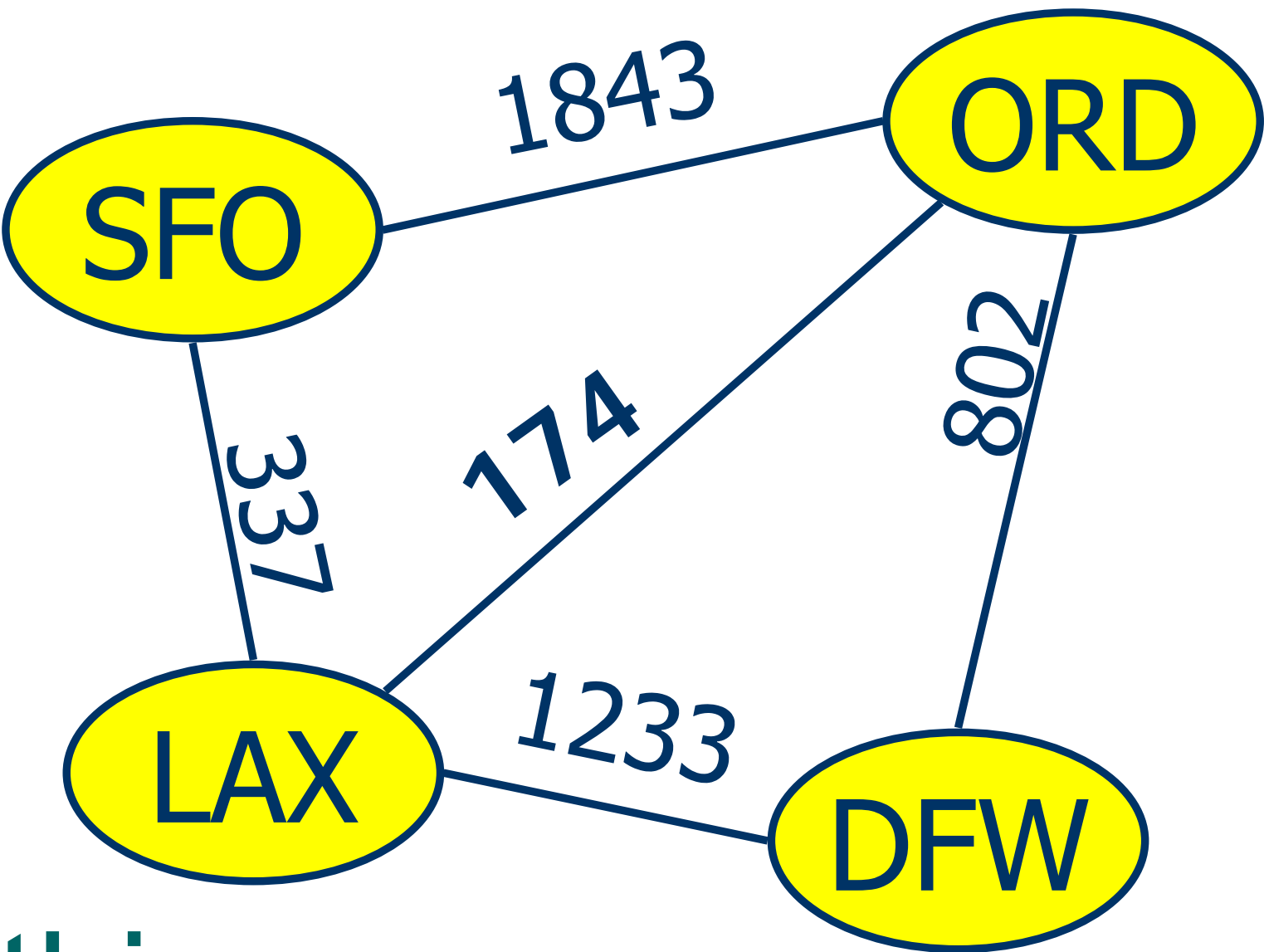


# Cấu trúc dữ liệu và Giải thuật



## Chương VIII: Cấu trúc Đồ thị

Đỗ Bích Diệp - Khoa CNTT

[cuuduongthancong.com](http://cuuduongthancong.com)

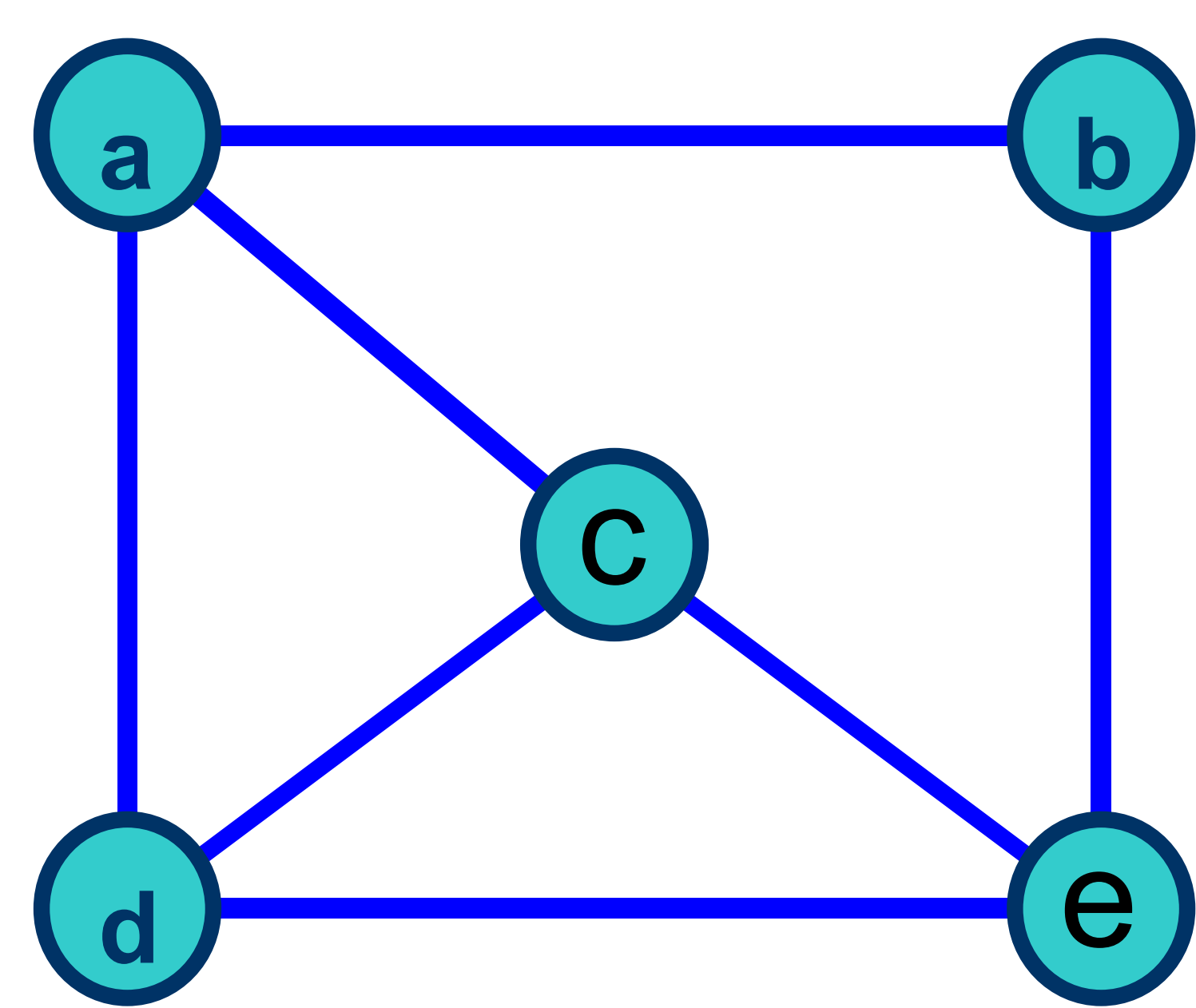
# Chương VIII: Đồ thị

- Nội dung
  - 1. Các khái niệm cơ bản
  - 2. Biểu diễn đồ thị
    - 1. Ma trận lân cận
    - 2. Danh sách lân cận
  - 3. Duyệt đồ thị
  - 4. Bài toán áp dụng
    - 1. Tìm cây khung cực tiểu
    - 2. Tìm đường đi ngắn nhất
    - 3. Bài toán bao đóng truyền ứng
    - 4. Bài toán sắp xếp tô pô

Đỗ Bích Diệp - Khoa CNTT

Đồ thị

- Một đồ thị  $G = (V, E)$  trong đó
  - $V$ : tập các đỉnh (vertices)
  - $E$ : tập các cung (edges) nối các đỉnh trong  $V$
- Một cung  $e = (u,v)$  là một cặp đỉnh
- Ví dụ:



$V = \{a,b,c,d,e\}$

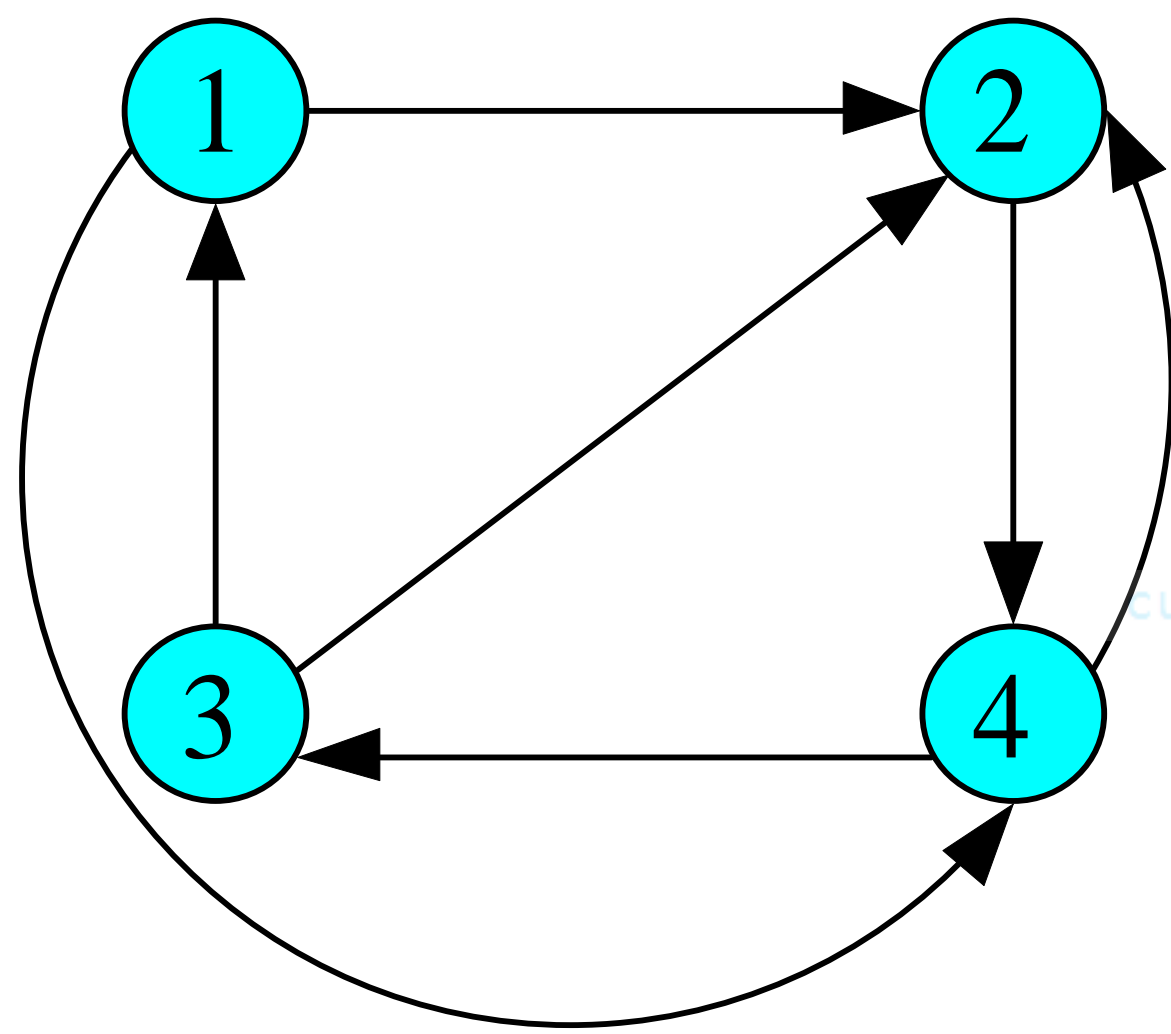
$E = \{(a,b), (a,c), (a,d), (b,e), (c,d), (c,e), (d,e)\}$

Đỗ Bích Diệp - Khoa CNTT

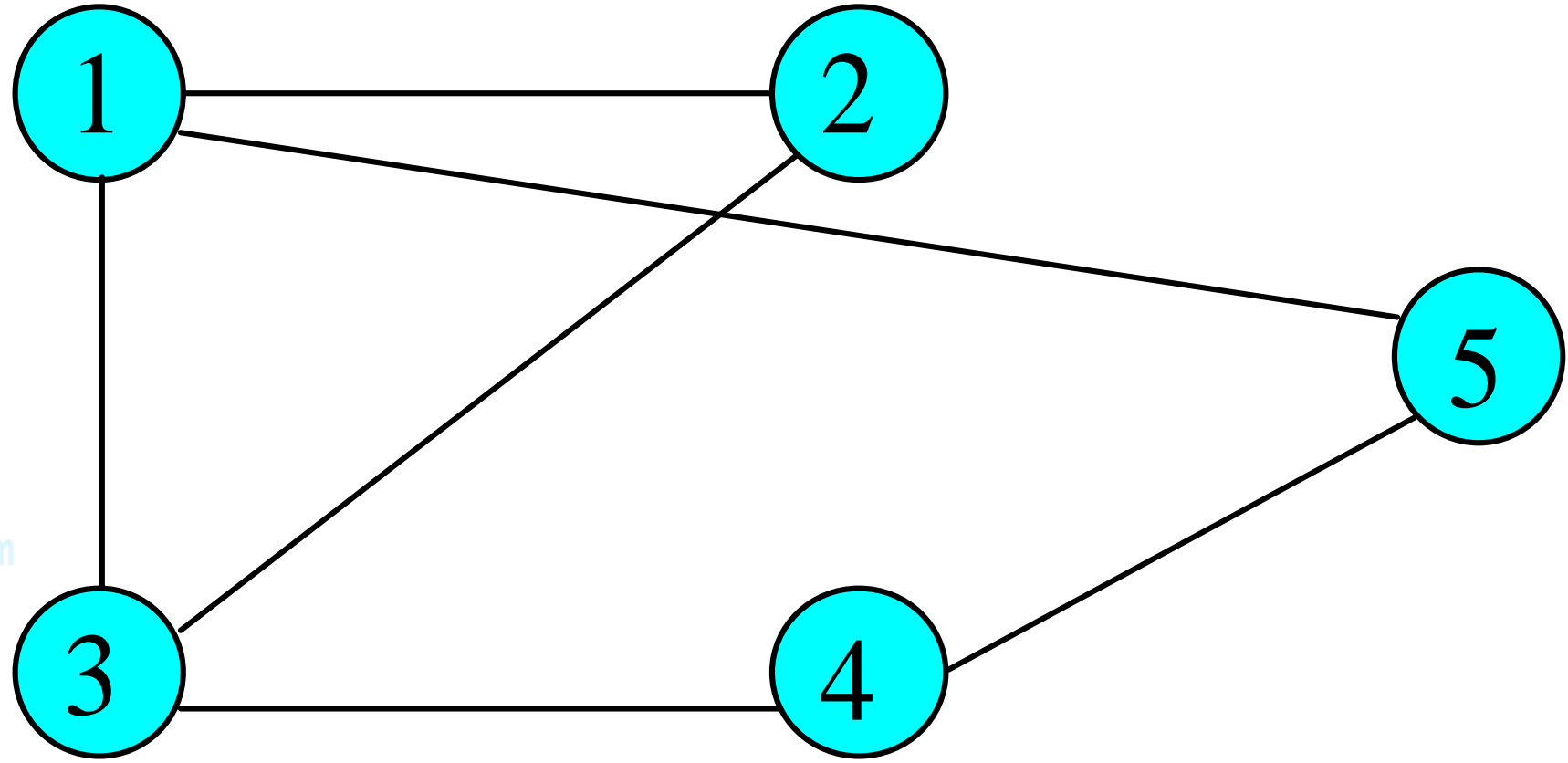
cuu duong than cong . com

Các khái niệm liên quan

- Đồ thị có hướng và Đồ thị vô hướng



Trong một cung, thứ tự của các đỉnh là quan trọng  
Cung  $(u,v)$  khác với cung  $(v,u)$

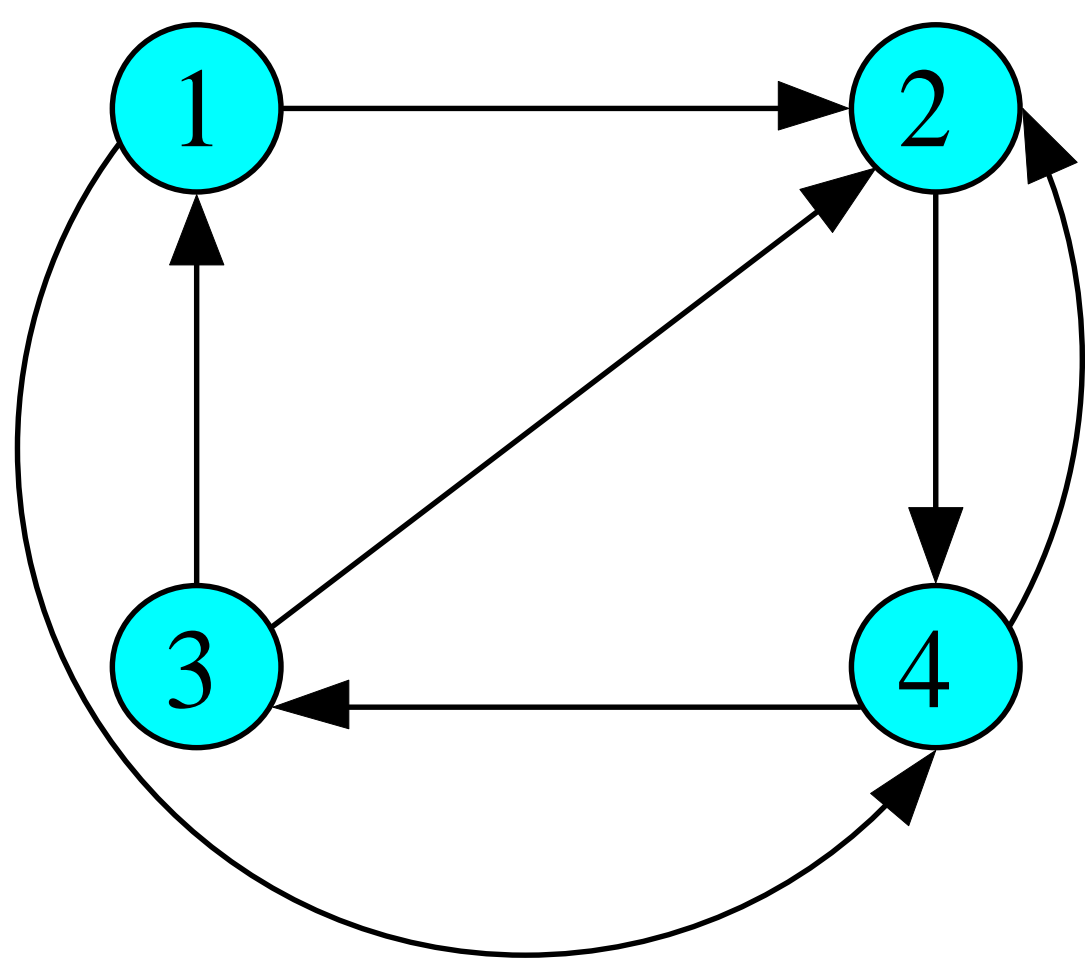


Trong một cung, thứ tự của các đỉnh là không quan trọng  
Cung  $(u,v)$  cũng giống như cung  $(v,u)$

Đỗ Bích Diệp - Khoa CNTT

## Các khái niệm liên quan

- **Bậc của một đỉnh (Degree):** Là số cung kề với đỉnh
  - Trong một đồ thị có hướng, một đỉnh có thể có
    - Bậc trong (in-degree)
    - Bậc ngoài (out-degree)
  - Ví dụ:
    - Đỉnh 1 có bậc 3
    - Đỉnh 1 có bậc trong là 1 và bậc ngoài là 2

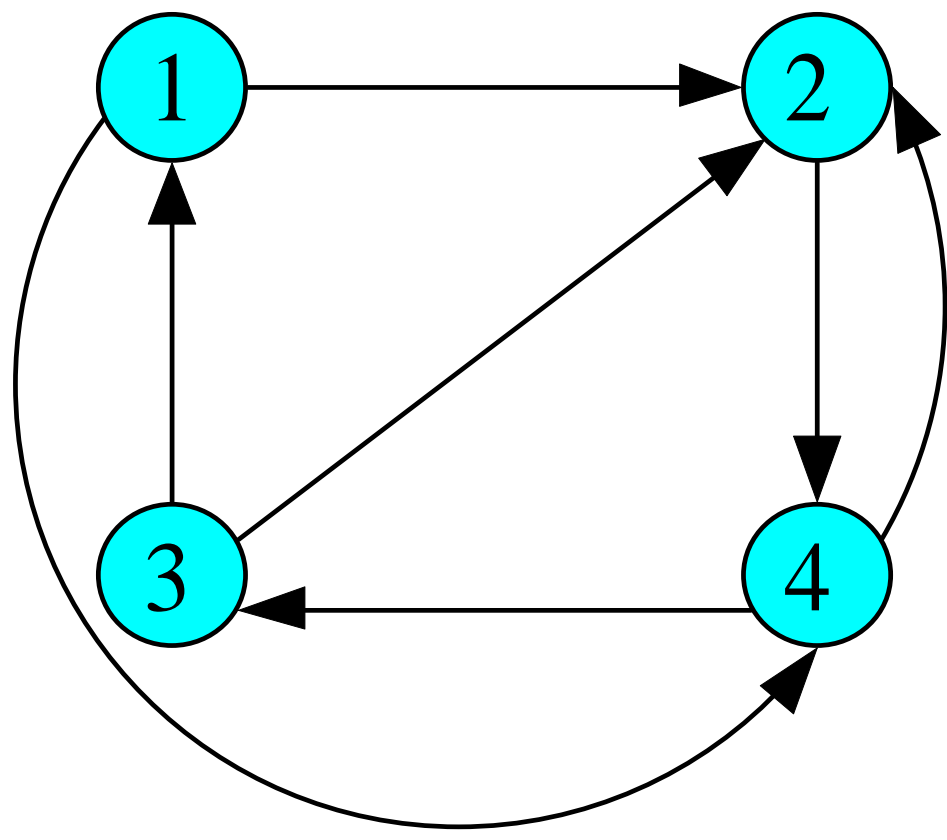


Đỗ Bích Diệp - Khoa CNTT

[cua duong thanh cong . com](http://cua-duong-thanh-cong.com)

## Các khái niệm liên quan

- **Đỉnh lân cận (Adjacent vertices)**
  - Trong đồ thị
    - 1, 2 là lân cận của nhau
    - 1,3 là lân cận của nhau
    - ....
- **Cung kề (Incident edges)**
  - Nếu có cung (u,v) thì cung này là cung kề của hai đỉnh u và v

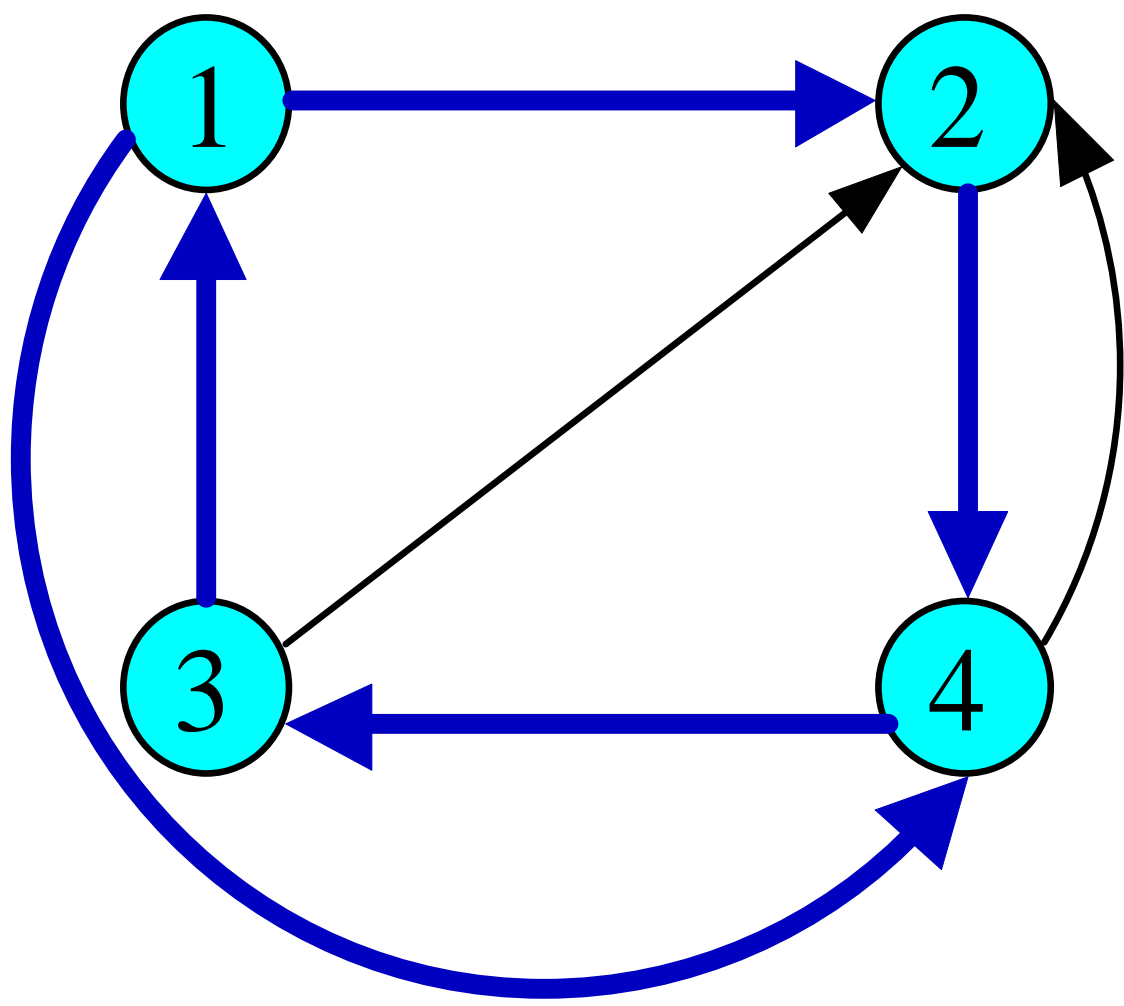


Đỗ Bích Diệp - Khoa CNTT



## Các khái niệm liên quan

- Đường đi
  - Dãy các đỉnh  $v_1, v_2, \dots, v_k$  mà tồn tại cung  $(v_i, v_{i+1})$  trong đồ thị ( $i = 1 \dots k-1$ )
- Đường đi đơn
  - Đường đi với các đỉnh không lặp lại
- Chu trình
  - Đường đi đơn với đỉnh đầu và cuối trùng nhau
- Độ dài đường đi
  - Số cung trên đường đi
- Đồ thị con



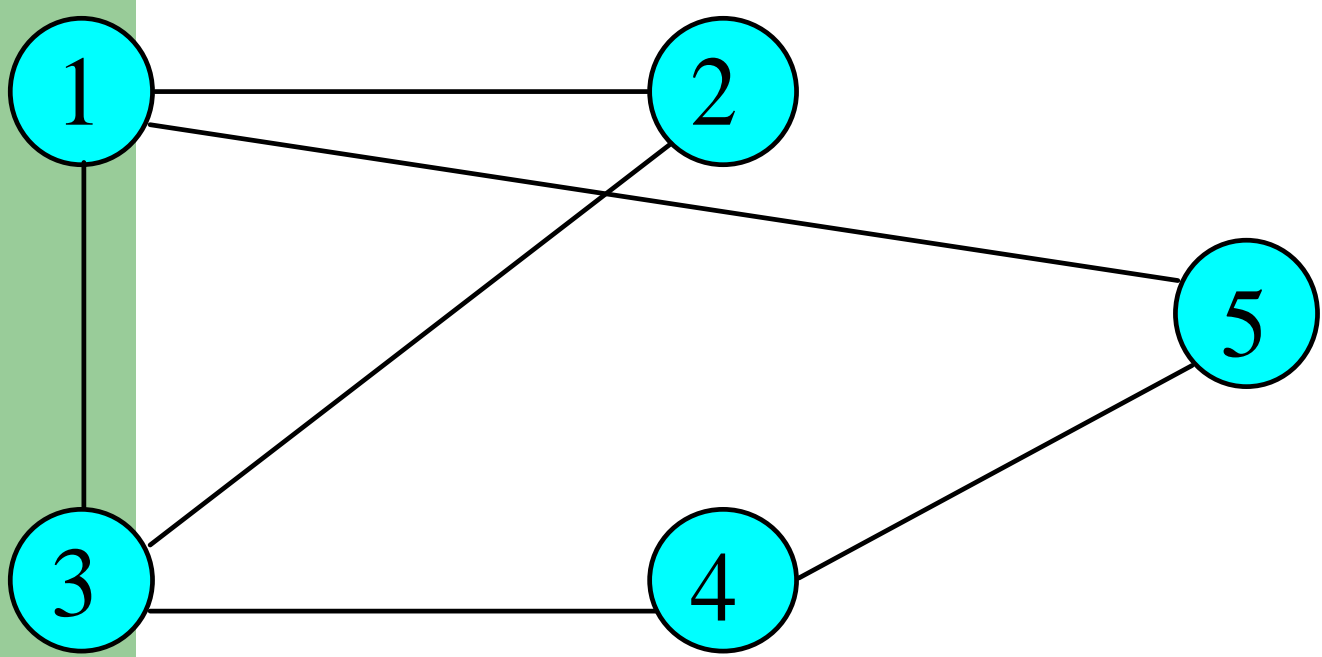
Path : 1, 2, 4, 3, 1, 4

Đỗ Bích Diệp - Khoa CNTT

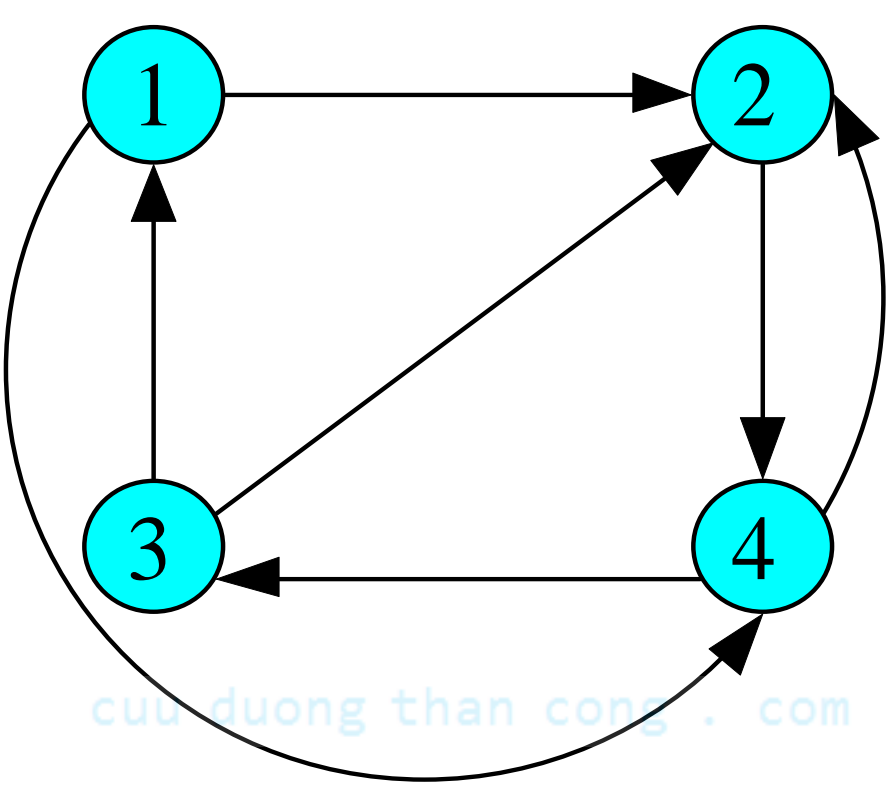
[cuduongthancong.com](http://cuduongthancong.com)

## Các khái niệm liên quan

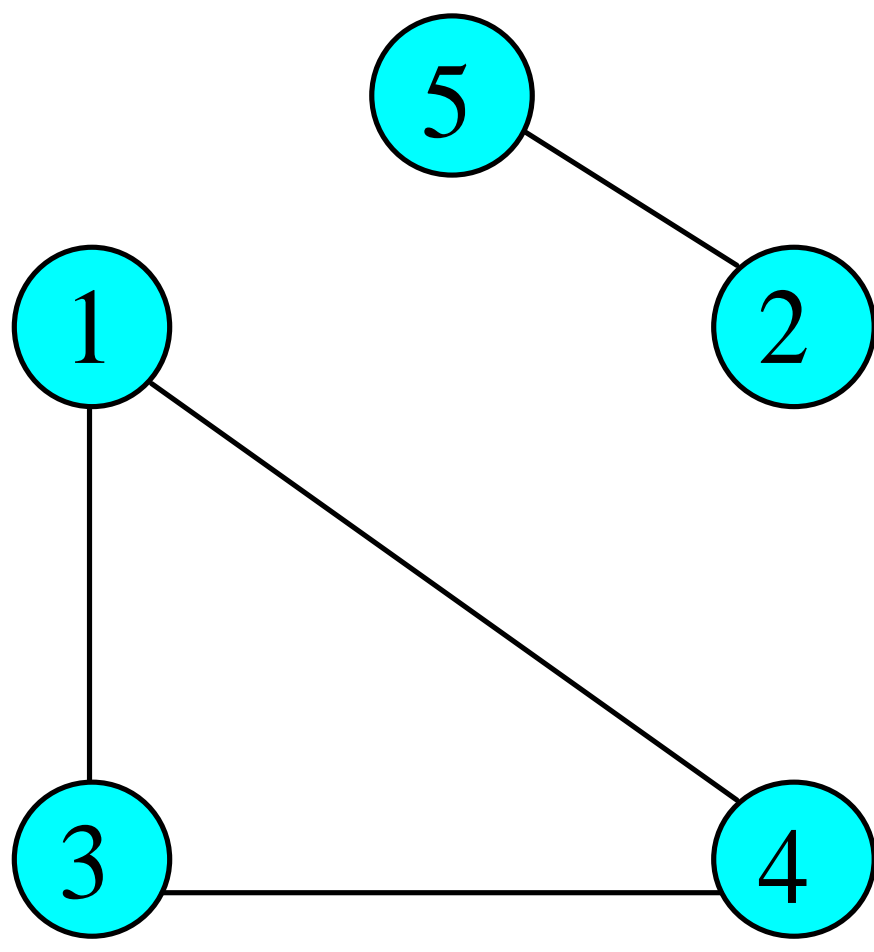
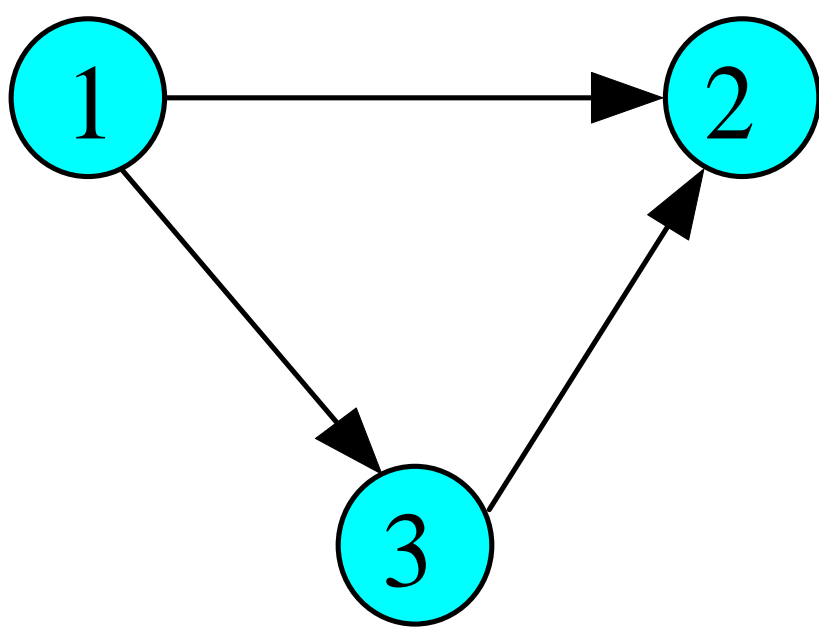
- Đồ thị liên thông (Connected Graph)



Đồ thị liên thông



[cuduongthancong.com](http://cuduongthancong.com)

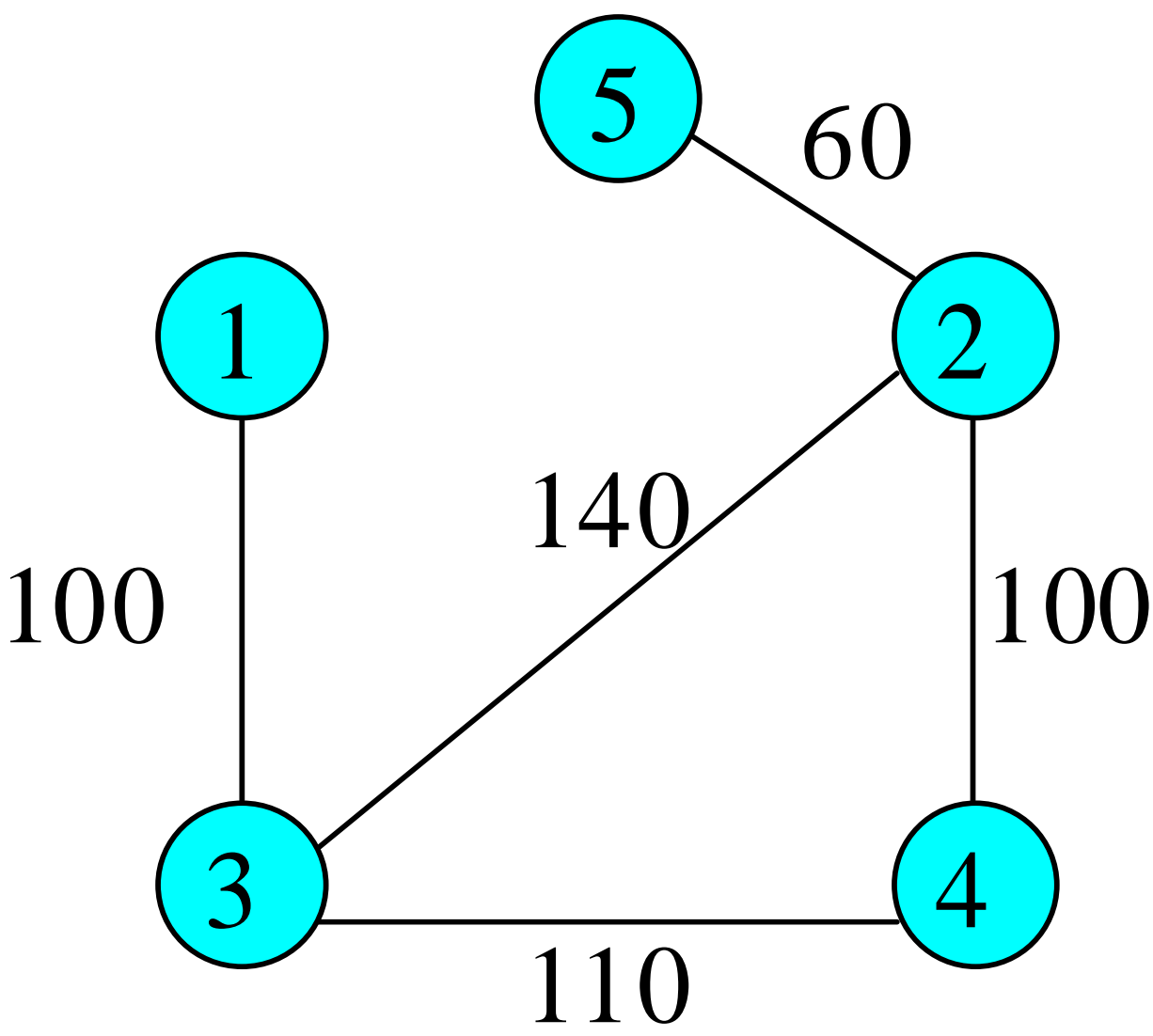


Đồ thị không liên thông

Đỗ Bích Diệp - Khoa CNTT

## Các khái niệm liên quan

- Đồ thị trọng số (Weight Graph)



Đỗ Bích Diệp - Khoa CNTT

[cuuduongthancong.com](http://cuuduongthancong.com)

## Kiểu dữ liệu trừu tượng Đồ thị

- Dữ liệu: Một tập không rỗng các đỉnh chứa các phần tử có kiểu nhất định, một tập không rỗng các cung có thể biểu diễn các phần tử có kiểu nhất định
- Các thao tác cơ bản
  - Graph create()
  - insertVertex( o )
  - insertEdge(u, v, o)
  - removeVertex(v)
  - removeEdge(e )
  - endVertices(e)
  - opposite(v, e)
  - areAdjacent(v, w)
  - adjacentVertices(v)
  - incidentEdges(v)
  - vertices()
  - edges()
  - numVertices()
  - numEdges()

Đỗ Bích Diệp - Khoa CNTT

## Một số tính chất của đồ thị

1. Nếu một đồ thị  $G$  có  $m$  cung thì tổng bậc của các đỉnh trong  $G$  sẽ là  $2m$
2. Nếu một đồ thị có hướng có  $m$  cung thì tổng bậc trong của các đỉnh, tổng bậc ngoài của các đỉnh đều là  $m$
3. Nếu đồ thị  $G$  là đồ thị đơn giản,  $G$  có  $n$  đỉnh và  $m$  cung thì
  1. Nếu  $G$  là đồ thị vô hướng  $m \leq n(n-1)/2$
  2. Nếu  $G$  là đồ thị có hướng thì  $m \leq n(n-1)$

Đỗ Bích Diệp - Khoa CNTT

[cuduongthancong.com](http://cuduongthancong.com)

## Biểu diễn đồ thị

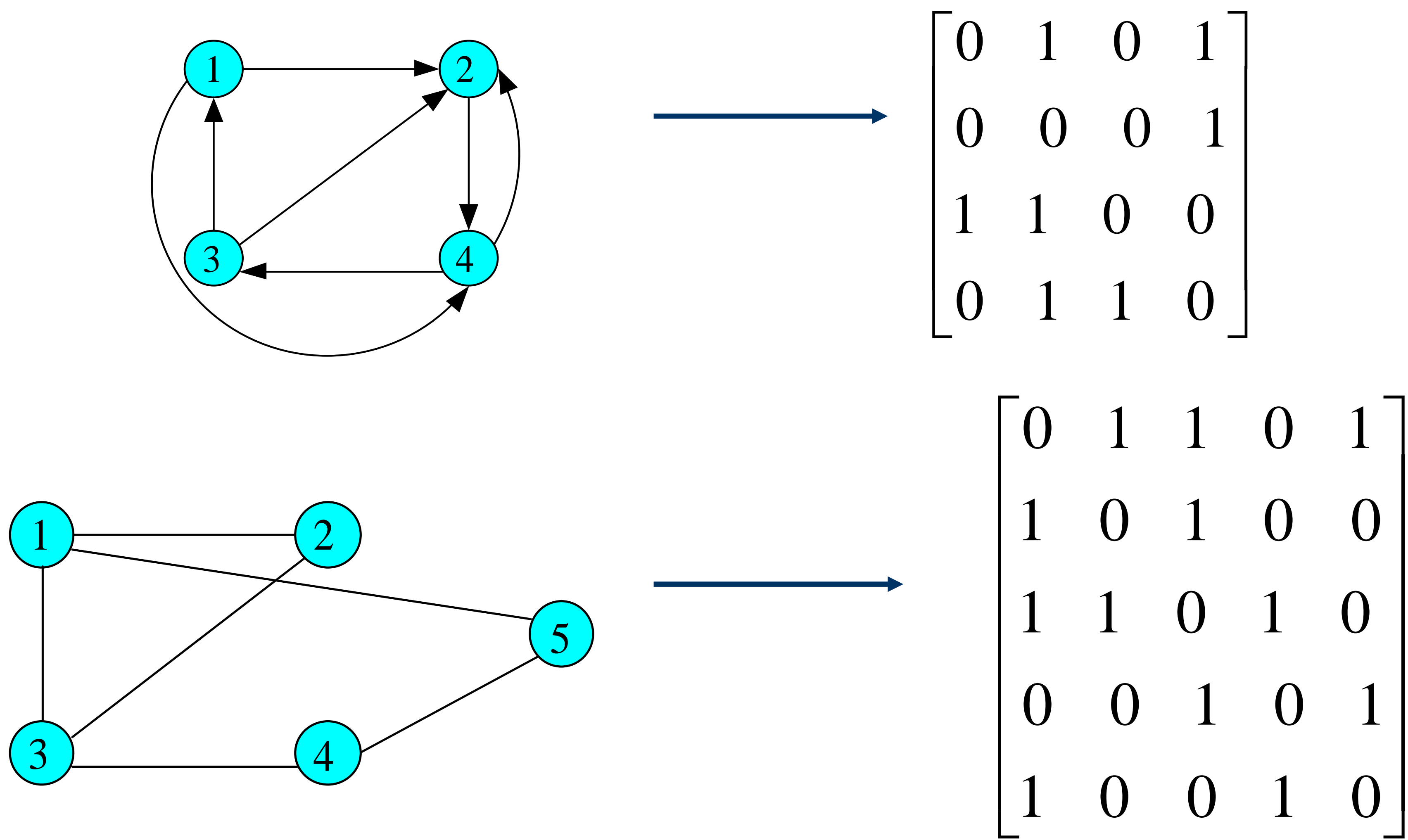
- Biểu diễn bằng ma trận lân cận
  - Đánh số các đỉnh trong tập  $V$  từ 1 đến  $n$
  - Ma trận biểu diễn đồ thị  $A$  ( $n \times n$ )
    - $A_{ij} = 1$  nếu trong  $G$  tồn tại cung  $(i,j)$
    - $A_{ij} = 0$  nếu trong  $G$  không tồn tại cung đó
  - Với đồ thị vô hướng thì nếu  $A_{ij} = 1$  thì  $A_{ji} = 1$
  - $A$  được gọi là ma trận lân cận của  $G$

Đỗ Bích Diệp - Khoa CNTT



## Biểu diễn đồ thị bằng ma trận lân cận

• Ví dụ



Đỗ Bích Diệp - Khoa CNTT

[cuuduongthancong.com](http://cuuduongthancong.com)

## Biểu diễn đồ thị bằng danh sách lân cận

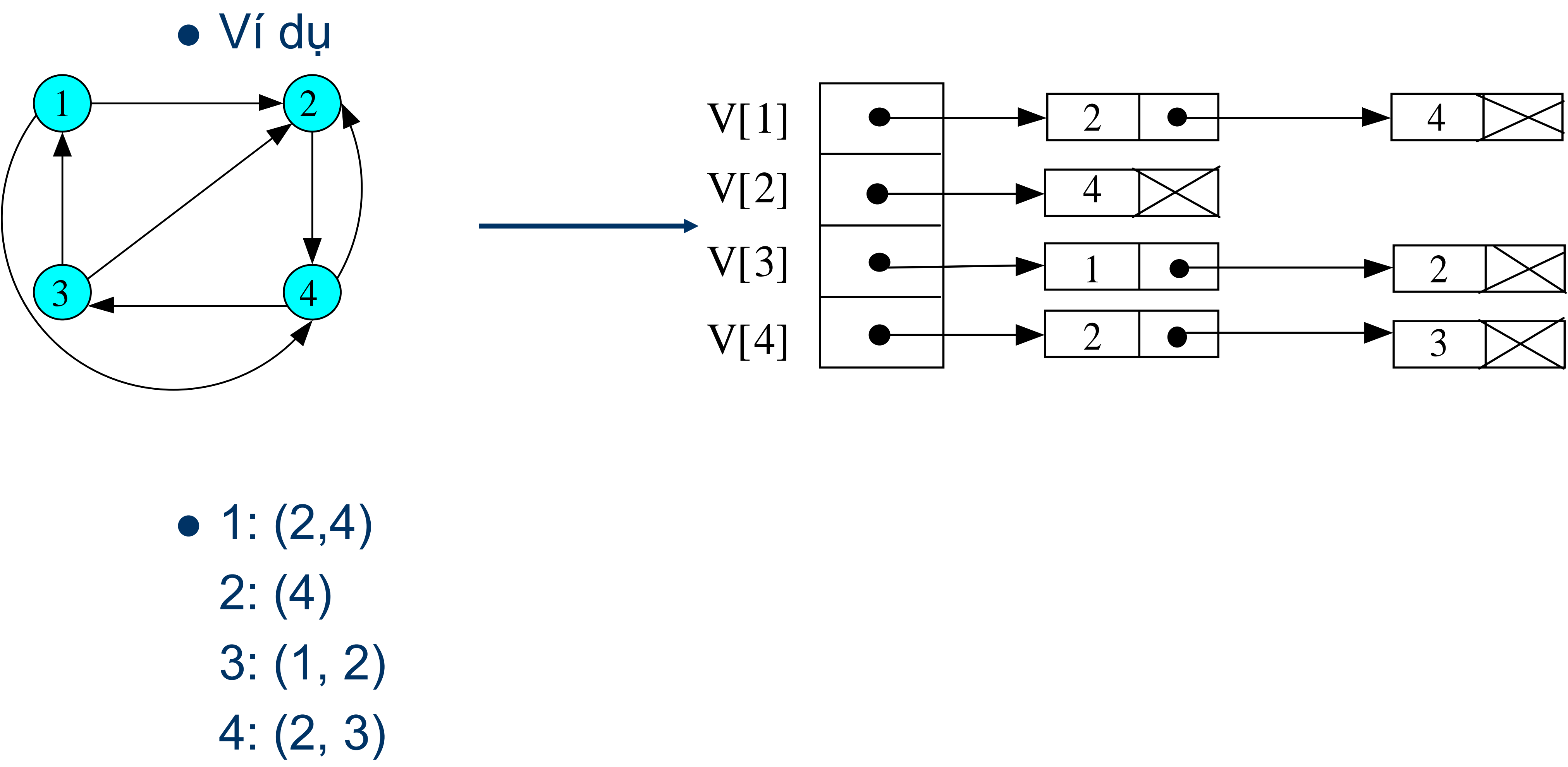
- Biểu diễn bằng danh sách lân cận
  - Mỗi đỉnh trong đồ thị sẽ ứng với một danh sách móc nối chứa các đỉnh lân cận của nó
  - Mỗi nút trong danh sách có quy cách

VERTEX	LINK
--------	------

    - VERTEX chứa giá trị tương ứng với số thứ tự của đỉnh lân cận
    - LINK chứa con trỏ trỏ tới nút tiếp theo trong danh sách
  - Mỗi danh sách như vậy có một nút đầu danh sách
  - Các nút đầu này là các phần tử của một vector  $V$  có kích thước  $n$ . Phần tử  $V[i]$  ứng với danh sách lân cận của nút thứ  $i$

Đỗ Bích Diệp - Khoa CNTT

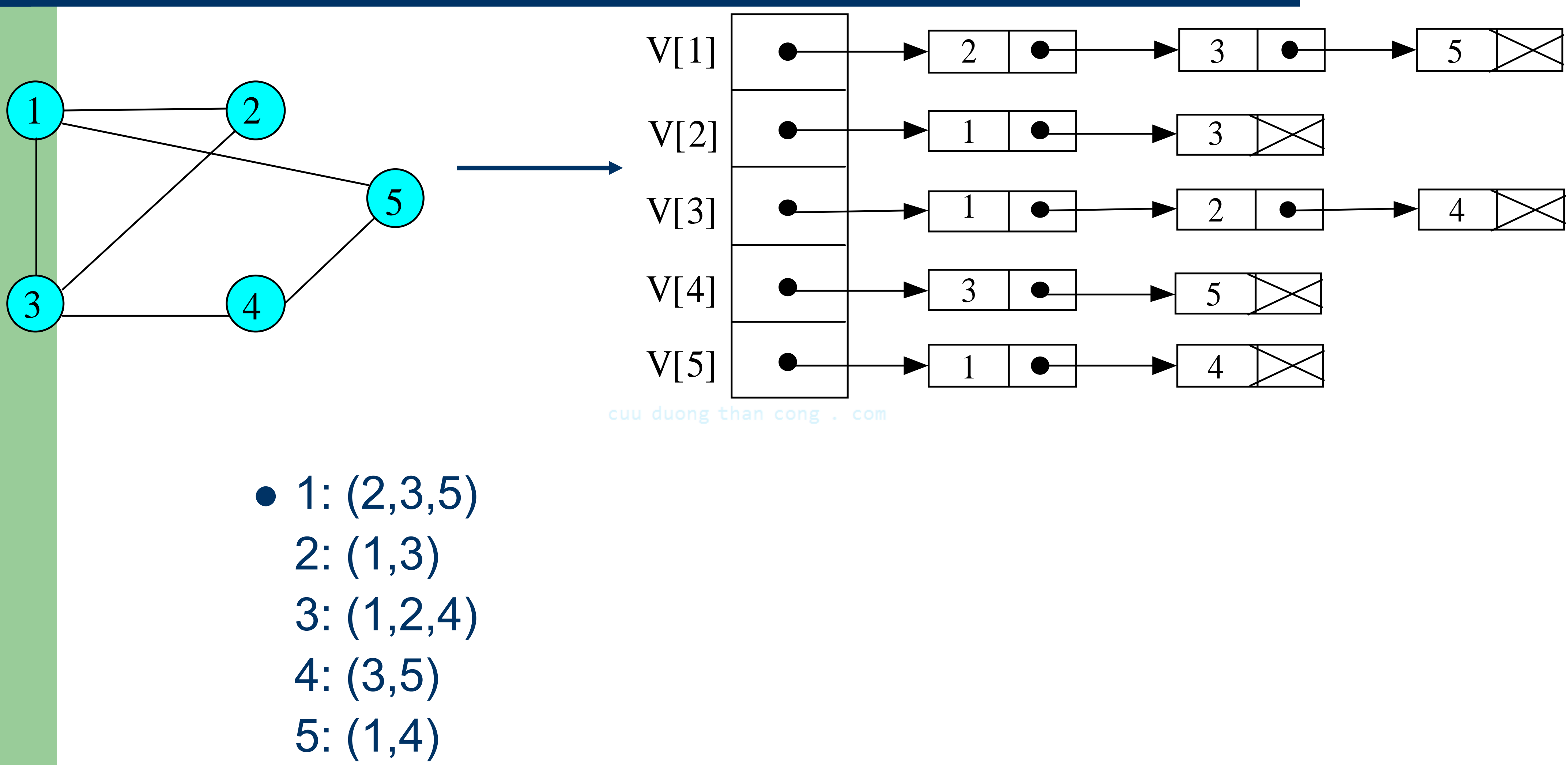
Biểu diễn đồ thị bằng danh sách lân cận



Đỗ Bích Diệp - Khoa CNTT

[cuuduongthancong.com](http://cuuduongthancong.com)

Biểu diễn đồ thị bằng danh sách lân cận



Đỗ Bích Diệp - Khoa CNTT



# Phép duyệt đồ thị

- Cho một đồ thị  $G(V,E)$  và một đỉnh  $v$  thuộc  $V$ . Duyệt đồ thị là thăm mọi đỉnh liên thông với  $v$ 
  - Có 2 phương pháp
    - Phương pháp duyệt theo chiều sâu (Depth First Search)
    - Phương pháp duyệt theo chiều rộng ( Breadth First Search)

Đỗ Bích Diệp - Khoa CNTT

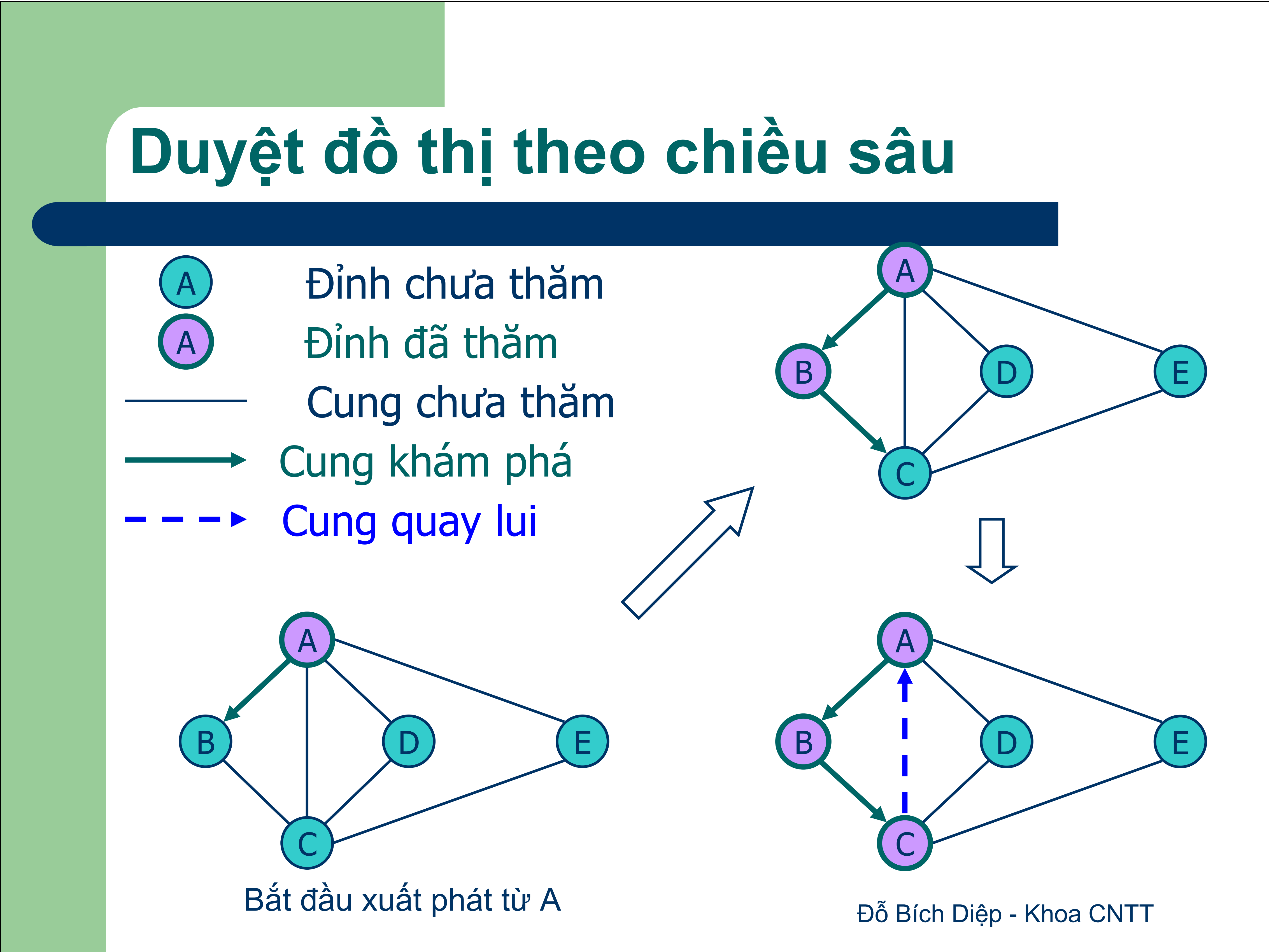
cuu duong than cong . com

# Duyệt theo chiều sâu

```
Algorithm DFS( $G, v$ )
Input đồ thị  $G$  và đỉnh bắt đầu duyệt  $v$  trong  $G$ 
Output đánh dấu các cung trong  $G$  trong phần đồ thị liên thông với đỉnh  $v$ 
thành hai loại cung khám phá (discovery edges) và cung quay lui (back
edges)

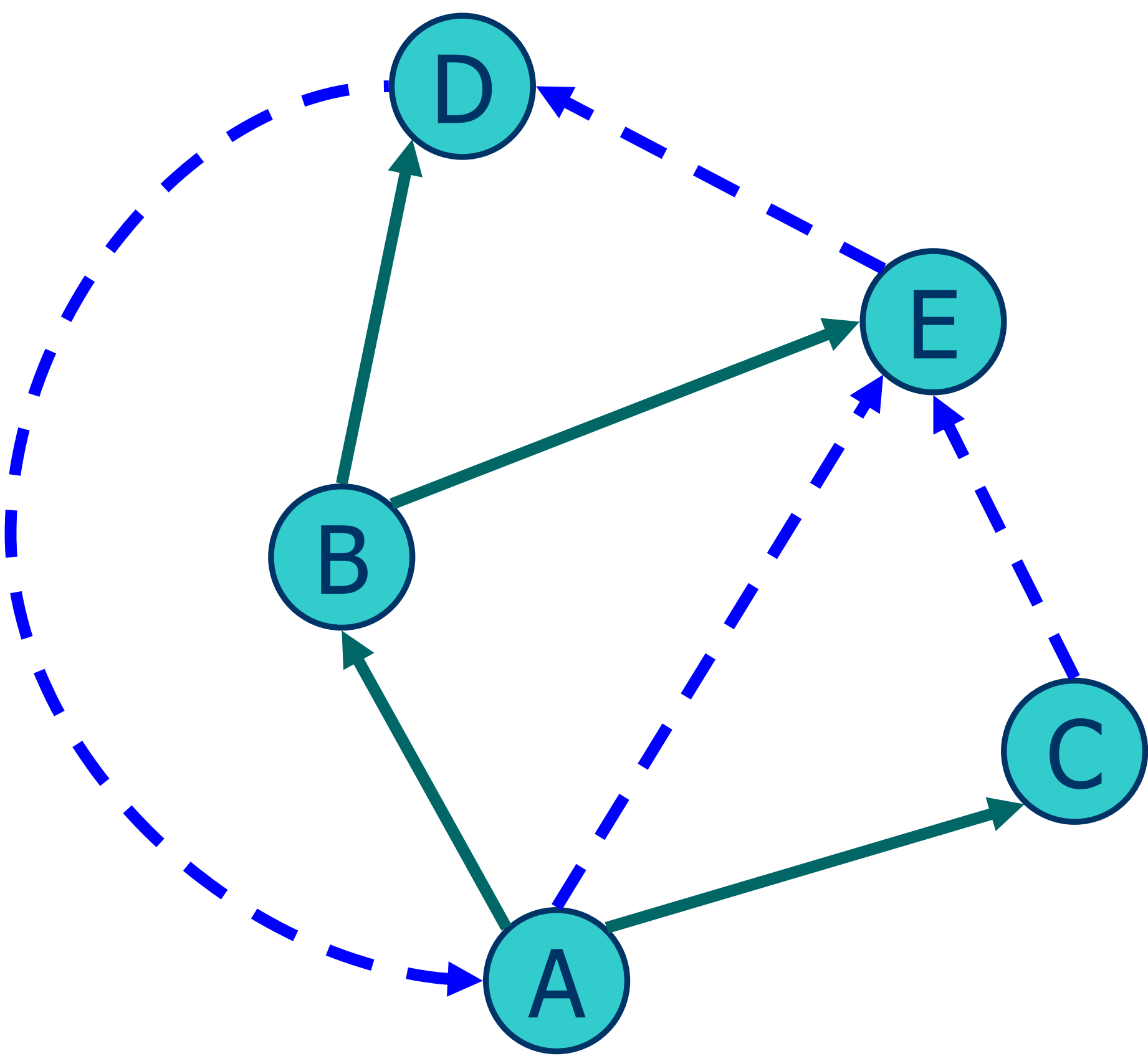
setLabel( $v, VISITED$ ) // đỉnh  $v$  đã được thăm
for all  $e \in G.incidentEdges(v)$ 
    if getLabel( $e$ ) = UNEXPLORED
         $w \leftarrow opposite(v,e)$ 
        if getLabel( $w$ ) = UNEXPLORED
            setLabel( $e, DISCOVERY$ )
            DFS( $G, w$ )
        else
            setLabel( $e, BACK$ )
```

Đỗ Bích Diệp - Khoa CNTT



## Duyệt đồ thị theo chiều sâu

- Duyệt theo chiều sâu trên đồ thị có hướng
  - Đi theo chiều của các cung trên đồ thị



Đỗ Bích Diệp - Khoa CNTT

[cuuduongthancong.com](http://cuuduongthancong.com)

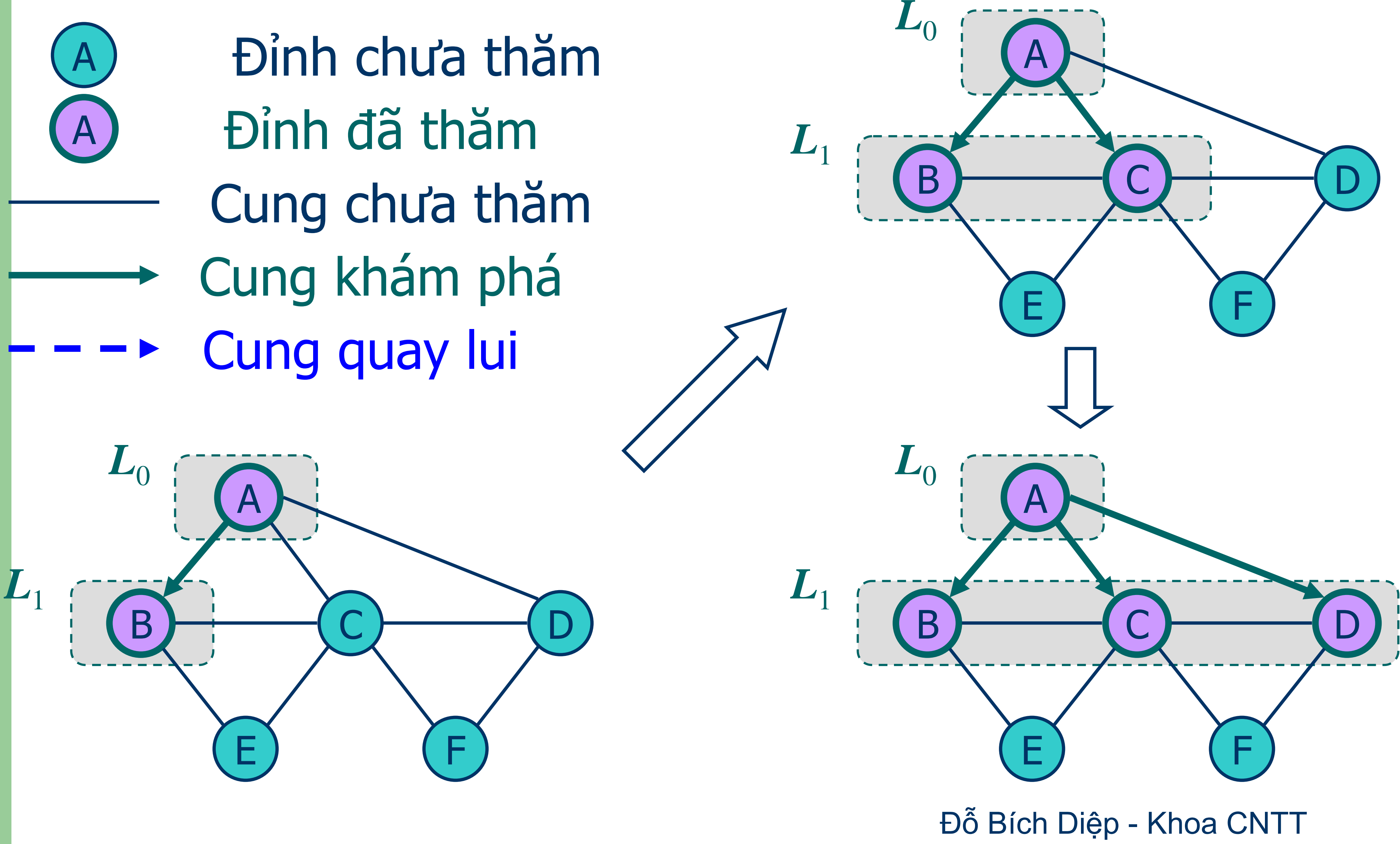
## Duyệt đồ thị theo chiều rộng

```
Algorithm BFS(G, s)
Q = một queue rỗng
Q.enqueue(s)
setLabel(s, VISITED)
while not Q.isEmpty()
    v = Q.dequeue()
    for all e ∈ G.incidentEdges(v)
        if getLabel(e) = UNEXPLORED
            w ← opposite(v,e)
            if getLabel(w) = UNEXPLORED
                setLabel(e, DISCOVERY)
                setLabel(w, VISITED)
                Q.enqueue(w)
            else
                setLabel(e, BACK)
```

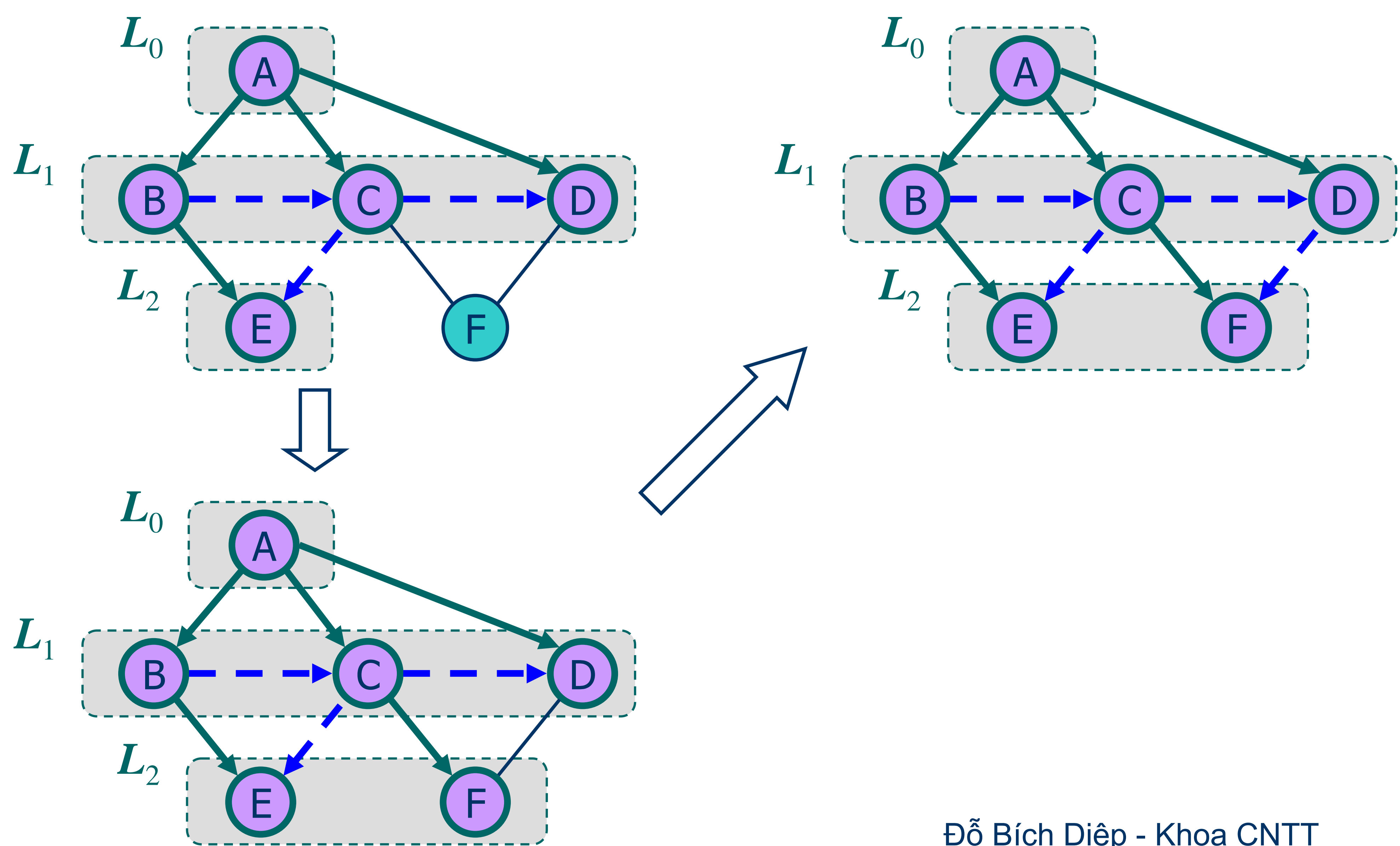
Đỗ Bích Diệp - Khoa CNTT



Duyệt đồ thị theo chiều rộng



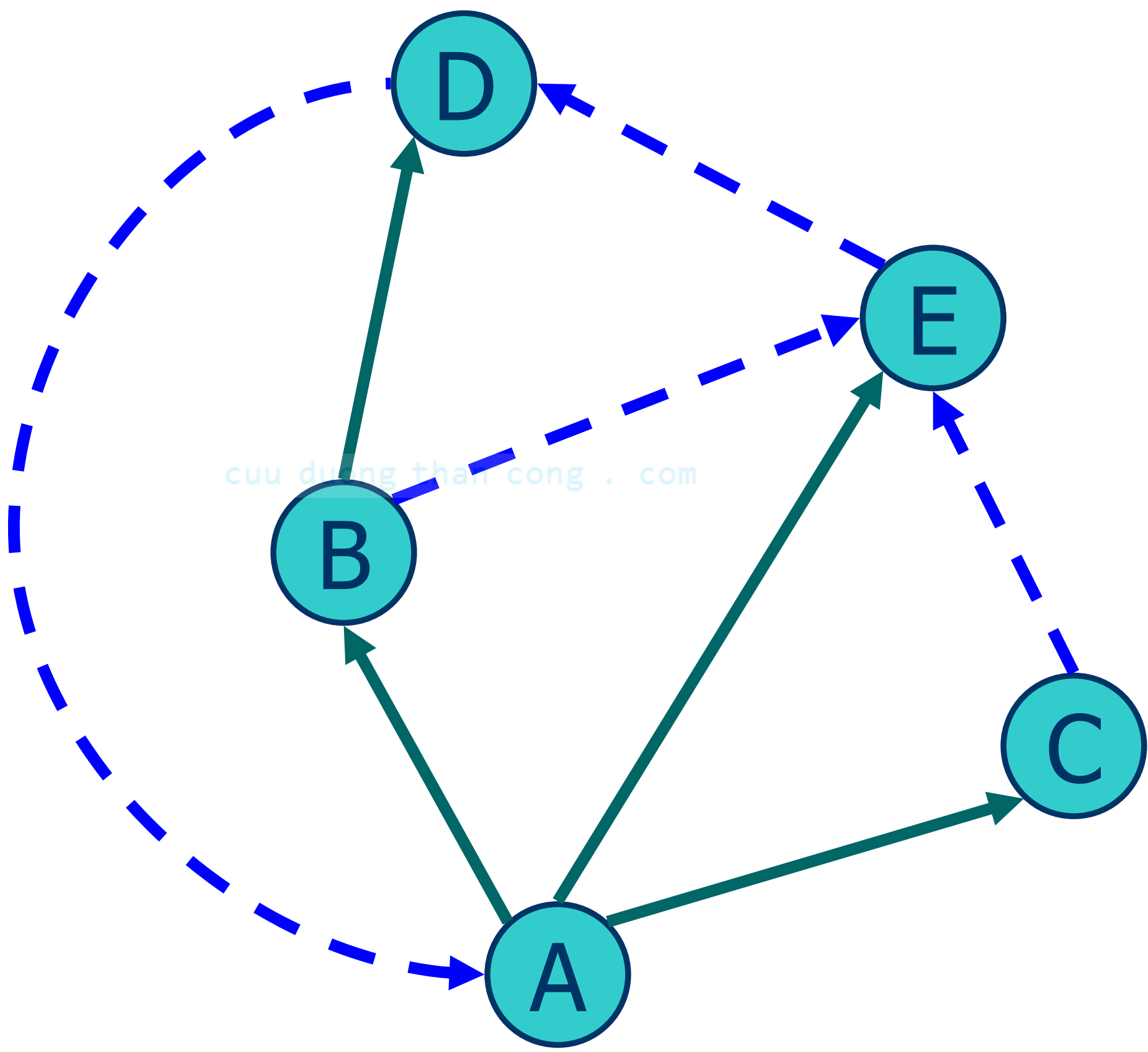
## Duyệt đồ thị theo chiều rộng



cuu duong than cong . com

## Duyệt đồ thị theo chiều sâu

- Duyệt đồ thị theo chiều rộng trên đồ thị có hướng



Đỗ Bích Diệp - Khoa CNTT