Phần 2

LÝ THUYẾT ĐỒ THỊ

Graph Theory

GV: Nguyễn Huy Đức

Bộ môn: Khoa học Máy tính – ĐH Thủy lợi

0903 402 655 ducnghuy@gmail.com

Nội dung

- Chương 1: Các khái niệm cơ bản
- Chương 2: Biểu diễn đồ thị
- Chương 3: Các thuật toán tìm kiếm trên đồ thị
- **Chương 4: Đồ thị Euler và đồ thị Hamilton**
- Chương 5: Cây và cây khung của đồ thị
- Chương 6: Bài toán đường đi ngắn nhất

Chương 4

ĐỒ THỊ EULER VÀ ĐỒ THỊ HAMILTON

Nội dung chương:

4.1 Đồ thị Euler

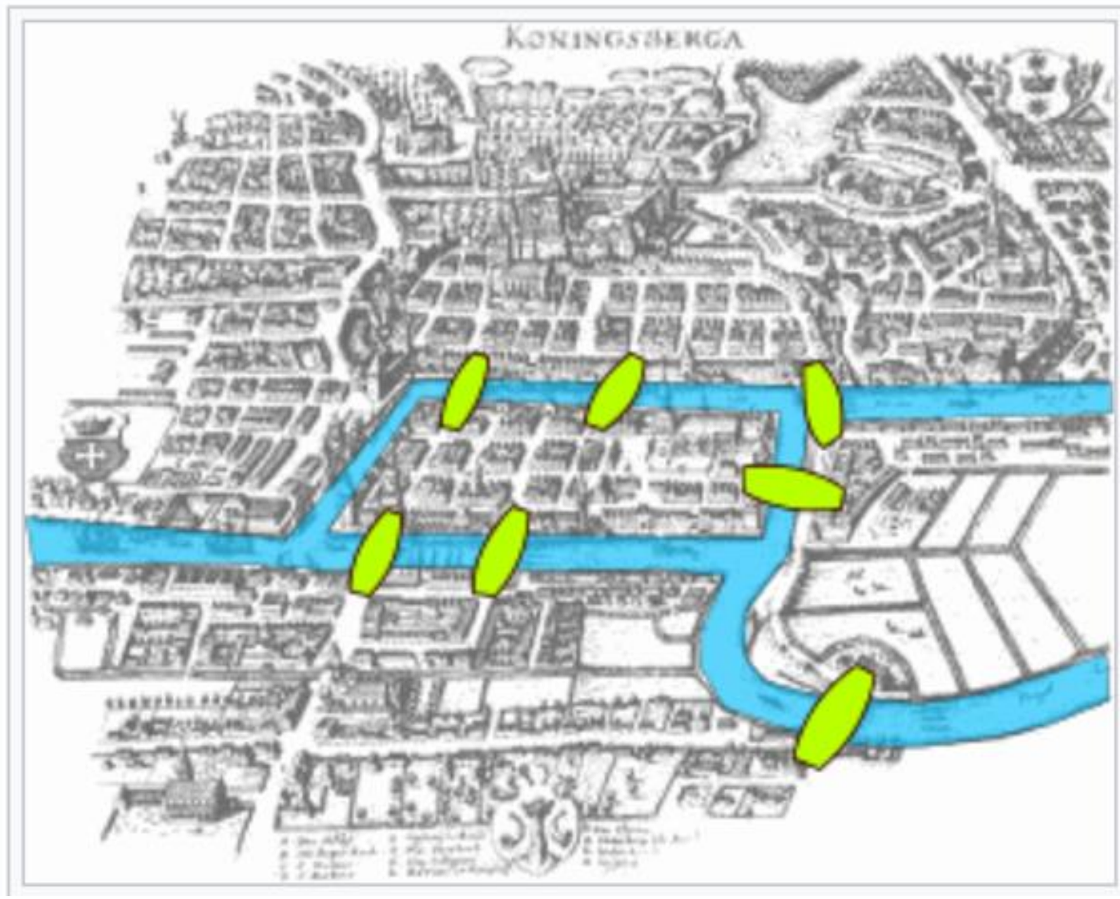
4.2 Đồ thị Hamilton

4.1 ĐỒ THỊ EULER

BÀI TOÁN 7 CÁI CẦU:

Thành phố Konigsberg thuộc Phổ (nay là Kaliningrad thuộc Cộng hoà LB Nga), được chia làm 4 vùng bằng các nhánh sông Pregel. Vào thế kỷ XVIII, người ta đã xây 7 chiếc cầu nối những vùng này với nhau.

Người dân ở đây tự hỏi: Liệu có cách nào xuất phát tại một địa điểm trong thành phố, đi qua 7 chiếc cầu, mỗi chiếc đúng 1 lần rồi quay trở về nơi xuất phát không ?

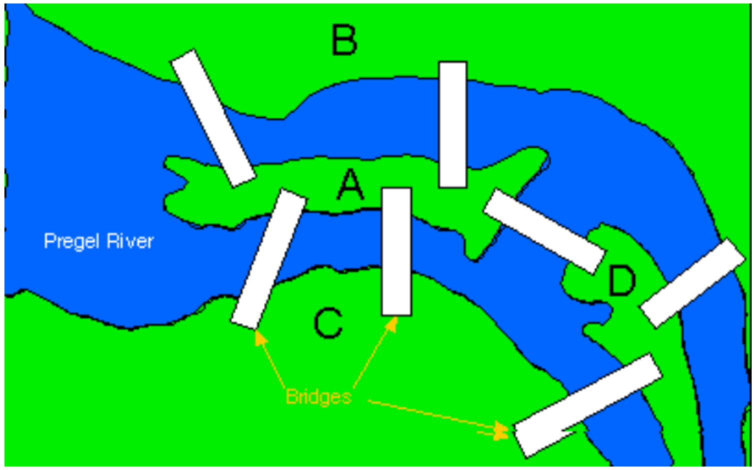


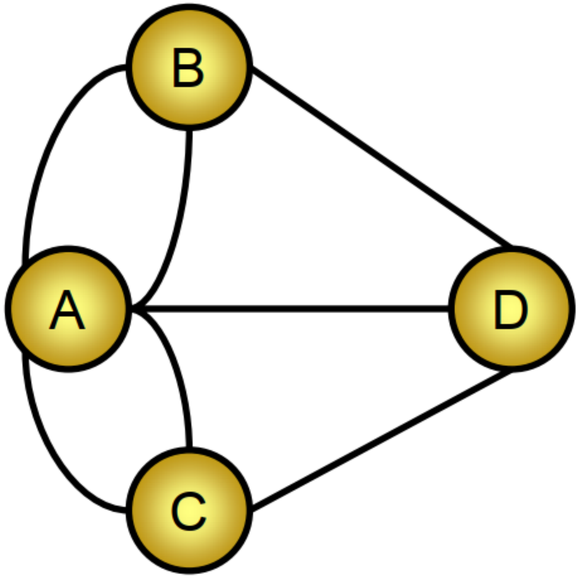
Bản đồ Königsberg thời Euler, mô tả vị trí thực của 7 cây cầu và sông Pregel

(nguồn https://vi.wikipedia.org/)

BÀI TOÁN 7 CÁI CẦU

Nhà toán học Thụy sĩ Leonhard Euler (1707 - 1783) đã giải bài toán này và có thể coi đây là ứng dụng đầu tiên của Lý thuyết đồ thị, ông đã mô hình hoá sơ đồ 7 cái cầu bằng một **đa đồ thị, bốn vùng được biểu diễn bằng 4 đỉnh, các cầu là các cạnh**. Bài toán tìm đường qua 7 cầu, mỗi cầu đúng một lần có thể tổng quát hoá bằng bài toán: **Có tồn tại chu trình đơn trong đa đồ thị chứa tất cả các cạnh ?**





4.1.1 ĐỊNH NGHĨA ĐỒ THỊ EULER

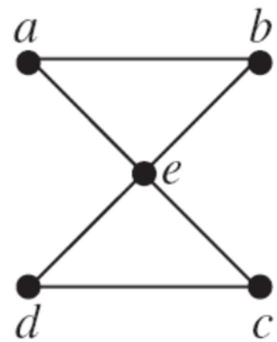
ĐỊNH NGHĨA:

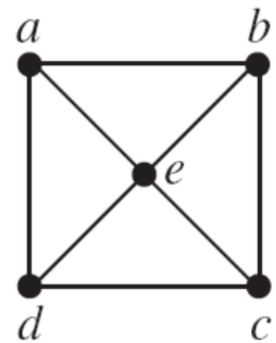
- Chu trình đơn chứa tất cả các cạnh của đồ thị được gọi là chu trình Euler.
- Đường đi đơn chứa tất cả các cạnh của đồ thị được gọi là đường đi Euler.
- Một đồ thị có chu trình Euler được gọi là đồ thị Euler
- Một đồ thị có đường đi Euler được gọi là đồ thị nửa Euler.

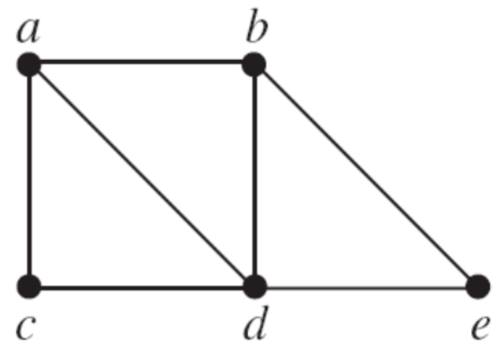
Rõ ràng, mọi đồ thị Euler đều là nửa Euler nhưng điều ngược lại không đúng.

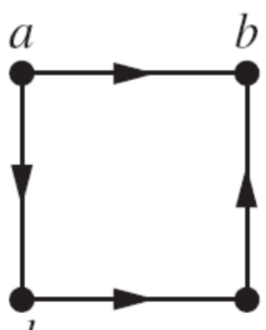
VÍ DỤ ĐỒ THỊ EULER

<u>Ví dụ 1</u>: Đồ thị nào sau đây có chu trình Euler?

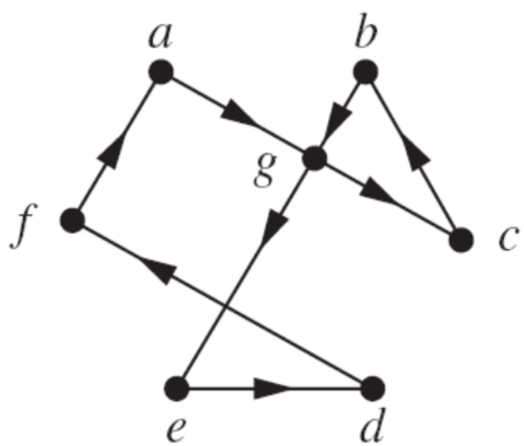




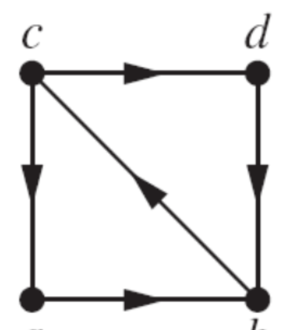




H_1



H_2



H_3

4.1.2 CÁC ĐỊNH LÝ ĐIỀU KIỆN

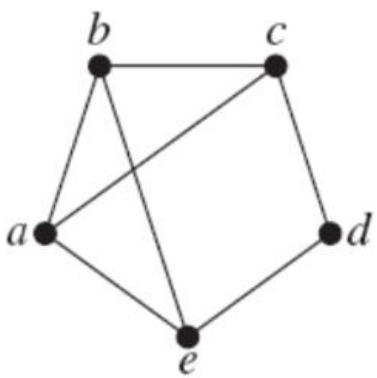
- **Định Lý 1:**

  1. Một đồ thị vô hướng liên thông G = (V, E) có chu trình Euler khi và chỉ khi mọi đỉnh của nó đều có bậc chẵn.

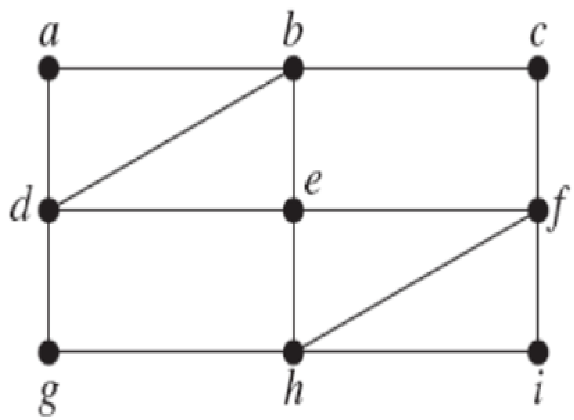
  2. Một đồ thị vô hướng liên thông có đường đi Euler nhưng không có chu trình Euler khi và chỉ khi nó có đúng 2 đỉnh bậc lẻ.

- **Ví dụ 2:** Đồ thị nào sau là đồ thị Euler, đồ thị nửa Euler ?

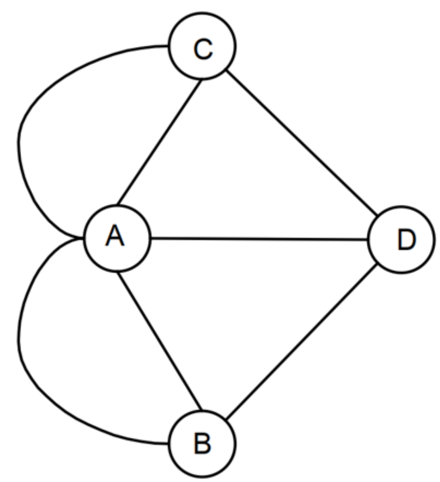
1.



2.



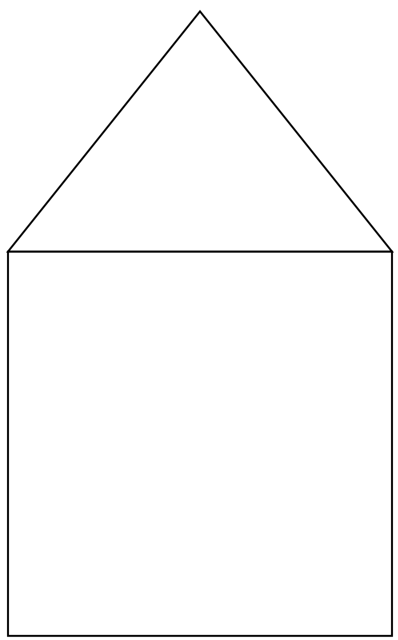
3.

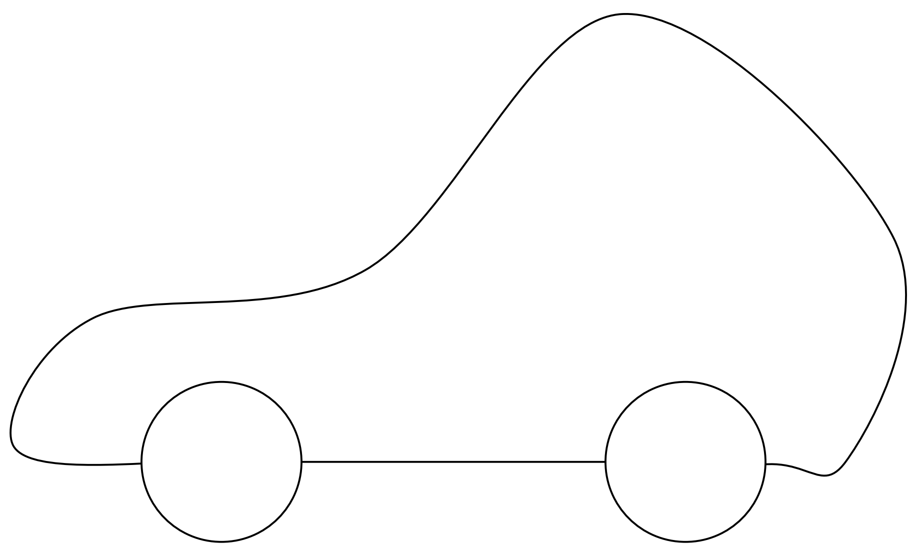


Bài toán 7 cây cầu

Vẽ một nét

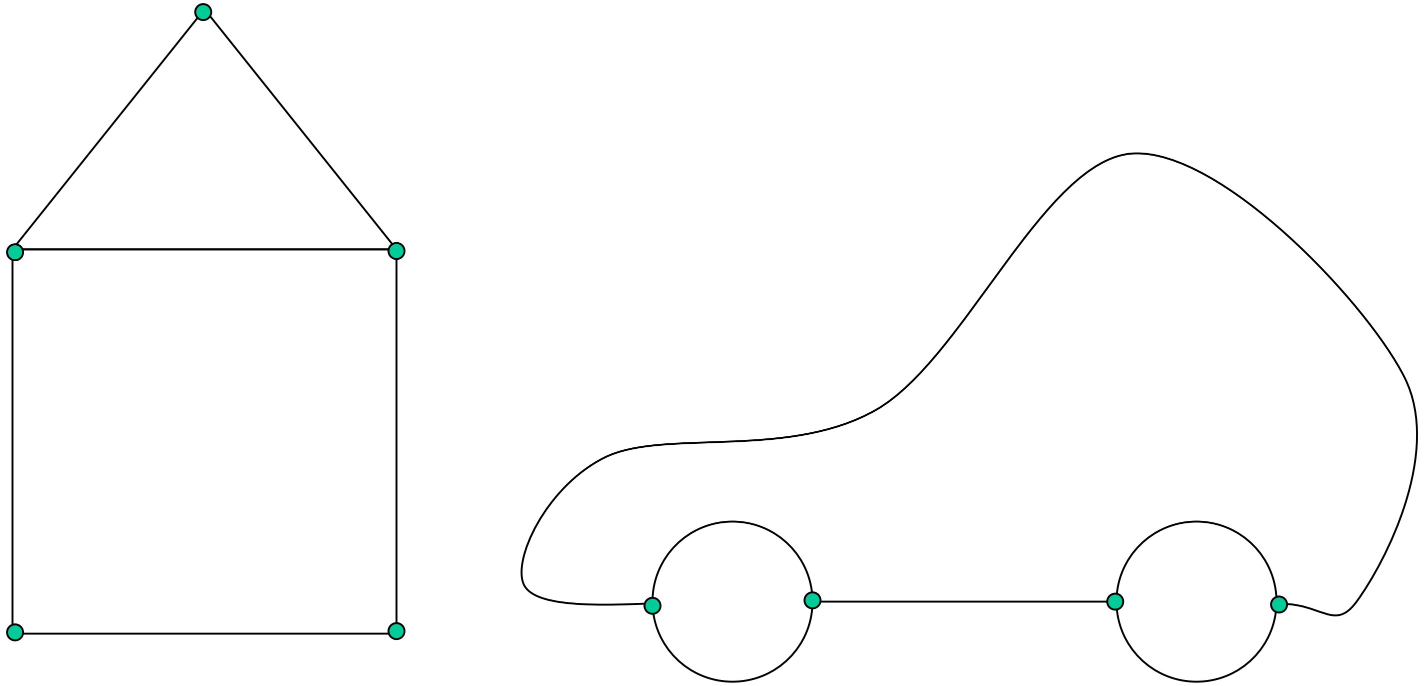
Hình nào trong các hình sau đây có thể tô bởi bút chì mà không được nhấc bút khỏi mặt giấy cũng như không được tô lại bất cứ đoạn nào (vẽ bởi một nét)?





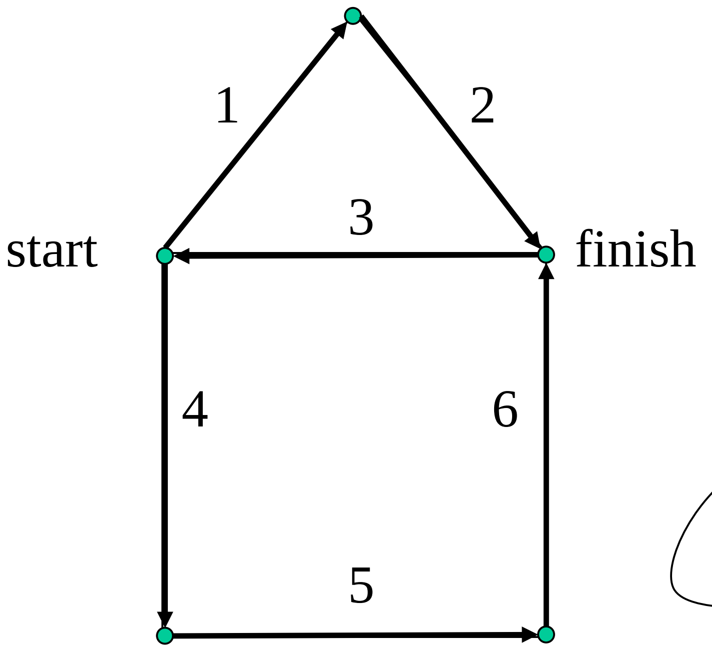
Vẽ một nét

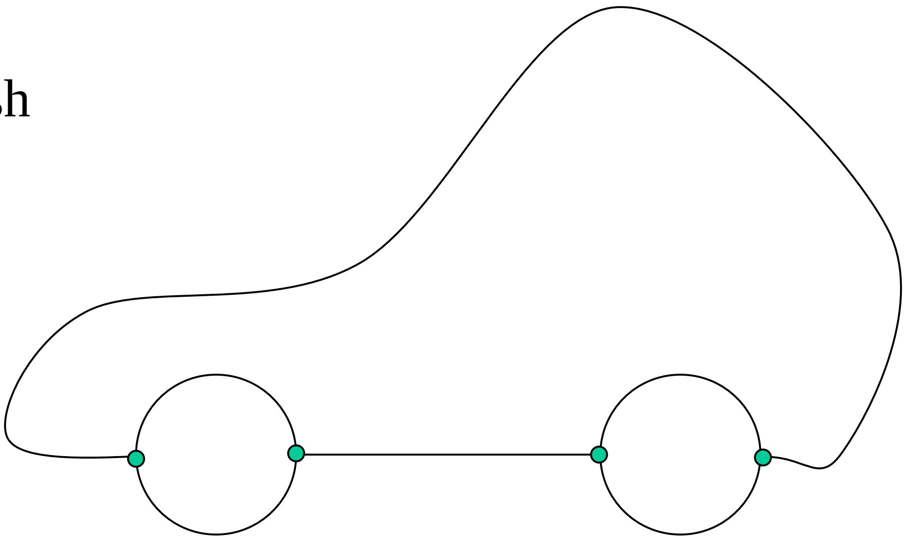
Trong ngôn ngữ đồ thị: Đồ thị nào trong hai đồ thị sau đây có đường đi Euler?



Vẽ một nét – Đường đi Euler

Trả lời: Ngôi nhà vẽ được bởi một nét, còn ôtô thì không thể.





4.1.2 CÁC ĐỊNH LÝ ĐIỀU KIỆN

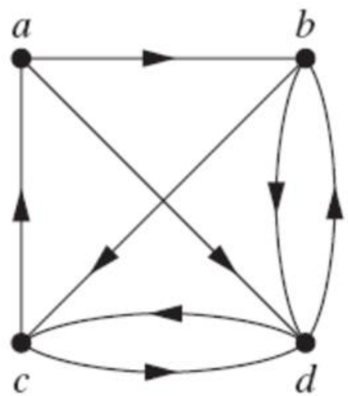
- **Định Lý 2:**

1. Đồ thị có hướng liên thông yếu G=<V, E> là đồ thị Euler khi và chỉ khi tất cả các đỉnh của nó đều có <u>bán bậc ra bằng bán bậc vào</u> (điều này làm cho đồ thị là liên thông mạnh).

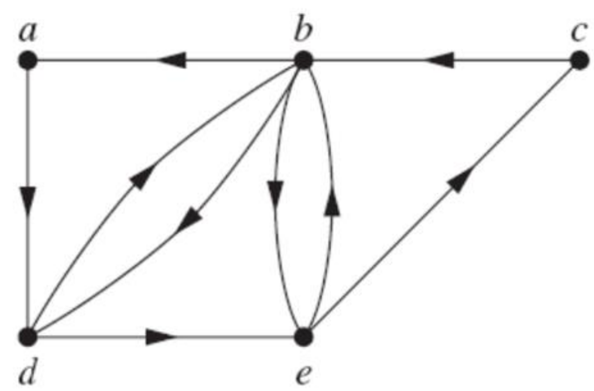
2. Một đồ thị có hướng liên thông yếu G = (V, E) có đường đi Euler nhưng không có chu trình Euler nếu tồn tại đúng hai đỉnh u, v ∈ V sao cho deg+(u) - deg-(u) = deg-(v) - deg+(v) = 1, còn tất cả những đỉnh khác u và v đều có bán bậc ra bằng bán bậc vào.

- Ví dụ 3: Đồ thị nào sau là đồ thị Euler, đồ thị nửa Euler ?

1.



2.



4.1.3 THUẬT TOÁN TÌM CHU TRÌNH EULER

Procedure *Euler* (*G*: đa đồ thị liên thông với tất cả các đỉnh bậc chẵn)

C := chọn 1 chu trình bất kì

H := *G* đã xóa đi cạnh của C

while *H* còn các cạnh

begin

    C' = chu trình trong *H* có đi qua đỉnh trong *C*

    H := *H* xóa đi cạnh của C' và đỉnh treo

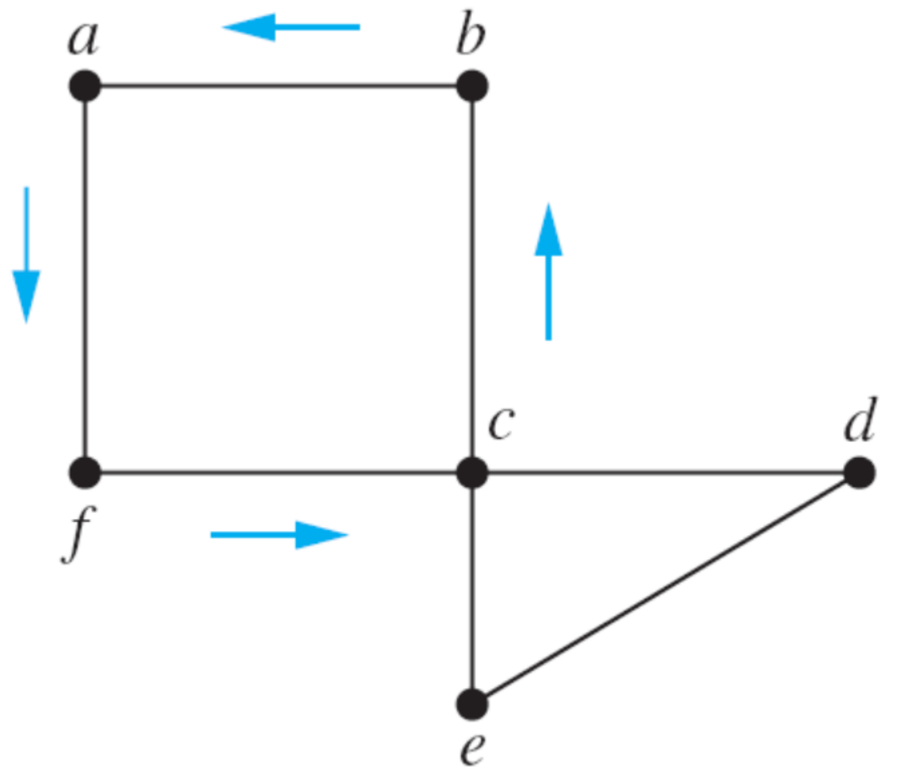
    C := *C* cộng thêm *C'* chèn vào tại một đỉnh thích hợp

end

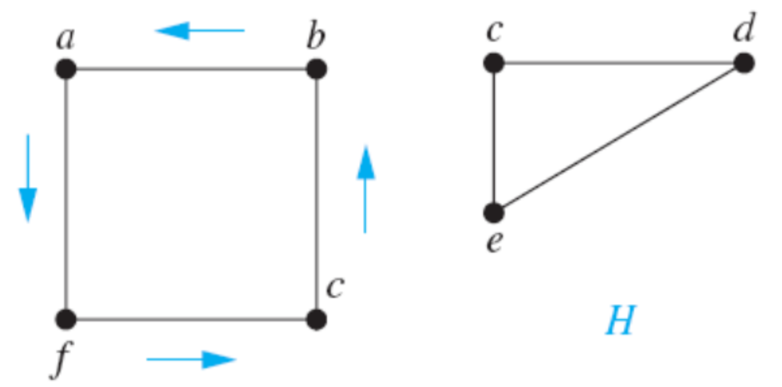
{ *chu trình C* là chu trình Euler}

4.1.3 THUẬT TOÁN TÌM CHU TRÌNH EULER

- **Ví dụ 4**: Tìm chu trình Euler của đồ thị sau ?



4.1.3 THUẬT TOÁN TÌM CHU TRÌNH EULER



<u>Giải</u>:

- Chọn C = chu trình **{a, f, c, b, a}**
- H = các cạnh {c,d}, {c, e}, {e, d}
- C' = chu trình **{c, d, e, c}**
- H = ∅
- C: = C ∪C'={a, f, c, b, a} ∪ { c, d, e, c} = {a, f, **c**, d, e, c, b, a}
- Chu trình Euler là {a,f,c,d,e,c,b,a}

Chương trình trên C++

```
#include <iostream>
using namespace std;
const int nm=102;
int n, m, a[nm][nm];
int ns, st[nm*nm];
int bac[nm];
void nhap()
{  cout<<"So dinh n= "; cin>>n;
   cout<<"So canh m= "; cin>>m;
   int i,u,v;
   //nhap cac canh cua do thi
   for(i=1;i<=m;++i)
   {  cout<<"Dinh dau: "; cin>>u;
      cout<<"Dinh cuoi: "; cin>>v;
      a[u][v]=a[v][u]=1;
      bac[u]++;bac[v]++;   }
}
```

Chương trình trên C++

```
void xuli()
{  int i,j;
   for(i=1;i<=n;++i) if (bac[i]%2) break;
//tim dinh bac le, neu co dinh bac le i thi
//bat dau tu i, nguoc lai bat dau tu dinh 1.
   if (i<=n) st[1]=i; else st[1]=1;  ns=1;
   while (ns)
   {  i=st[ns];
      for(j=1;j<=n;++j)
      {  if (a[i][j])
         {  st[++ns]=j;a[i][j]--;a[j][i]--; break; }  }
      if (i==st[ns])
      {  printf("%d ",i); ns--; }
   }
}
int main()
{  nhap();xuli();  }
```

THUẬT TOÁN FLEURY TÌM CHU TRÌNH EULER

Thuật toán đơn giản cho phép tìm chu trình Euler bằng tay:

THUẬT TOÁN FLEURY:

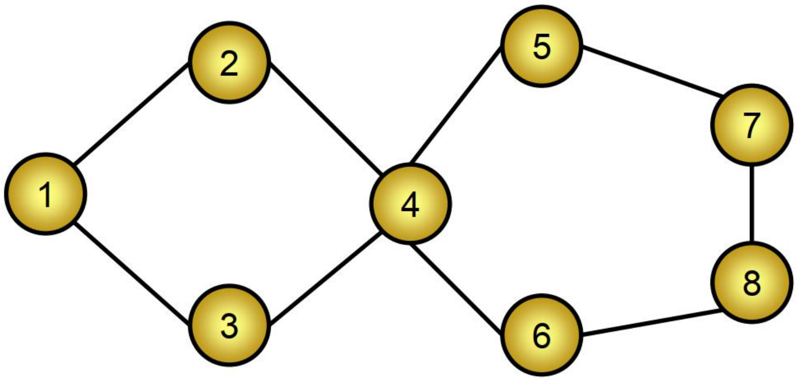
Xuất phát từ 1 đỉnh bất kỳ của đồ thị (G) và tuân theo 2 quy tắc sau:

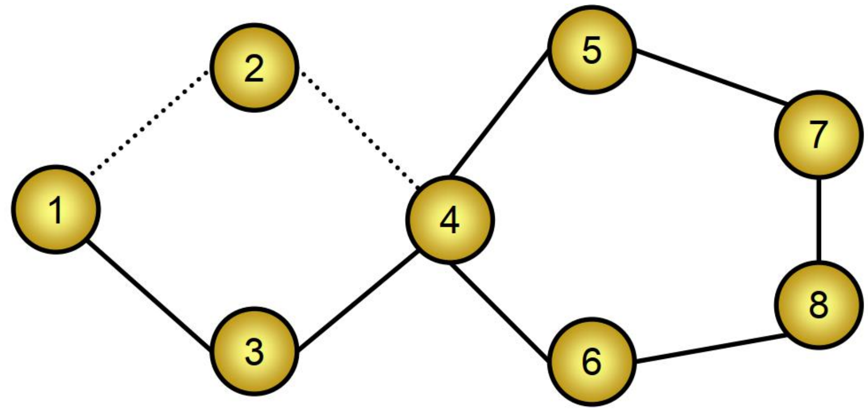
(1) Mỗi khi đi qua một cạnh nào đó thì xóa nó đi, sau đó xóa đỉnh cô lập (nếu có).

(2) Không bao giờ đi qua cầu trừ khi không còn cách đi nào khác.

THUẬT TOÁN FLEURY TÌM CHU TRÌNH EULER

- **Ví dụ 5:** Với đồ thị sau:





Nếu xuất phát từ đỉnh 1, có hai cách đi tiếp: hoặc sang 2 hoặc sang 3, giả sử ta sẽ sang 2 và xoá cạnh (1, 2) vừa đi qua. Từ 2 chỉ có cách duy nhất là sang 4, nên cho dù (2, 4) là cầu ta cũng phải đi sau đó xoá luôn cạnh (2, 4). Đến đây, các cạnh còn lại của đồ thị có thể vẽ như trên bằng nét liền, các cạnh đã bị xoá được vẽ bằng nét đứt.

Bây giờ đang đứng ở đỉnh 4 thì ta có 3 cách đi tiếp: sang 3, sang 5 hoặc sang 6. Vì (4, 3) là cầu nên ta sẽ không đi theo cạnh (4, 3) mà sẽ đi (4, 5) hoặc (4, 6). Nếu đi theo (4, 5) và cứ tiếp tục đi như vậy, ta sẽ được chu trình Euler là (1, 2, 4, 5, 7, 8, 6, 4, 3, 1). Còn đi theo (4, 6) sẽ tìm được chu trình Euler là: (1, 2, 4, 6, 8, 7, 5, 4, 3, 1).

Bài tập

- Cho đơn đồ thị vô hướng G 10 đỉnh, ma trận kề như sau:

	1	2	3	4	5	6	7	8	9	10
1	**0**	**1**	**0**	**0**	**1**	**0**	**0**	**1**	**0**	**1**
2	**1**	**0**	**1**	**1**	**0**	**1**	**0**	**0**	**0**	**0**
3	**0**	**1**	**0**	**1**	**0**	**0**	**0**	**0**	**0**	**0**
4	**0**	**1**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
5	**1**	**0**	**0**	**0**	**0**	**1**	**1**	**1**	**1**	**1**
6	**0**	**1**	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**0**
7	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**1**	**1**	**1**
8	**1**	**0**	**0**	**0**	**1**	**0**	**1**	**0**	**0**	**1**
9	**0**	**0**	**0**	**0**	**1**	**0**	**1**	**0**	**0**	**0**
10	**1**	**0**	**0**	**0**	**1**	**0**	**1**	**1**	**0**	**0**

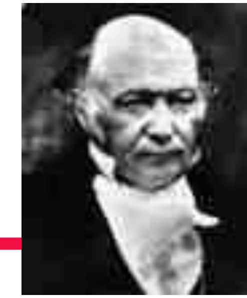
Hãy chứng tỏ G là đồ thị Euler

- Hướng dẫn: + Kiểm tra liên thông
  + Chứng tỏ tất cả các đỉnh đều có bậc là số chẵn.
  + Bậc của đỉnh i bằng tổng của dòng i hoặc cột i trong ma trận kề

4.2 ĐỒ THỊ HAMILTON

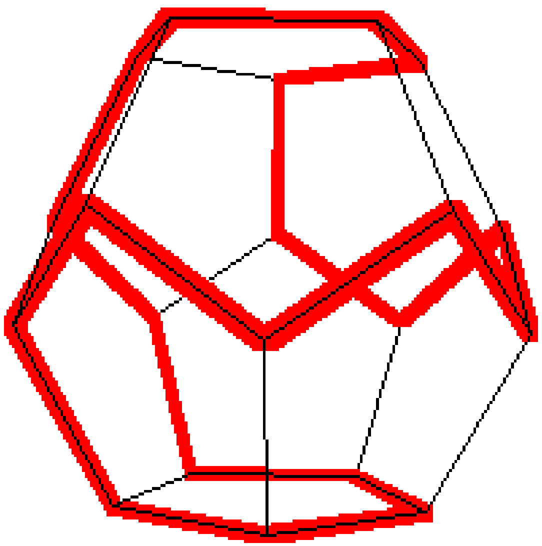
Trò chơi vòng quanh thế giới

(Round-the-World Puzzle)

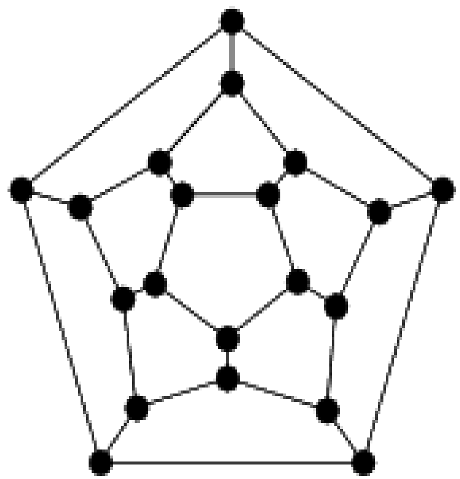


Sir William Rowan Hamilton 1805-1865

- Bạn có thể chỉ ra cách đi qua tất cả các đỉnh của dodecahedron (thập nhị diện) mỗi đỉnh đúng một lần?`



Dodecahedron puzzle



Đồ thị tương ứng

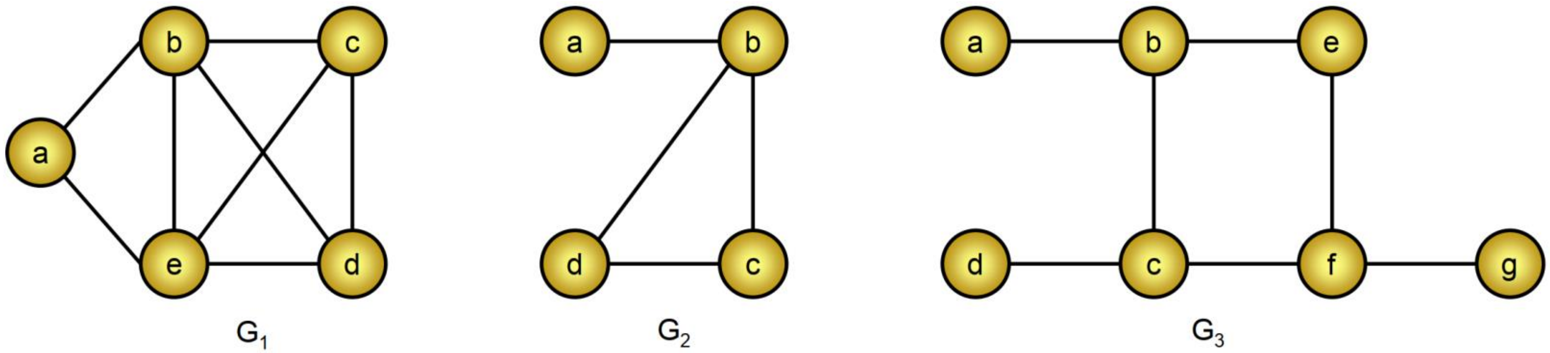
Hộp trò chơi

4.2.1 ĐỊNH NGHĨA ĐỒ THỊ HAMILTON

ĐỊNH NGHĨA:

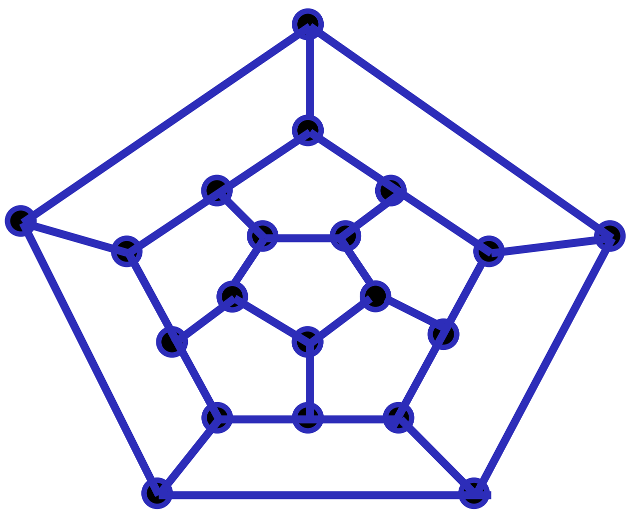
- Đường đi qua tất cả các đỉnh của đồ thị mỗi đỉnh đúng một lần được gọi là đường đi Hamilton.
- Chu trình bắt đầu tại một đỉnh *v* nào đó qua tất cả các đỉnh còn lại mỗi đỉnh đúng một lần sau đó quay trở lại *v* được gọi là chu trình Hamilton.
- Đồ thị có chu trình Hamilton được gọi là đồ thị Hamilton.
- Đồ thị có đường đi Hamilton được gọi là đồ thị nửa Hamilton.

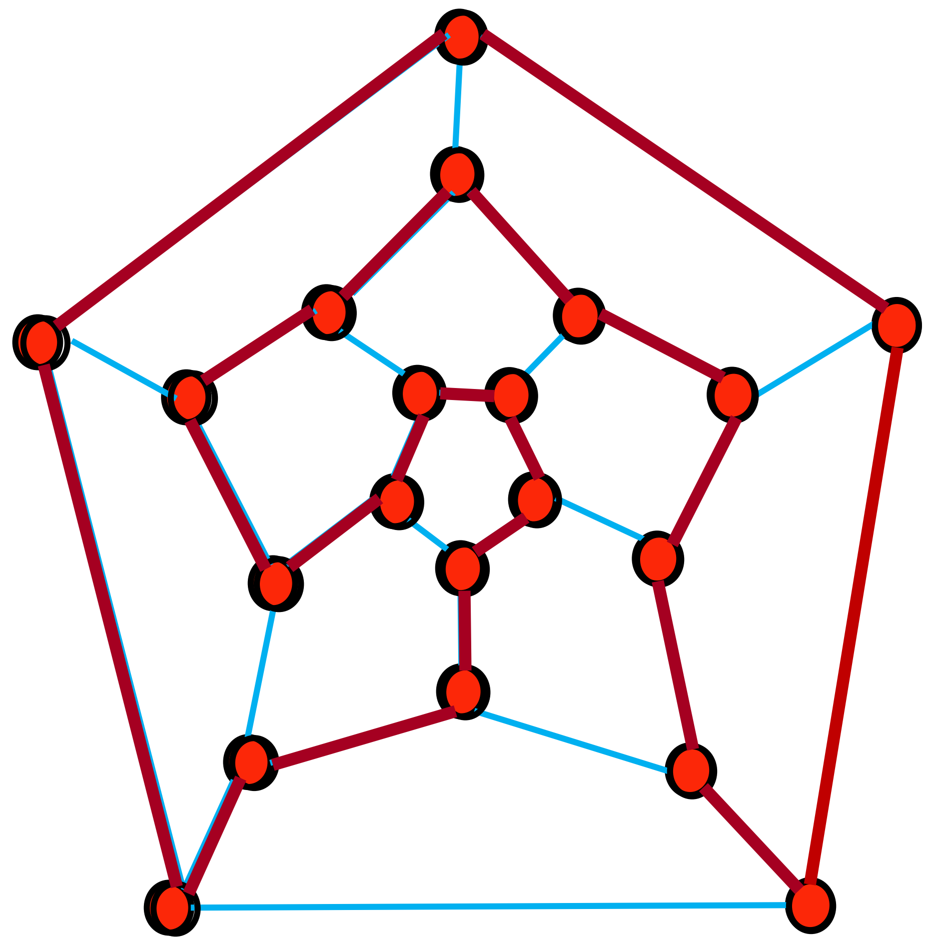
Ví dụ: Đồ thị nào có chu trình Hamilton, đường đi Hamilton ?



Ví dụ

- Đồ thị Hamilton





4.2.2 ĐỊNH LÝ VỀ SỰ TỒN TẠI ĐƯỜNG ĐI HAMILTON

- Cho đến nay, việc tìm ra một **tiêu chuẩn để nhận biết đồ thị Hamilton vẫn còn mở**, mặc dù đây là vấn đề trung tâm của lý thuyết đồ thị. Hiện vẫn chưa có thuật toán thực sự hiệu quả để kiểm tra một đồ thị có phải là đồ thị Hamilton hay không
- Các kết quả thu được phần lớn là điều kiện đủ để một đồ thị là đồ thị Hamilton, có dạng "*Nếu G có số cạnh đủ lớn thì G là Hamilton*".

Sau đây là một số định lý:

❖ **Định lý 3** (Dirac -1952): Đồ thị vô hướng G có n đỉnh (n ≥ 3). Khi đó nếu mọi đỉnh v của G đều có **deg(v) ≥ n/2** thì G có chu trình Hamilton.

❖ **Định lý 4:** Đồ thị có hướng G liên thông mạnh và có n đỉnh. Nếu **$deg^+(v) \geq n / 2$ và $deg^-(v) \geq n / 2$** với mọi đỉnh v thì G có chu trình Hamilton.

4.2.2 ĐỊNH LÝ VỀ SỰ TỒN TẠI ĐƯỜNG ĐI HAMILTON

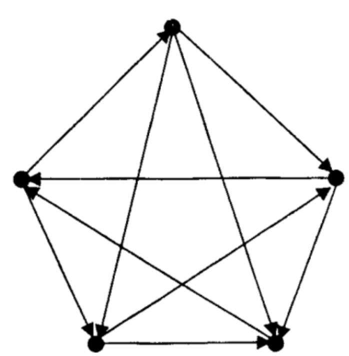
❖ **Định lý Ore:** Nếu *G* đơn đồ thị vô hướng liên thông với $n \geq 3$ đỉnh, và **deg(*u*) + deg(*v*) ≥ *n*** với mọi cặp đỉnh không kề nhau *u*, *v*, thì *G* có chu trình Hamilton.

❖ **Định lý 5:**

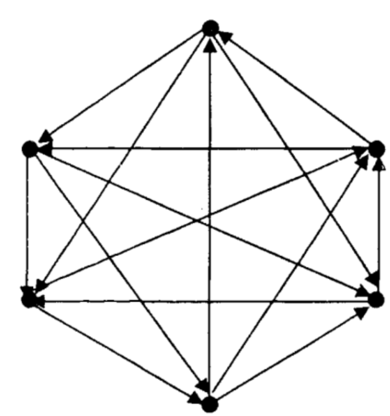
(i) Mọi đồ thị đấu loại là nửa Hamilton.

(ii) Mọi đồ thị đấu loại liên thông mạnh là Hamilton.

✓ **Ví dụ:** Đồ thi đấu loại D_5, D_6



D_5



D_6

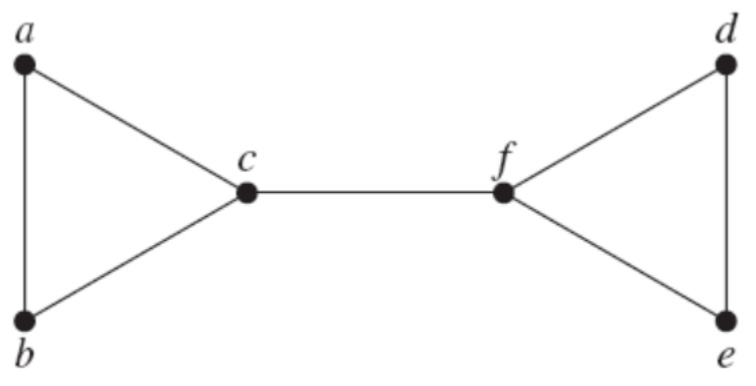
Paul Adrien Maurice Dirac
1902 - 1984
(USA)

Oystein Ore
1899 - 1968
(Norway)

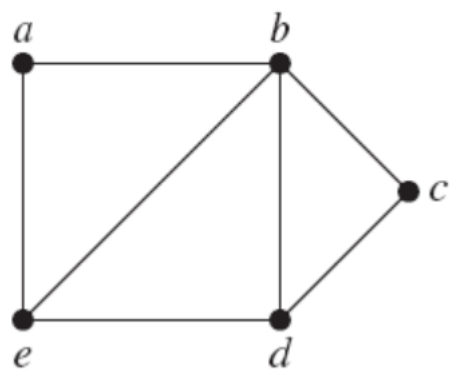
Bài tập

Xác định các đồ thị sau có chu trình và đường đi Hamilton?

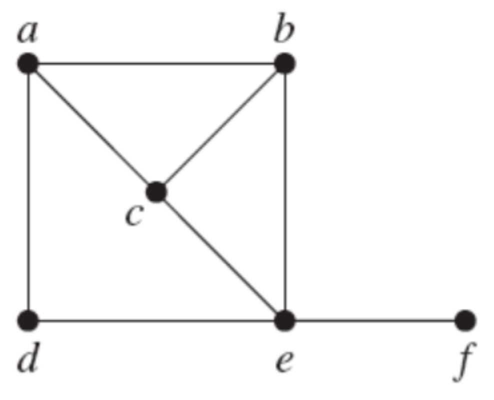
30.



31.



32.



4.2.3 THUẬT TOÁN TÌM CHU TRÌNH HAMILTON

- Cho tới nay, người ta vẫn chưa tìm ra một phương pháp nào thực sự hiệu quả hơn phương pháp quay lui để tìm chu trình Hamilton cũng như đường đi Hamilton trong trường hợp đồ thị tổng quát.
- Sau đây mô tả thuật toán liệt kê tất cả chu trình Hamilton bắt đầu tại đỉnh k dựa trên thuật toán quay lui:

4.2.3 THUẬT TOÁN TÌM CHU TRÌNH HAMILTON

Thuật toán *Hamilton(int k)*

```
{
/* Liệt kê các chu trình Hamilton của đồ thị bằng cách phát triển dãy đỉnh
(X[1], X[2], . . ., X[k-1] ) của đồ thị G = (V, E) */
for y ∈ Ke(X[k-1])

  {    if (k==n+1) and (y == v0) then
                              Ghinhan(X[1], X[2], . . ., X[n], v0);
      else {
                X[k]=y; chuaxet[y] = false;
               Hamilton(k+1);
               chuaxet[y] = true;  }
  }
}
```

4.2.3 THUẬT TOÁN TÌM CHU TRÌNH HAMILTON

Khi đó, việc liệt kê chu trình Hamilton được thực hiện như sau:

Begin

for (v $\in$ V) chuaxet[v] = true;

*/*thiết lập trạng thái các đỉnh*/*

*X[1] = v0; /*v0 là một đỉnh nào đó của đồ thị*/*

chuaxet[v0] = false;

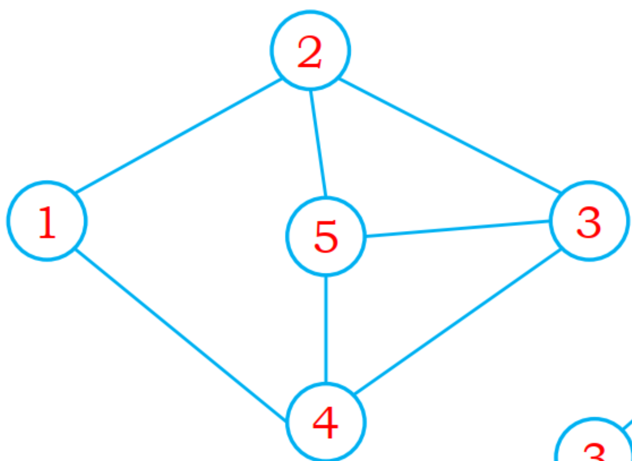
Hamilton(2);

End.

Bài tập: SV lập trình thuật toán quay lui tìm chu trình Hamilton.

4.2.3 THUẬT TOÁN TÌM CHU TRÌNH HAMILTON

Ví dụ: Áp dụng thuật toán tìm chu trình Hamilton cho đồ thị sau



Cây tìm kiếm chu trình Hamilton