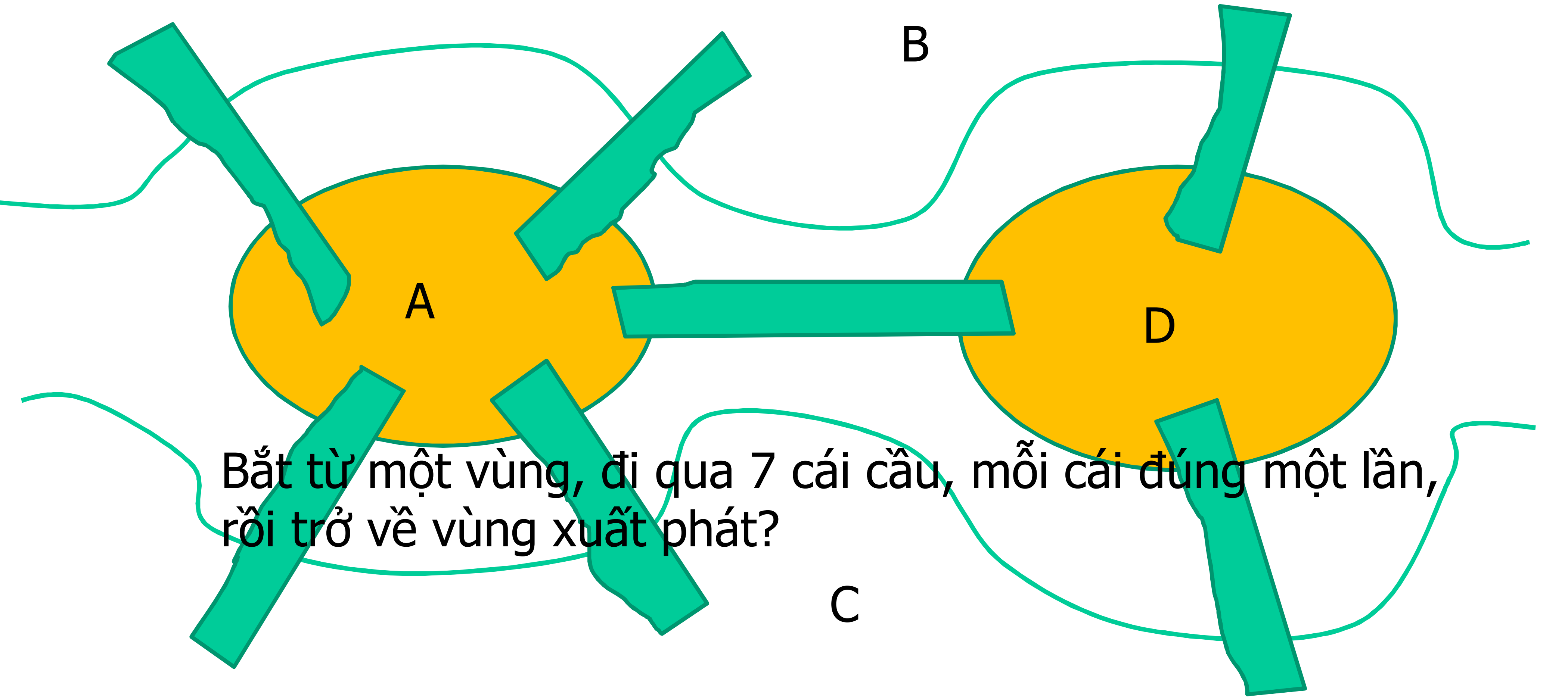
ĐỒ THỊ EULER VÀ HAMILTON

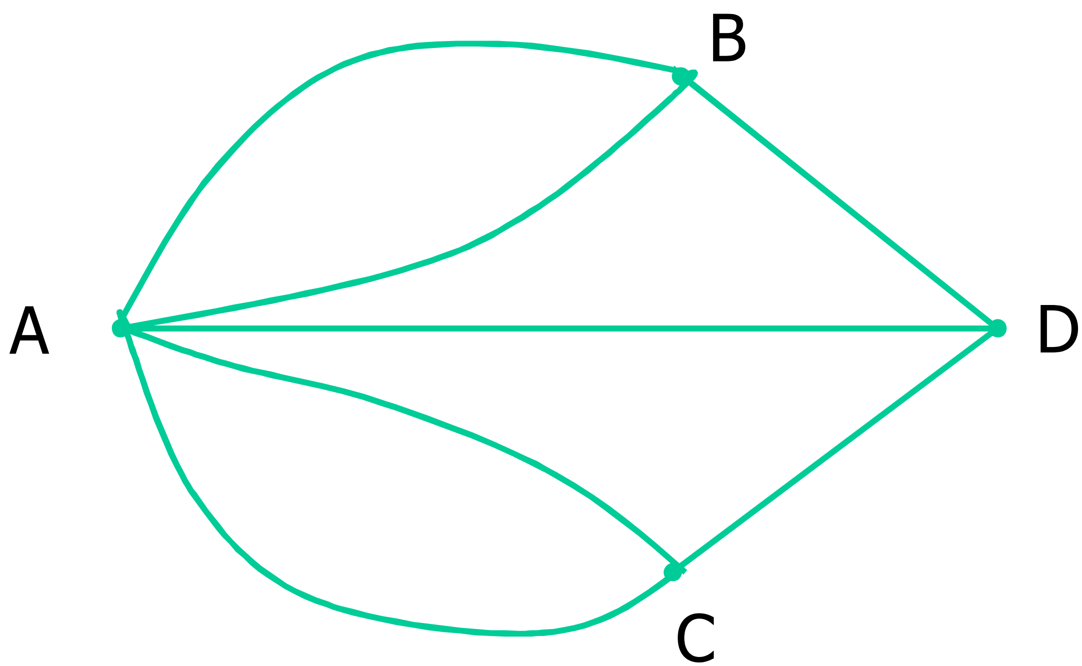
- Đồ thị Euler
- Đồ thị Hamilton

ĐỒ THỊ EULER



Bắt từ một vùng, đi qua 7 cái cầu, mỗi cái đúng một lần, rồi trở về vùng xuất phát?

ĐỒ THỊ EULER

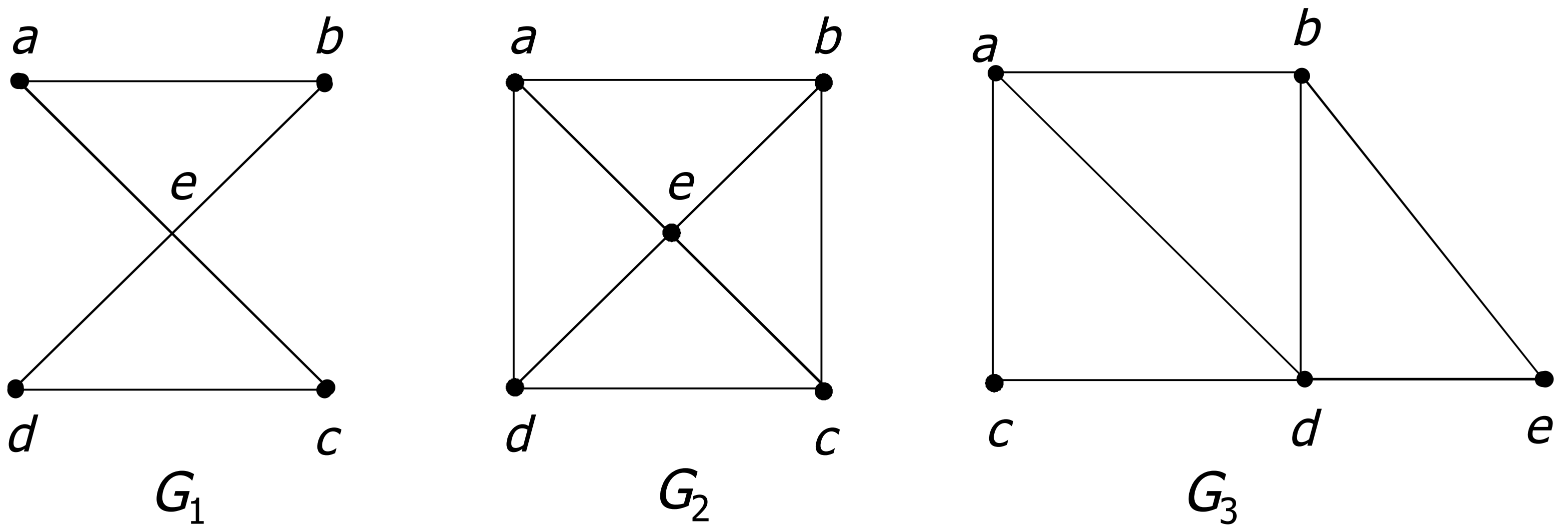


Bài toán không có lời giải

ĐỒ THỊ EULER

- Đường đi đơn trong G đi qua mỗi cạnh của nó một lần gọi là đường đi Euler
- Chu trình đơn trong G đi qua mỗi cạnh của nó một lần gọi là chu trình Euler
- Đồ thị được gọi là đồ thị Euler nếu nó có chu trình Euler và gọi là đồ thị nửa Euler nếu nó có đường đi Euler

ĐỒ THỊ EULER



- G_1 có chu trình Euler, ví dụ a, e, c, d, e, b, a
- G_3 có đường đi Euler a, c, d, e, b, d, a, b nhưng không có chu trình Euler
- G_2 không có đường đi và chu trình Euler

ĐỒ THỊ EULER

- Một đồ thị Euler là nữa Euler (tại sao?)
- **Định lý 1:** Đồ thị vô hướng liên thông G là đồ thị Euler khi và chỉ khi mọi đỉnh của G đều có bậc chẵn
- **Chứng minh?**

ĐỒ THỊ EULER

- Giả sử G là đồ thị Euler
  - Gọi C là chu trình Euler trong G
  - Xét đỉnh v bất kỳ của G, giả sử C đi qua nó k lần
  - Vì mỗi lần C đi qua v có duy nhất một cặp cạnh vào và ra khỏi v thuộc chu trình $\Rightarrow$ có 2k cạnh kề với v nên deg(v)=2k (chẵn)

ĐỒ THỊ EULER

- Giả sử deg(v) chẵn với mọi v trong G, chu trình Euler trong G được xây dựng như sau:
  - Bắt đầu từ 1 đỉnh u đi theo các cạnh một cách tùy ý nhưng không lặp lại cạnh nào đã đi qua cho đến khi không thể đi tiếp được nữa phải dừng ở đỉnh w, lúc này mọi cạnh tới w đã đi qua (vì ngược lại sẽ có cạnh ra khỏi w chưa đi qua, nên chưa thể dừng ở w)
  - Rõ ràng $w \equiv u$, vì nếu $w \neq u$ thì số lần tới w nhiều hơn số lần ra khỏi w là 1 nên deg(w) lẻ trái với giả thiết, vậy ta có chu trình C= u,v,..., z,u

ĐỒ THỊ EULER

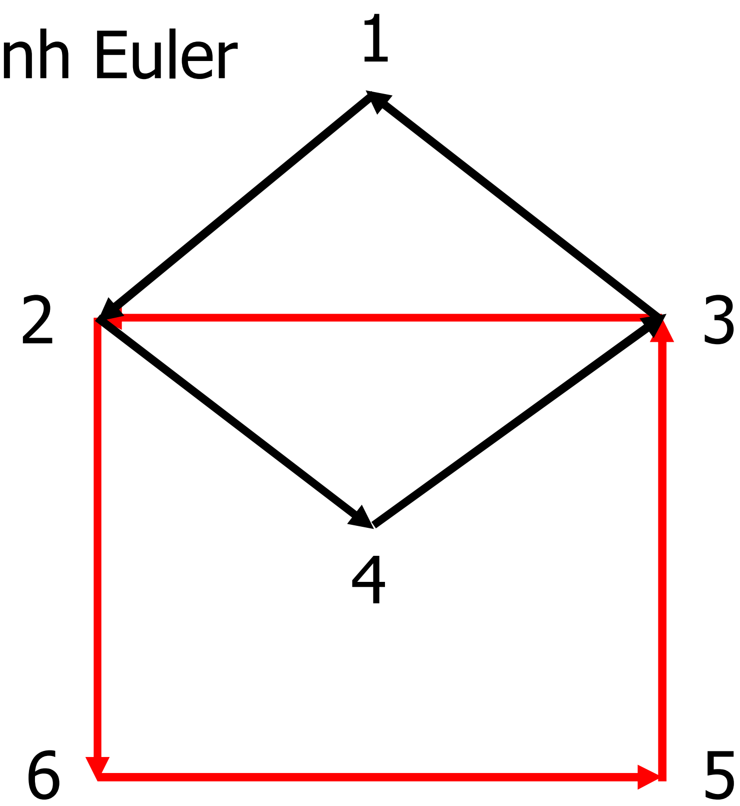
- Nếu mọi cạnh của G thuộc chu trình thì C là chu trình Euler
- Ngược lại gọi s là một đỉnh trong chu trình này liên thuộc với 1 cạnh mà C chưa đi qua (s tồn lại do G liên thông)

ĐỒ THỊ EULER

- Mở rộng (breakout) C thành chu trình lớn hơn bằng cách khởi hành lại từ s, đi theo chu trình C cho đến khi hoàn tất nó tại s, rồi tiếp tục đi theo cạnh liên thuộc với s mà chu trình cũ chưa đi qua nói trên cho đến khi phải dừng lại, ta được một chu trình mới chứa chu trình cũ
- Cứ tiếp tục quá trình "thành lập và mở rộng chu trình" cho đến khi thu được chu trình C không thể mở rộng hơn được nữa (do G hữu hạn, điều này sẽ xẩy ra khi không còn cạnh nào liên thuộc với một đỉnh trong C mà chưa được đi qua)

ĐỒ THỊ EULER

Xây dựng chu trình Euler



ĐỒ THỊ EULER

- Gọi e =(x, y) là một cạnh bất kỳ của G, vì G liên thông nên có đường đi u, u_1, u_2, …, x từ u đến x
- Cạnh (u, u_1) phải thuộc chu trình C (vì không còn cạnh nào tới u chưa đi qua), suy ra u_1 thuộc chu trình C
- Tương tự (u_1, u_2) thuộc chu trình nên u_2 thuộc chu trình, cứ tiếp tục ta suy ra x thuộc chu trình C

ĐỒ THỊ EULER

- Vì mọi cạnh đi qua x đều thuộc chu trình C nên e =(x, y) thuộc chu trình C
- Vì vậy C là một chu trình Euler và do đó G là đồ thị Euler

ĐỒ THỊ EULER

- Thuật toán Flor: Xuất phát từ một đỉnh u bất kỳ của đồ thị, đi theo các cạnh của nó một cách tùy ý, và đảm bảo 2 qui tắc sau:
  - Xóa bỏ cạnh đã đi qua và đồng thời xóa bỏ cả những đỉnh cô lập được tạo thành
  - Ở mỗi bước, chỉ đi qua cạnh cầu khi không còn lựa chọn nào khác

ĐỒ THỊ EULER

```
Euler_Cycle(G) // dựa trên thuật toán Flor
S ← ∅;  C ← ∅;  u ← Select(V[G]); Push(S, u)
While S ≠ ∅
   do   x ←Top(S)
          if Adj[x] ≠ ∅
                then   y ← First(Adj[x])
                       Push(S,y)
                       Adj[x] ← Adj[x] -{y}
                       Adj[y] ← Adj[y]-{x}
                else   x ← Pop(S)
                       Push(C, x)
```

ĐỒ THỊ EULER

- Độ phức tạp của thuật toán Euler_Cycle là O(m), m là số cạnh của đồ thị

ĐỒ THỊ EULER

- **Hệ quả 1**: Đồ thị vô hướng liên thông G là nửa Euler khi và chỉ khi nó có không quá 2 đỉnh bậc lẽ
- Chứng minh?

ĐỒ THỊ EULER

- Chứng minh hệ quả:
  - Nếu G không có đỉnh bậc lẽ thì nó là đồ thị Euler nên nó là nữa Euler
  - Nếu G có đúng 2 đỉnh bậc lẽ là u và v thì tạo đồ thị H bằng cách thêm vào G đỉnh w và 2 cạnh (w, u), (w, v). Khi đó H là Euler, nên có chu trình Euler C. Xoá bỏ khỏi chu trình này đỉnh w và hai cạnh (w, u), (w, v) ta có đường đi Euler

ĐỒ THỊ EULER

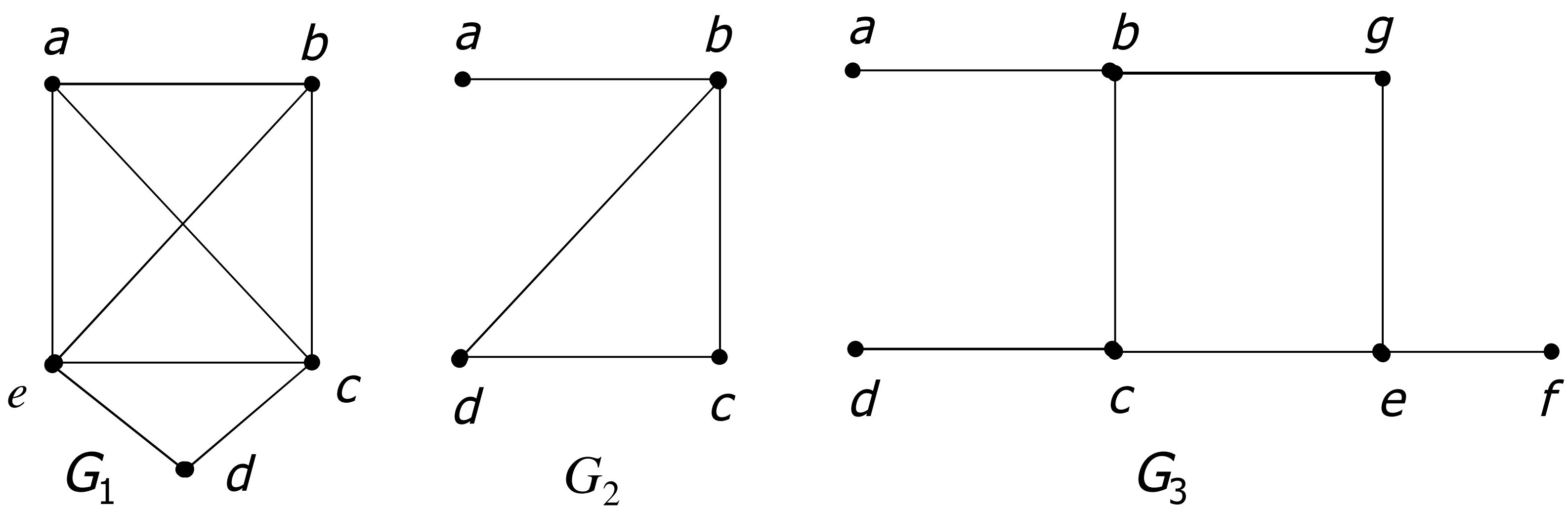
- **Định lý 2** Đồ thị có hướng liên thông mạnh G là đồ thị Euler khi và chỉ khi

  $\deg^{+}(v) = \deg^{-}(v)$ với mọi v của G

ĐỒ THỊ HAMILTON

- Đường đi qua tất cả các đỉnh của đồ thị, mỗi đỉnh đúng một lần gọi là đường đi Hamilton
- Chu trình đi qua tất cả các đỉnh của đồ thị, mỗi đỉnh đúng một lần gọi là chu trình Hamilton
- Đồ thị được gọi là đồ thị Hamilton nếu nó có chu trình Hamilton và gọi là đồ thị nửa Hamilton nếu nó có đường đi Hamilton

ĐỒ THỊ HAMILTON



G_1 có chu trình Hamilton *a, b, c, d, e, a*
G_2 không có chu trình Hamilton nhưng có đường đi Hamilton: *a, b, c, d*
*G*3 không có chu trình và đường đi Hamilton

ĐỒ THỊ HAMILTON

Lưu ý

- Chưa có điều kiện cần và đủ để một đồ thị là đồ thị Hamilton (bài toán mở)
- Chỉ có một số điều kiện đủ
- Chưa có thuật giải hiệu quả để xây dựng chu trình Hamilton

ĐỒ THỊ HAMILTON

Qui tắc tìm chu trình Hamilton

- Nếu một đỉnh của G có deg(v)$\leq$1 thì G không có chu trình Hamilton
- Nếu x có bậc 2 thì cả hai cạnh tới x (kề x) đều thuộc chu trình Hamilton
- Chu trình Hamilton không chứa chu trình con thực sự nào

ĐỒ THỊ HAMILTON

- Mọi đồ thị K_n, $n \geq 3$ đều có chu trình Hamilton
- Cho đồ thị G, giả sử G có k đỉnh mà nếu xóa k đỉnh này cùng với các cạnh liên kết với chúng thì đồ thị nhận được có hơn k thành phần. Chứng minh G không có chu trình Hamilton?

ĐỒ THỊ HAMILTON

- **Định lý 3** (Dirak) Đơn đồ thị vô hướng G với n>2 đỉnh, mỗi đỉnh có bậc không nhỏ hơn n/2 là đồ thị Hamilton
- **Định lý 4** (Ore) Đơn đồ thị vô hướng G với $n \geq 3$ sao cho $deg(u) + deg(v) \geq n$ với mọi cặp đỉnh u và v không kề nhau là đồ thị Hamilton
- **Định lý 5** (Dirak tổng quát) Giả sử G là đồ thị có hướng liên thông mạnh với n đỉnh. Nếu

  $$deg^{+}(v) \geq n/2 \text{ và } deg^{-}(v) \geq n/2 \text{ với mọi } v$$

  thì G là đồ thị Hamilton

ĐỒ THỊ HAMILTON

```
Hamilton_Cycle(G, k, v0) //Tìm đỉnh x[k] trong n đỉnh của G
    for y∈ Adj[x[k-1]]
      do  if k=n and y=v0
              then   print(v0, x[1],…x[n-1],v0) //x[0]=v0
              else  if  unvisited[y]=true
                      then  x[k] ← y
                            unvisited[y] ← false
                            Hamilton_Cycle(G, k+1,v0)
                            unvisited[y] ← true
```

ĐỒ THỊ HAMILTON

Lưu ý:

- Giải thuật Hamilton_Cycle(G, k, v_0) tìm tất cả các chu trình Hamilton của G
- Trước khi thực hiện GT cần gán unvisited[y]= true với mọi v $\in$ V[G] và x[0]=v_0 là đỉnh đầu của các chu trình

ĐỒ THỊ HAMILTON

- Độ phức tạp thuật toán Hamilton_Cycle là O(n!), n là số đỉnh của đồ thị