§7. CHU TRÌNH HAMILTON, ĐƯỜNG ĐI HAMILTON, ĐỒ THỊ HAMILTON

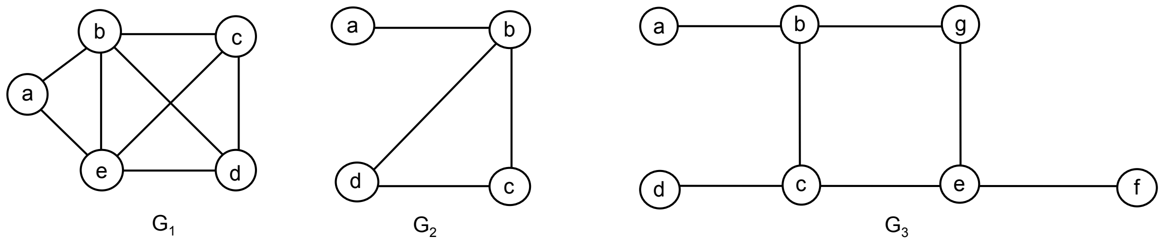
I. ĐỊNH NGHĨA

Cho đồ thị G = (V, E) có n đỉnh

1. Chu trình $(x_1, x_2, ..., x_n, x_1)$ được gọi là chu trình Hamilton nếu $x_i \neq x_j$ với $1 \leq i < j \leq n$
2. Đường đi $(x_1, x_2, ..., x_n)$ được gọi là đường đi Hamilton nếu $x_i \neq x_j$ với $1 \leq i < j \leq n$

Có thể phát biểu một cách hình thức: Chu trình Hamilton là chu trình xuất phát từ 1 đỉnh, đi thăm tất cả những đỉnh còn lại mỗi đỉnh đúng 1 lần, cuối cùng quay trở lại đỉnh xuất phát. Đường đi Hamilton là đường đi qua tất cả các đỉnh của đồ thị, mỗi đỉnh đúng 1 lần. Khác với khái niệm chu trình Euler và đường đi Euler, một chu trình Hamilton không phải là đường đi Hamilton bởi có đỉnh xuất phát được thăm tới 2 lần.

Ví dụ: Xét 3 đơn đồ thị G_1, G_2, G_3 sau:



Đồ thị G_1 có chu trình Hamilton (a, b, c, d, e, a). G_2 không có chu trình Hamilton vì deg(a) = 1 nhưng có đường đi Hamilton (a, b, c, d). G_3 không có cả chu trình Hamilton lẫn đường đi Hamilton

II. ĐỊNH LÝ

1. Đồ thị vô hướng G, trong đó tồn tại k đỉnh sao cho nếu xoá đi k đỉnh này cùng với những cạnh liên thuộc của chúng thì đồ thị nhận được sẽ có nhiều hơn k thành phần liên thông. Thì khẳng định là G không có chu trình Hamilton. Mệnh đề phản đảo của định lý này cho ta điều kiện cần để một đồ thị có chu trình Hamilton
2. Định lý 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. Đây là một điều kiện đủ để một đồ thị có chu trình Hamilton.
3. 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

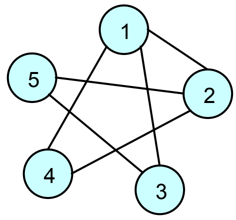
III. CÀI ĐẶT

Dưới đây ta sẽ cài đặt một chương trình liệt kê tất cả các chu trình Hamilton xuất phát từ đỉnh 1, các chu trình Hamilton khác có thể có được bằng cách hoán vị vòng quanh. Lưu ý rằng cho tới nay, người ta vẫn **chưa tìm ra** một phương pháp nào thực sự hiệu quả hơn phương pháp quay lui để tìm dù chỉ một chu trình Hamilton cũng như đường đi Hamilton trong trường hợp đồ thị tổng quát.

Input: file văn bản HAMILTON.INP

- Dòng 1 ghi số đỉnh n (≤ 100) và số cạnh m của đồ thị cách nhau 1 dấu cách
- m dòng tiếp theo, mỗi dòng có dạng hai số nguyên dương u, v cách nhau 1 dấu cách, thể hiện u, v là hai đỉnh kề nhau trong đồ thị

Output: file văn bản HAMILTON.OUT liệt kê các chu trình Hamilton



HAMILTON.INP	HAMILTON.OUT
5 6 1 2 1 3 2 4 3 5 4 1 5 2	1 3 5 2 4 1 1 4 2 5 3 1

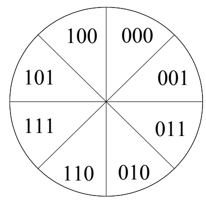
PROG07_1.PAS * Thuật toán quay lui liệt kê chu trình Hamilton

```
program All_of_Hamilton_Circuits;
const
  max = 100;
var
  f: Text;
  a: array[1..max, 1..max] of Boolean;   {Ma trận kề của đồ thị: a[u, v] = True ⇔ (u, v) là cạnh}
  Free: array[1..max] of Boolean;        {Mảng đánh dấu Free[v] = True nếu chưa đi qua đỉnh v}
  X: array[1..max] of Integer;           {Chu trình Hamilton sẽ tìm là; 1=X[1]→X[2] → ... →X[n] →X[1]=1}
  n: Integer;

procedure Enter;       {Nhập dữ liệu từ thiết bị nhập chuẩn Input}
var
  i, u, v, m: Integer;
begin
  FillChar(a, SizeOf(a), False);
  ReadLn(n, m);
  for i := 1 to m do
    begin
      ReadLn(u, v);
      a[u, v] := True;
      a[v, u] := True;
    end;
end;

procedure PrintResult; {In kết quả nếu tìm thấy chu trình Hamilton}
var
  i: Integer;
begin
  for i := 1 to n do Write(X[i], ' ');
  WriteLn(X[1]);
end;

procedure Try(i: Integer); {Thử các cách chọn đỉnh thứ i trong hành trình}
var
  j: Integer;
begin
  for j := 1 to n do                     {Đỉnh thứ i (X[i]) có thể chọn trong những đỉnh}
    if Free[j] and a[x[i - 1], j] then   {kề với X[i - 1] và chưa bị đi qua }
      begin
        x[i] := j;                {Thử một cách chọn X[i]}
        if i < n then             {Nếu chưa thử chọn đến X[n]}
          begin
            Free[j] := False;     {Đánh dấu đỉnh j là đã đi qua}
            Try(i + 1);           {Để các bước thử kế tiếp không chọn phải đỉnh j nữa}
            Free[j] := True;      {Sẽ thử phưng án khác cho X[i] nên sẽ bỏ đánh dấu đỉnh vừa thử}
          end
        else {Nếu đã thử chọn đến X[n]}
          if a[j, X[1]] then PrintResult; {và nếu X[n] lại kề với X[1] thì ta có chu trình Hamilton}
      end;
end;
```

```
begin
   {Định hướng thiết bị nhập/xuất chuẩn}
   Assign(Input, 'HAMILTON.INP'); Reset(Input);
   Assign(Output, 'HAMILTON.OUT'); Rewrite(Output);
   Enter;
   FillChar(Free, n, True);          {Khởi tạo: Các đỉnh đều chưa đi qua}
   x[1] := 1; Free[1] := False;      {Bắt đầu từ đỉnh 1}
   Try(2);                           {Thử các cách chọn đỉnh kế tiếp}
   Close(Input);
   Close(Output);
end.
```

Bài tập:

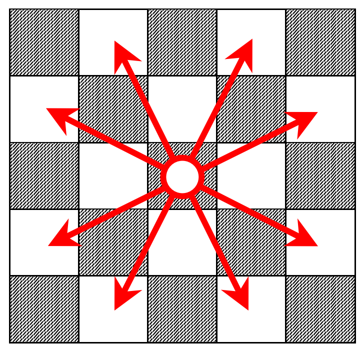
1. Lập chương trình nhập vào một đồ thị và chỉ ra đúng một chu trình Hamilton nếu có.
2. Lập chương trình nhập vào một đồ thị và chỉ ra đúng một đường đi Hamilton nếu có.
3. Trong đám cưới của Péc-xây và An-đrơ-nét có 2n hiệp sỹ. Mỗi hiệp sỹ có không quá n - 1 kẻ thù. Hãy giúp Ca-xi-ô-bê, mẹ của An-đrơ-nét xếp 2n hiệp sỹ ngồi quanh một bàn tròn sao cho không có hiệp sỹ nào phải ngồi cạnh kẻ thù của mình. Mỗi hiệp sỹ sẽ cho biết những kẻ thù của mình khi họ đến sân rồng.
4. Gray code: Một hình tròn được chia thành 2^n hình quạt đồng tâm. Hãy xếp tất cả các xâu nhị phân độ dài n vào các hình quạt, mỗi xâu vào một hình quạt sao cho bất cứ hai xâu nào ở hai hình quạt cạnh nhau đều chỉ khác nhau đúng 1 bít. Ví dụ với n = 3 ở hình vẽ bên



5. *__Thách đố:__ Bài toán mã đi tuần: Trên bàn cờ tổng quát kích thước n x n ô vuông (n chẵn và $6 \le n \le 20$). Trên một ô nào đó có đặt một quân mã. Quân mã đang ở ô (X_1, Y_1) có thể di chuyển sang ô (X_2, Y_2) nếu $|X_1-X_2|.|Y_1-Y_2| = 2$ (Xem hình vẽ).

   Hãy tìm một hành trình của quân mã từ ô xuất phát, đi qua tất cả các ô của bàn cờ, mỗi ô đúng 1 lần.

   Ví dụ:



Với n = 8; ô xuất phát (3, 3).							
45	42	3	18	35	20	5	8
2	17	44	41	4	7	34	21
43	46	1	36	19	50	9	6
16	31	48	59	40	33	22	51
47	60	37	32	49	58	39	10
30	15	64	57	38	25	52	23
61	56	13	28	63	54	11	26
14	29	62	55	12	27	24	53

Với n = 10; ô xuất phát (6, 5)									
18	71	100	43	20	69	86	45	22	25
97	42	19	70	99	44	21	24	87	46
72	17	98	95	68	85	88	63	26	23
41	96	73	84	81	94	67	90	47	50
16	83	80	93	74	89	64	49	62	27
79	40	35	82	1	76	91	66	51	48
36	15	78	75	92	65	2	61	28	53
39	12	37	34	77	60	57	52	3	6
14	33	10	59	56	31	8	5	54	29
11	38	13	32	9	58	55	30	7	4

Gợi ý: Nếu coi các ô của bàn cờ là các đỉnh của đồ thị và các cạnh là nối giữa hai đỉnh tương ứng với hai ô mã giao chân thì dễ thấy rằng hành trình của quân mã cần tìm sẽ là một đường đi Hamilton. Ta có thể xây dựng hành trình bằng thuật toán quay lui kết hợp với phương pháp duyệt ưu tiên Warnsdorff: Nếu gọi deg(x, y) là số ô kề với ô (x, y) và chưa đi qua (kề ở đây theo nghĩa đỉnh kề chứ không phải là ô kề cạnh) thì từ một ô ta sẽ **không thử xét lần lượt các hướng đi** có thể, mà ta sẽ **ưu tiên thử hướng đi tới ô có deg nhỏ nhất trước**. Trong **trường hợp có tồn tại đường đi**, phương pháp này hoạt động với tốc độ tuyệt vời: Với mọi n chẵn trong khoảng từ 6 tới 18, với mọi vị trí ô xuất phát, trung bình thời gian tính từ lúc bắt đầu tới lúc tìm ra một nghiệm < 1 giây. Tuy nhiên trong **trường hợp n lẻ, có lúc không tồn tại đường đi**, do phải duyệt hết mọi khả năng nên thời gian thực thi lại hết sức tồi tệ. (Có xét ưu tiên như trên hay xét thứ tự như trước kia thì cũng vậy thôi. Không tin cứ thử với n lẻ: 5, 7, 9 ... và ô xuất phát (1, 2), sau đó ngồi xem máy tính toát mồ hôi).