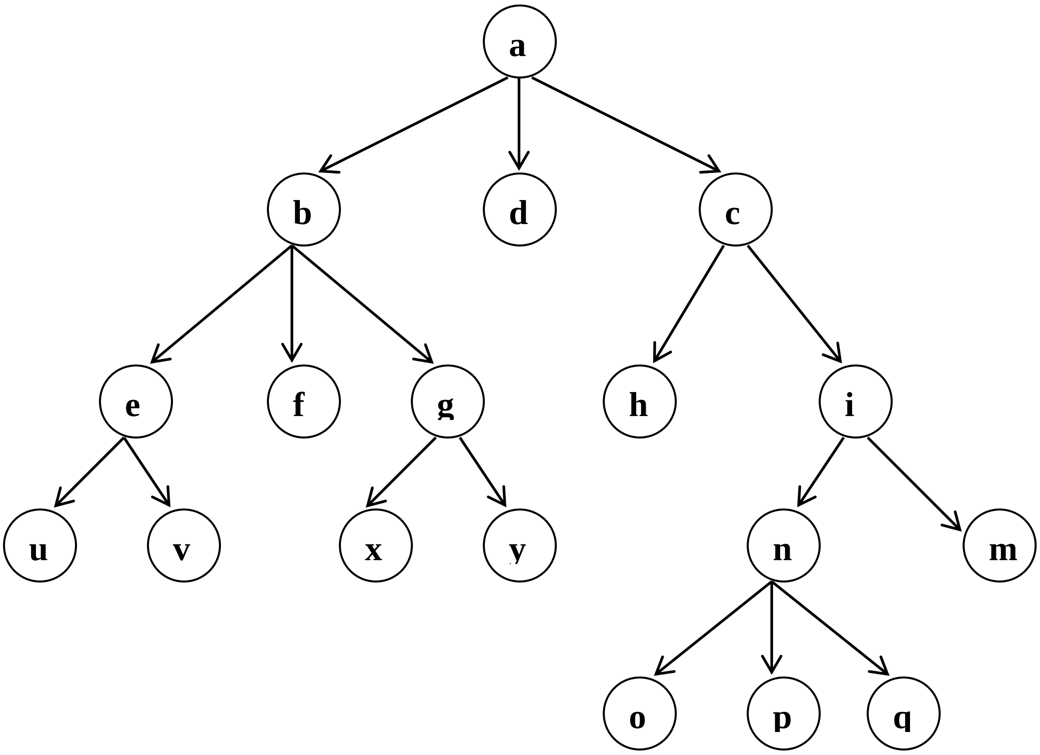
Mỗi đỉnh từ gốc là con của một đỉnh trong. Mỗi đỉnh trong có m con, mà có i đỉnh trong ⇒ có m.i đỉnh khác gốc ⇒ cây có tất cả (m.i + 1) đỉnh.

❖ **Bài tập củng cố:**

Bài 1. Đồ thị nào trong các đồ thị dưới đây là cây?



Bài 2. Cho cây có gốc G=(V,E)



Trong cây trên:

a. Đỉnh nào là gốc
b. Đỉnh nào là đỉnh trong
c. Đỉnh nào là lá
d. Chiều cao của G bằng bao nhiêu
e. Mức của các đỉnh u, m, p là bao nhiêu
f. Cây G được gọi là cây 3-phân có đúng không? Tại sao?
g. Muốn nhận được cây nhị phân đầy đủ từ cây G ở trên cần bỏ đi những đỉnh nào
h. Hãy chỉ ra các cây con của đỉnh a

BÀI 2
CÂY NHỊ PHÂN VÀ PHÉP DUYỆT CÂY

❖ **Mục tiêu học tập:** Sau khi học xong bài này, người học có thể:
- Trình bày các khái niệm cơ bản và tính chất của cây nhị phân
- Duyệt cây nhị phân

1. Phép duyệt cây

Duyệt cây là đưa ra một danh sách liệt kê tất cả các đỉnh của cây, mỗi đỉnh một lần. Có 3 phép duyệt cây thường dùng là duyệt tiền tự (PreOrder), duyệt trung tự (InOrder) và duyệt hậu tự (PostOrder). Cả ba phương pháp duyệt trên đều được định nghĩa đệ quy đối với cây nhị phân (mỗi nút có tối đa 2 con lần lượt được gọi là con trái và con phải của nút).

- ***Định nghĩa phép duyệt tiền tự (PreOrder)***

  Duyệt nút gốc,
  Duyệt con trái theo phương pháp duyệt tiền tự,
  Duyệt con phải theo phương pháp duyệt tiền tự,

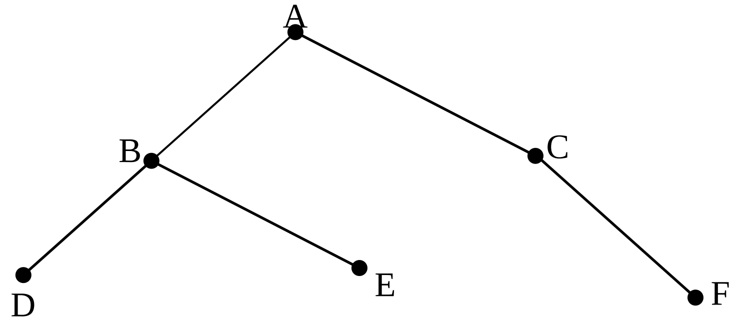
- ***Định nghĩa phép duyệt trung tự (InOrder)***

  Duyệt con trái theo phương pháp duyệt trung tự,
  Duyệt nút gốc,
  Duyệt con phải theo phương pháp duyệt trung tự,

- ***Định nghĩa phép duyệt hậu tự (PostOrder)***

  Duyệt con trái theo phương pháp duyệt hậu tự,
  Duyệt con phải theo phương pháp duyệt hậu tự,
  Duyệt nút gốc.

Ví dụ: Duyệt cây nhị phân sau theo tiền tự, trung tự, hậu tự



Hình 1. Cây nhị phân

➢ Danh sách duyệt tiền tự: A B D E C F

➢ Danh sách duyệt trung tự: D B E A C F

➢ Danh sách duyệt hậu tự: D E B F C A

2. Ký pháp nghịch đảo Ba Lan (Reverse Polish Notation - RPN)

2.1 Cây biểu thức số học

Cây biểu thức số học là một cây nhị phân, trong đó:

+ Mỗi nút trong biểu diễn cho một toán tử 2 ngôi θ

+ Mỗi nút lá biểu diễn cho một toán hạng của biểu thức.

Nếu nút trong biểu diễn cho toán tử 2 ngôi θ và có 2 con:

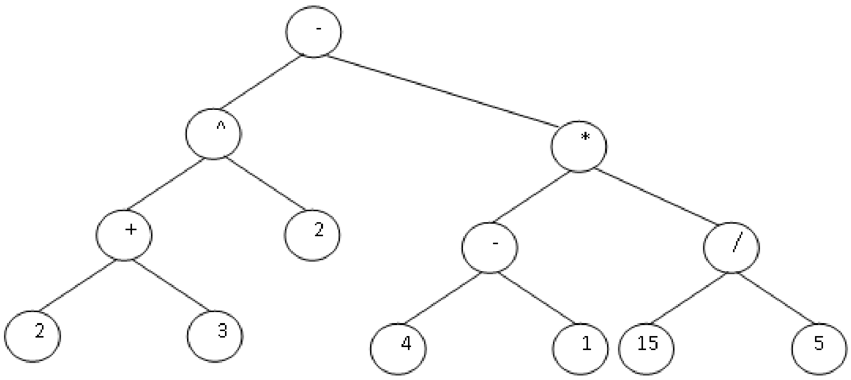
+ Con trái biểu diễn cho biểu thức E_1,

+ Con phải biểu diễn cho biểu thức E_2,

Khi đó nút trong này biểu diễn cho biểu thức số học $E_1\theta E_2$

Ví dụ: Xét biểu thức số học $E = (2+3)\string^2 - (4-1)*(15/5)$

Được biểu diễn bằng cây biểu thức số học sau:



Hình 2. Cây biểu thức số học

Đối với cây biểu thức số học nếu duyệt tiền tự ta được biểu thức tiền tố, duyệt trung tự ta được biểu thức trung tố và duyệt hậu tự ta được biểu thức hậu tố của biểu thức số học ban đầu.

+ Biểu thức tiền tố: - ^ + 2 3 2 * - 4 1 / 15 5

+ Biểu thức trung tố: 2 + 3 ^ 2 - 4 - 1 * 15 / 5

+ Biểu thức hậu tố: 2 3 + 2 ^ 4 1 - 15 5 / * -

2.2 Ký pháp nghịch đảo Ba Lan (RPN)

Trong máy tính để tính giá trị của biểu thức E người ta sử dụng biểu thức dạng hậu tố gọi là ký pháp nghịch đảo Ba Lan (Reverse Polish Notation - RPN) của biểu thức E.

Trong máy tính giá trị của biểu thức E được tính bằng cách lưu các số của biểu thức E được biểu diễn theo ký pháp RPN vào trong một bộ nhớ Stack (Stack là một cấu trúc lưu trữ có tính chất là phần tử nào được lưu trữ trước thì được lấy ra sau).

Mỗi khi một toán tử được đưa vào thì máy sẽ lấy 2 toán hạng ở đầu Stack ra, áp dụng phép toán được biểu diễn bởi toán tử cho 2 toán hạng này rồi lại đẩy kết quả vào stack, cứ như thế cho đến khi đạt được kết quả sau cùng.

Quá trình tính giá trị cho biểu thức E ở ví dụ trên được mô tả bởi hình vẽ:

Biểu thức nhập dưới dạng ký pháp PRN:

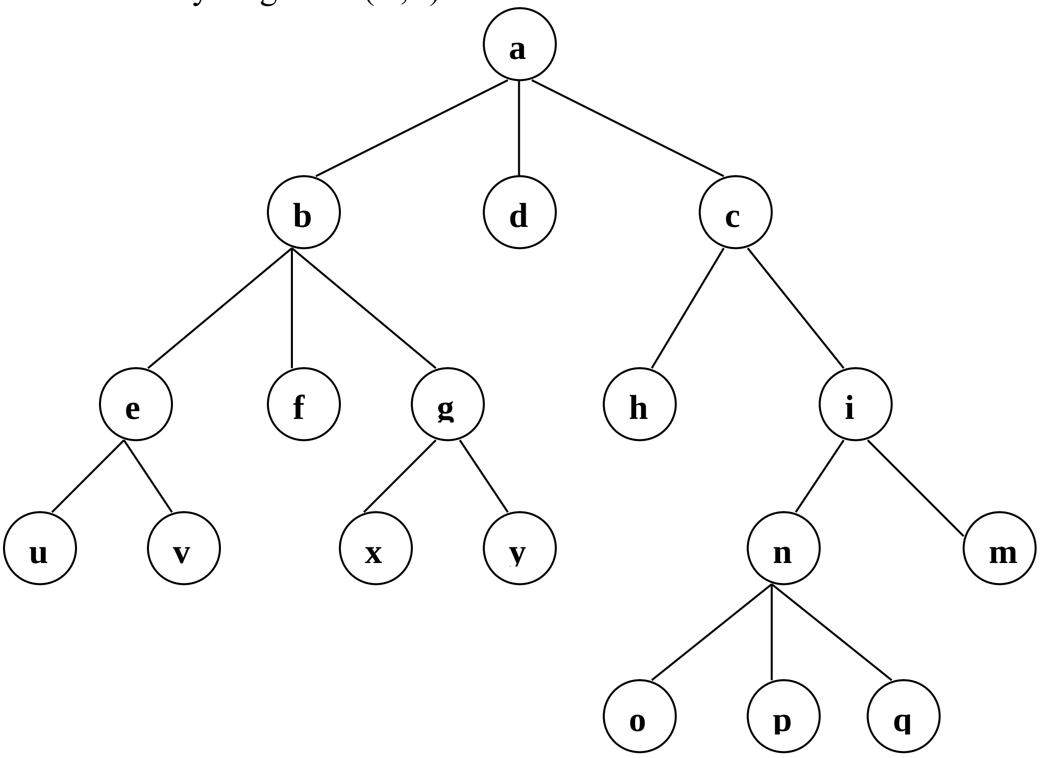
E = 2 3 + 2 ^ 4 1 - 15 5 / * -

Quá trình lưu trữ của cấu trúc Stack như sau:

16
-
9
*
3
/
5
15
3
-
1
4
25
^
2
5
+
3
2

❖ **Bài tập củng cố:**

Bài 1. Cho cây có gốc G=(V,E)



Xác định thứ tự các đỉnh của cây trên nếu ta duyệt nó theo kiểu:

a. Tiền tự
b. Trung tự
c. Hậu tự

Bài 2. Tìm giá trị của biểu thức tiền tố

+ - * 2 3 5 / ↑ 2 3 4

Bài 3. Tìm giá trị của biểu thức hậu tố

7 2 3 * - 4 ↑ 9 3 / +

Bài 4. Tìm cây có gốc biểu diễn biểu thức

((x+y)↑2) + ((x – 4)/3)

BÀI 3
BÀI TOÁN CÂY KHUNG NHỎ NHẤT

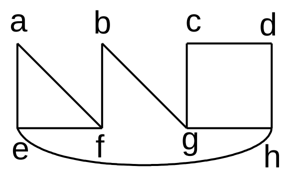
❖ **Mục tiêu học tập:** Sau khi học xong bài này, người học có thể:
- Xác định bài toán quy về bài toán cây khung
- Tìm cây khung nhỏ nhất trên đồ thị có trọng số

1. Cây khung

Cho G là một đơn đồ thị. Một cây được gọi là cây khung của G nếu nó là một đồ thị con của G và chứa tất cả các đỉnh của G.

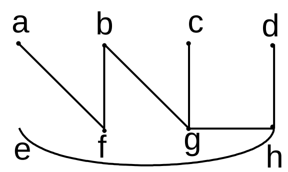
Ví dụ

Cho đơn đồ thị G sau:



Hình 1. Đơn đồ thị G

Một cây khung tạo ra từ G bằng cách xóa đi các cạnh tạo ra chu trình đơn: ae, ef và cd là:



Hình 2. Một cây khung của G

2. Định lý

Một đơn đồ thị là liên thông khi và chỉ khi nó có cây khung.

Chứng minh

➢ Giả sử G có cây khung T ⇒ T chứa tất cả các đỉnh của G. Vì T là cây nên luôn có đường đi giữa 2 đỉnh bất kỳ của T. Mà T là đồ thị con chứa tất cả các đỉnh của G ⇒ luôn tồn tại đường đi giữa 2 đỉnh bất kỳ trong G. Vậy G liên thông.

➢ Giả sử G là liên thông. Nếu G không phải là một cây ⇒ G có chu trình đơn. Xóa đi một cạnh trong chu trình đơn này. Khi đó, đồ thị nhận được có số cạnh ít hơn G nhưng số đỉnh vẫn bằng số đỉnh của G và vẫn liên thông. Nếu đồ thị con này không là cây thì nó còn chứa chu trình đơn. Lặp lại quá trình trên, ta lại xóa đi một cạnh của chu trình đơn này. Quá trình cứ tiếp tục cho đến khi không còn chu trình đơn nào. Điều này luôn thực hiện được vì ta chỉ xét các đồ thị hữu hạn. Khi quá trình kết thúc, ta nhận được một cây khung của G.

3. Cây khung bé nhất

Cây khung bé nhất trong một đồ thị liên thông, có trọng số là một cây khung có tổng trọng số trên các cạnh của nó là nhỏ nhất.

Như vậy bài toán được đặt ra ở đây là trong số các cây khung của đồ thị ban đầu ta phải tìm ra cây khung có tổng trọng số là nhỏ nhất, Ta có 2 thuật toán tìm cây khung nhỏ nhất phổ biến là thuật toán Prim và thuật toán Kruskal

3.1 Thuật toán Prim

Thuật toán Prim được giới thiệu lần đầu tiên vào năm 1957. Thuật toán Prim được dùng để tìm cây khung nhỏ nhất trong một đồ thị liên thông có trọng số.

> Thuật toán Prim bắt đầu bằng việc chọn một cạnh bất kỳ có trọng số nhỏ nhất, đặt nó vào cây khung. Sau đó, lần lượt ghép vào cây các cạnh có trọng số nhỏ nhất liên thuộc với một đỉnh của cây và không tạo ra chu trình trong cây. Thuật toán dừng lại khi (n – 1) cạnh được ghép vào cây.

Chú ý: Có nhiều hơn một cây khung nhỏ nhất ứng với một đồ thị liên thông và có trọng số.

Thuật toán Prim dưới dạng giả mã:

Procedure Prim (G: đồ thị liên thông có trọng số với n đỉnh).

T: = cạnh có trọng số nhỏ nhất.

for i = 1 to n – 2.

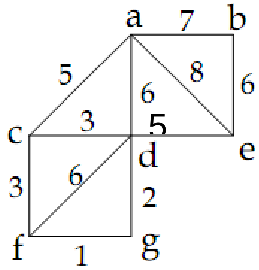
begin

e:= cạnh có trọng số tối thiểu liên thuộc với một đỉnh trong T và không tạo ra chu trình trong T nếu ghép nó vào T.

T: = T với e được ghép vào

end {T là cây khung nhỏ nhất của G}.

Ví dụ: Dùng thuật toán Prim để tìm cây khung nhỏ nhất trong đơn đồ thị có trọng số sau:



3.2 Thuật toán Kruskal

Để tìm cây khung nhỏ nhất của một đơn đồ thị liên thông có trọng số. Ta còn có thể dùng thuật toán Kruskal.

Để thực hiện thuật toán Kruskal, ta xuất phát từ việc chọn cạnh có trọng số nhỏ nhất của đồ thị. Sau đó lần lượt ghép thêm vào các cạnh có trọng số tối thiểu và không tạo thành chu trình với các cạnh đã được chọn. Thuật toán dừng lại khi (n – 1) cạnh được chọn.

Thuật toán Kruskal dưới dạng giả mã:

Procedure Kruskal (G: đồ thị n đỉnh, liên thông có trọng số).

T: = đồ thị rỗng

for i: = 1 to n – 1

begin

e: = một cạnh bất kỳ của G với trọng số nhỏ nhất và không tạo ra chu trình trong T, khi ghép nó vài T.

T: = T với cạnh e đã được ghép vào.

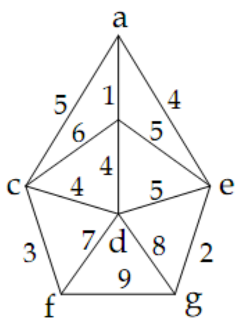
end {T là cây khung nhỏ nhất}.

Sự khác nhau giữa thuật toán Prim và thuật toán Kruskal.

Trong thuật toán Prim, ta chọn các cạnh có trọng số nhỏ nhất, liên thông với các đỉnh đã thuộc cây và không tạo ra chu trình.

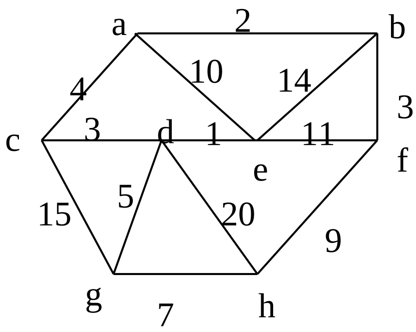
Trong thuật toán Kruskal, ta chọn các cạnh có trọng số tối thiểu mà không nhất thiết phải liên thuộc với các đỉnh của cây và không tạo ra chu trình.

Ví dụ: Dùng thuật toán Kruskal, tìm cây bao trùm ngắn nhất của đồ thị sau:



❖ **Bài tập củng cố:**

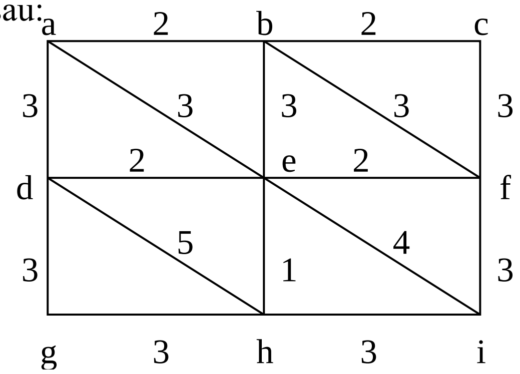
Bài 1. Tìm cây bao trùm ngắn nhất của đồ thị G sau bằng thuật toán Kruskal và thuật toán Prim (bắt đầu từ đỉnh d).



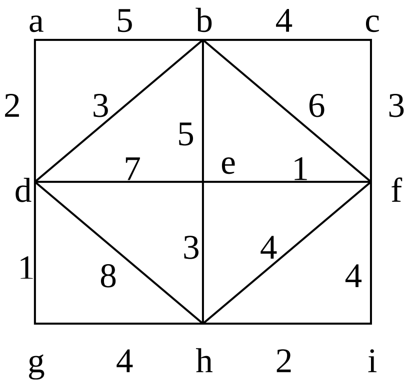
Bài 2. Tìm cây bao trùm dài nhất của đồ thị G trong bài 01.

Bài 3. Dùng thuật toán Kruskal và thuật toán Prim tìm cây bao trùm ngắn nhất của các đồ thị sau:

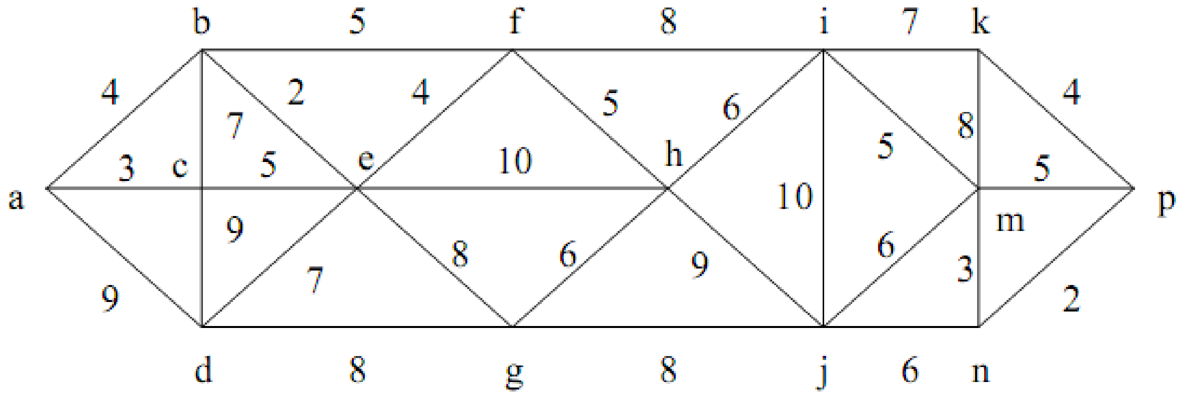
a.

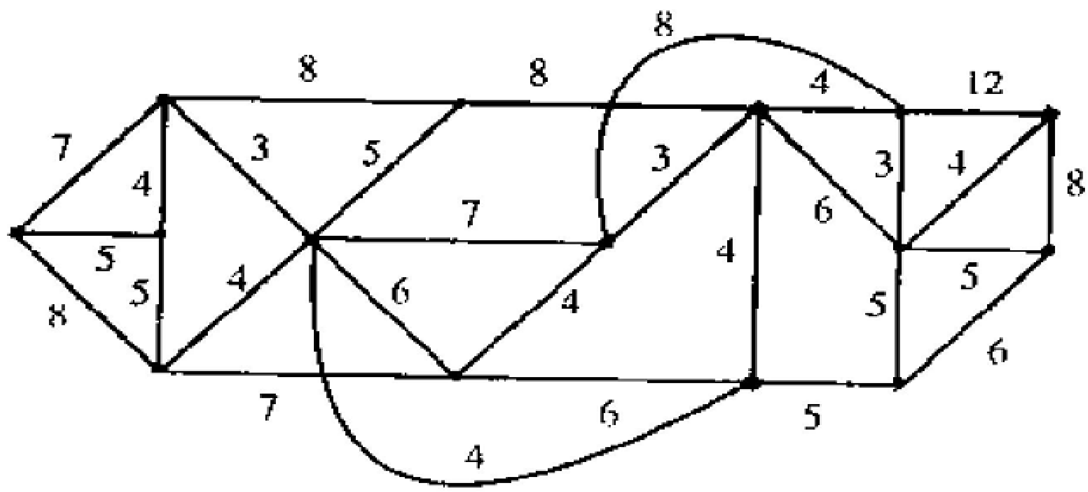


b.



Bài 4. Tìm cây bao trùm nhỏ nhất chứa cạnh km của đồ thị sau bằng 2 phương pháp:




8
8
8
4
12
7
4
3
5
3
3
4
8
7
6
5
5
4
6
4
4
5
8
5
6
7
6
5
4

TÀI LIỆU THAM KHẢO

❖ ***TÀI LIỆU THAM KHẢO ĐỂ BIÊN SOẠN NỘI DUNG MÔN HỌC:***

1. ThS. Bùi Anh Kiệt – ThS. Trương Quốc Bảo, giáo trình Toán rời rạc, tủ sách Đại học Cần Thơ 2004.
2. Phạm Văn Thiều – Đặng Hữu Thịnh, Toán rời rạc ứng dụng trong tin học (bản dịch), NXB Giáo Dục, 2007.
3. Nguyễn Đức Nghĩa – Tô Hiến Thành, Toán rời rạc, NXB ĐHQG Hà Nội, 2009.
4. Đỗ Đức Giáo, Toán rời rạc ứng dụng trong tin học, NXB Giáo Dục, 2009.
5. Vũ Đình Hòa, Định lý và vấn đề về đồ thị hữu hạn, Nhà xuất bản Giáo dục, 2001.
6. Rosen K.H, Discrete Mathematics and its Applications 7th Edition. MCGraw – Hill Book Company, 2011
7. C.L. Liu, Elements of Discrete Mathematics, MCGraw – Hill Book Company, 1985

❖ ***TÀI LIỆU THAM KHẢO ĐỀ NGHỊ CHO HỌC VIÊN:***

1. ThS. Bùi Anh Kiệt – ThS. Trương Quốc Bảo, giáo trình Toán rời rạc, tủ sách Đại học Cần Thơ 2004.
2. Phạm Văn Thiều – Đặng Hữu Thịnh, Toán rời rạc ứng dụng trong tin học (bản dịch), NXB Giáo Dục, 2007.
3. Nguyễn Đức Nghĩa – Tô Hiến Thành, Toán rời rạc, NXB ĐHQG Hà Nội, 2009.
4. Đỗ Đức Giáo, Toán rời rạc ứng dụng trong tin học, NXB Giáo Dục, 2009.
5. Vũ Đình Hòa, Định lý và vấn đề về đồ thị hữu hạn, Nhà xuất bản Giáo dục, 2001.
6. Rosen K.H, Discrete Mathematics and its Applications 7th Edition. MCGraw – Hill Book Company, 2011
7. C.L. Liu, Elements of Discrete Mathematics, MCGraw – Hill Book Company, 1985