



TRƯỜNG ĐẠI HỌC MỞ TP. HCM

Cơ hội học tập cho mọi người

Khoa Công Nghệ Thông Tin

Chương 5 **ĐỒ THỊ** (GRAPH)

GV: Lê Ngọc Hiếu

TP.HCM - Tháng 06 -2019



Mở đầu





Kiến thức cần thiết khi tìm hiểu về ĐỒ THỊ:

- Các CTDL cơ bản, các phương pháp cơ bản về xếp thứ tự và tìm kiếm trên LIST; Cây nhị phân tìm kiếm;
- Kiểu dữ liệu cơ bản, dữ liệu lưu trữ trong máy tính.
- Các kiến thức về cơ sở lập trình & kỹ thuật lập trình.

Kỹ năng cần có:

- Có thể sử dụng Visual Studio 2010
- Có thể lập trình C++

Mục tiêu dạy học

-  Nắm được khái niệm “**Đồ thị**”.
-  Biết cách **biểu diễn** đồ thị trên máy tính
-  Bài toán **duyet** đồ thị
-  Bài toán **tìm kiếm** trên đồ thị

Nội dung chính

5.1 Các khái niệm

- Đồ thị
- Một số khái niệm liên quan

5.2 Một số phương pháp biểu diễn đồ thị trên máy tính

5.3 Một số phương pháp duyệt đồ thị

5.4 Một số phương pháp tìm kiếm trên đồ thị

5.5 Tổng kết chương

5.6 Bài tập chương 5

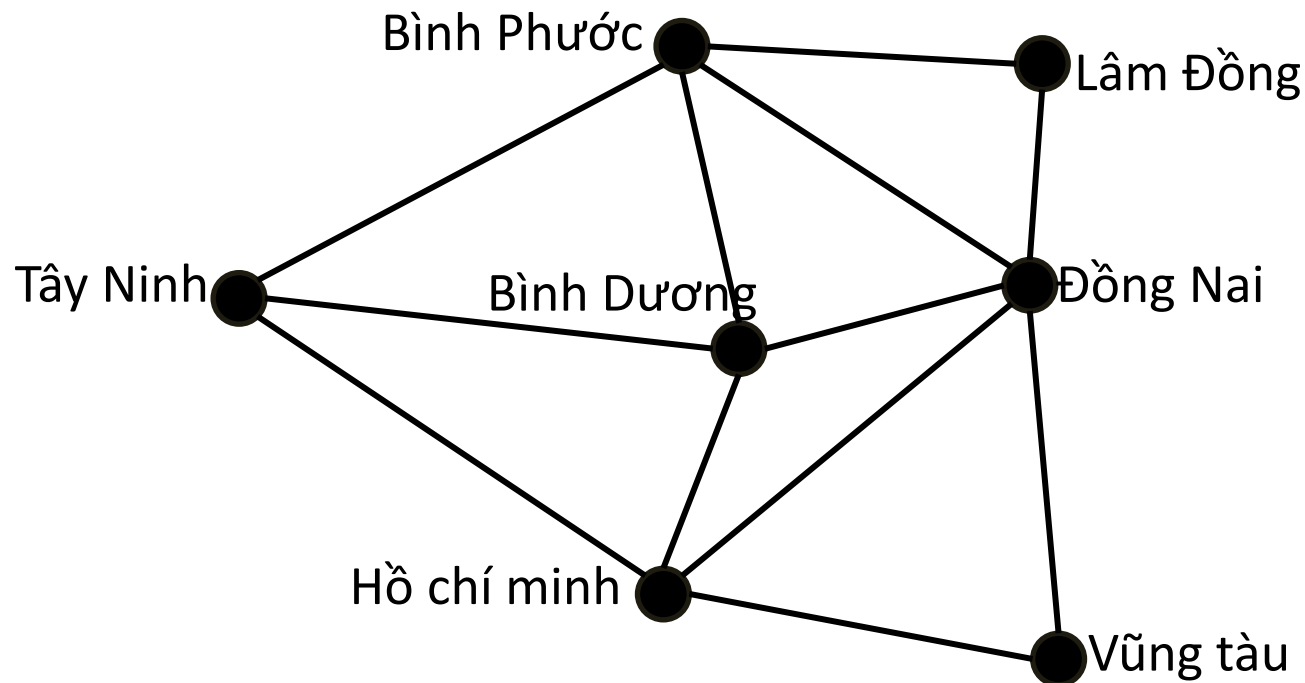
Tài liệu tham khảo

The background of the slide features a complex network diagram. It consists of numerous circular nodes, some colored in blue, green, and yellow, connected by a web of thin, light gray lines. The nodes are distributed across the slide, with some forming dense clusters and others being more isolated. The overall effect is a sense of interconnectedness and data flow.

5.1 CÁC KHÁI NIỆM


5.1 – CÁC KHÁI NIỆM ĐỒ THỊ (GRAPH)

BIỂU ĐỒ



5.1 – CÁC KHÁI NIỆM ĐỒ THỊ (GRAPH)

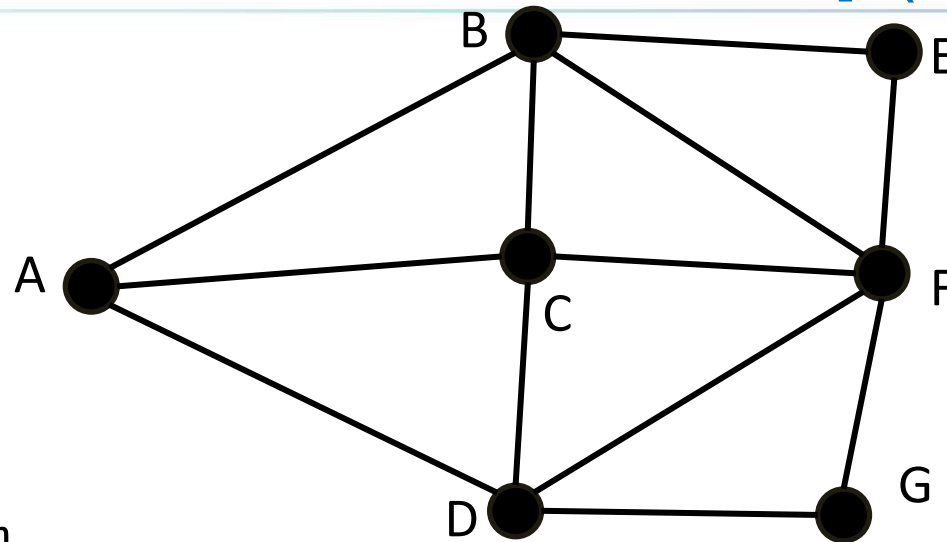
ĐỊNH NGHĨA

 Đồ thị (**Graph**) $G = (V, E)$ là một bộ gồm 2 thành phần:

- Các phần tử của V gọi là các **đỉnh** (**Vertex**) ($V \neq \emptyset$),
- Các phần tử của E gọi là các **cạnh** (**Edge**), mỗi cạnh tương ứng với 2 đỉnh.

5.1 – CÁC KHÁI NIỆM ĐỒ THỊ (GRAPH)

Ví dụ 5.1



$$V = \{A, B, C, D, E, F, G\}$$


$$E = \{ (A, B), (A, C), (A, D), (B, C), (B, E), (B, F), (C, F), (C, D), (D, F), (D, G) \}$$

5.1 – CÁC KHÁI NIỆM

MỘT SỐ KHÁI NIỆM LIÊN QUAN

MỘT SỐ KHÁI NIỆM LIÊN QUAN

ĐỈNH KỀ

 Trong đồ thị G , nếu cạnh $e \in E$, e nối 2 đỉnh u, v ($u, v \in V$) thì ta nói u và v là hai đỉnh kề với nhau, kí hiệu $e(u, v)$ hoặc $e = \overline{uv}$

Ví dụ: Cho đồ thị $G = (V, E)$ (như hình 6.1)


$V = \{A, B, C, D, E, F, G\}$

$E = \{ (A, B), (A, C), (A, D), (B, C), (B, E), (B, F), (C, F), (C, D), (D, F), (D, G) \}$

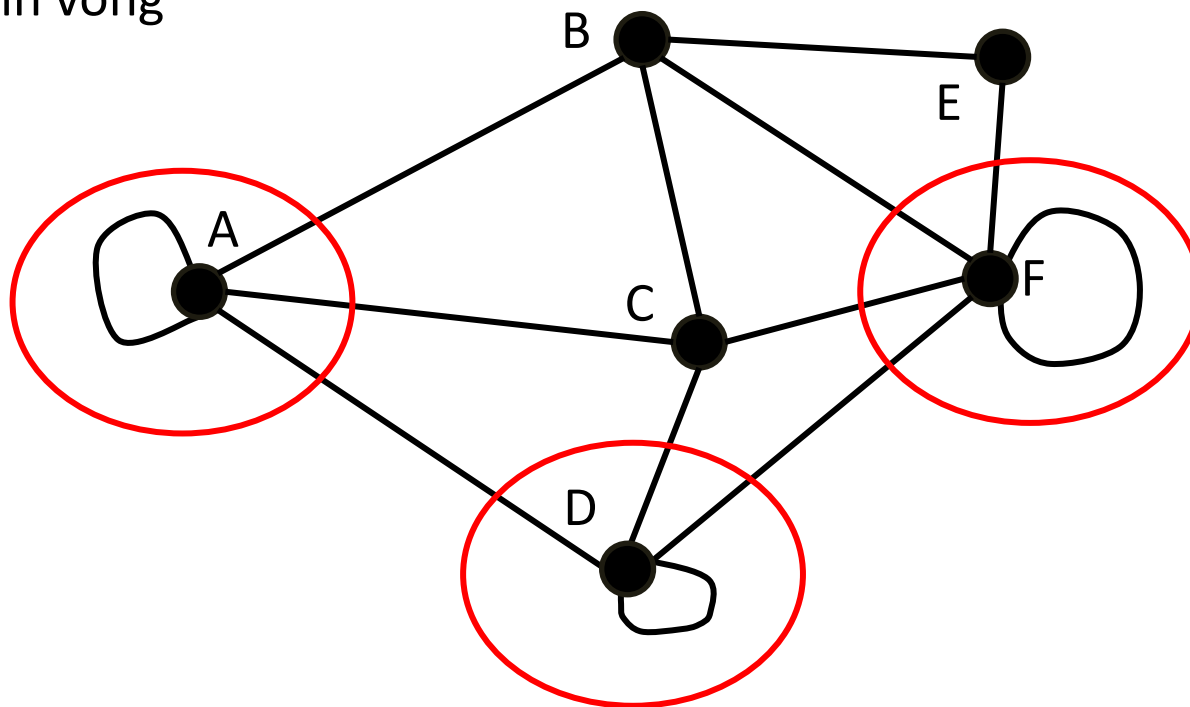
Ta nói **A** kề **B**, **B** kề **A**, **A** kề **C**, **C** kề **A**, **A** kề **D**, **D** kề **A**, ..., **D** kề **G**, **G** kề **D**;

MỘT SỐ KHÁI NIỆM LIÊN QUAN

CẠNH VÒNG


 Trong đồ thị G , nếu cạnh $e \in E$, e nối 2 đỉnh u, u' (u trùng với u') thì ta nói e là cạnh vòng

Ví dụ 5.2:

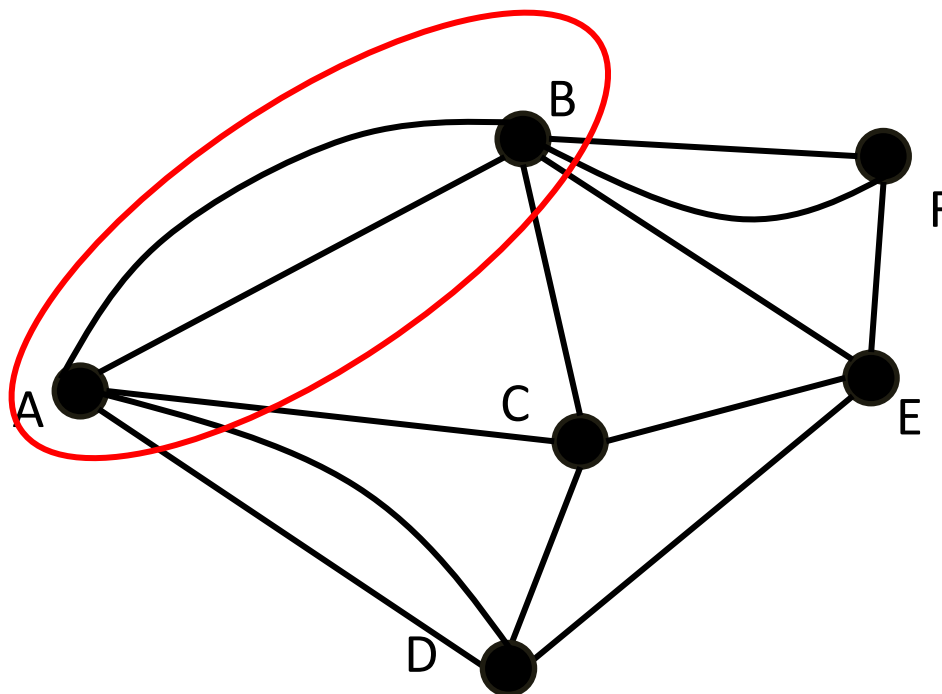


MỘT SỐ KHÁI NIỆM LIÊN QUAN

CẠNH SONG SONG


 Trong đồ thị G , nếu cạnh $e_1, e_2 \in E$, e_1 nối 2 đỉnh u, v , e_2 cũng nối hai đỉnh u, v thì ta nói e_1, e_2 là 2 cạnh song song

Ví dụ 5.3:



MỘT SỐ KHÁI NIỆM LIÊN QUAN

BẬC CỦA ĐỈNH

 Trong đồ thị G , xét một đỉnh $v \in V$, ta nói bậc của đỉnh v được xác định bằng số cạnh tới v , trong đó mỗi vòng (loop) tại v được tính là 2 cạnh tới v , kí hiệu $d(v)$

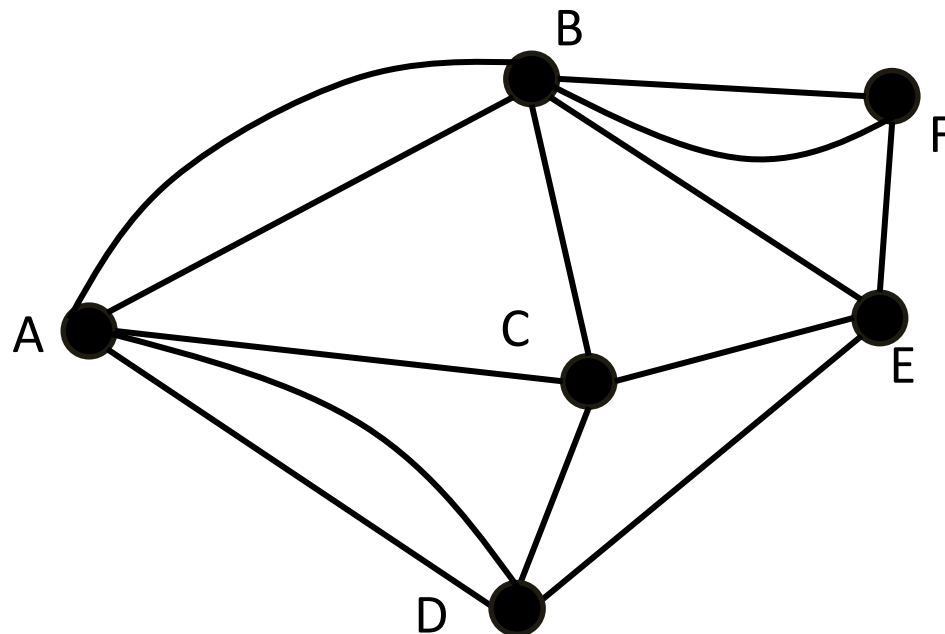
Ví dụ : xét lại ví dụ 5.3

$$d(A) = 5;$$

$$d(B) = 6;$$


...

$$d(F) = 4;$$



MỘT SỐ KHÁI NIỆM LIÊN QUAN

ĐỒ THỊ CÓ TRỌNG SỐ

 Cho $G = (V, E)$, nếu $\forall e \in E$, e được đặt tương ứng với một trọng số (**weight**), kí hiệu $w(e)$, thì ta G là một **đồ thị có trọng số**.

Ví dụ 5.7: Cho $G = (V, E)$

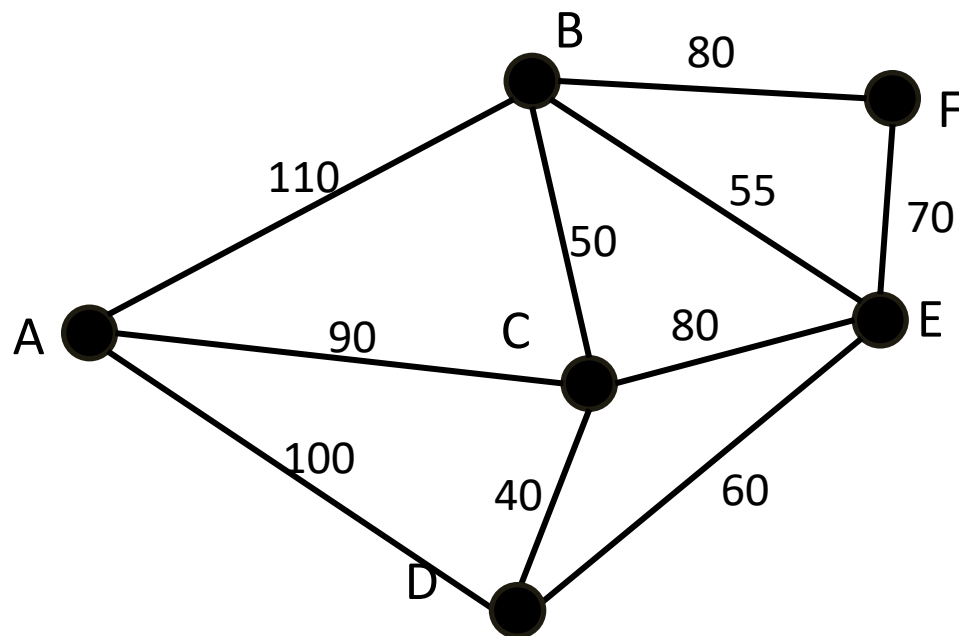
$V = \{A, B, C, D, E, F\}$

$E = \{(A,B), (A,C), (A,D), \dots\}$

$w(A,B) = 110$

$w(A,C) = 90$

....



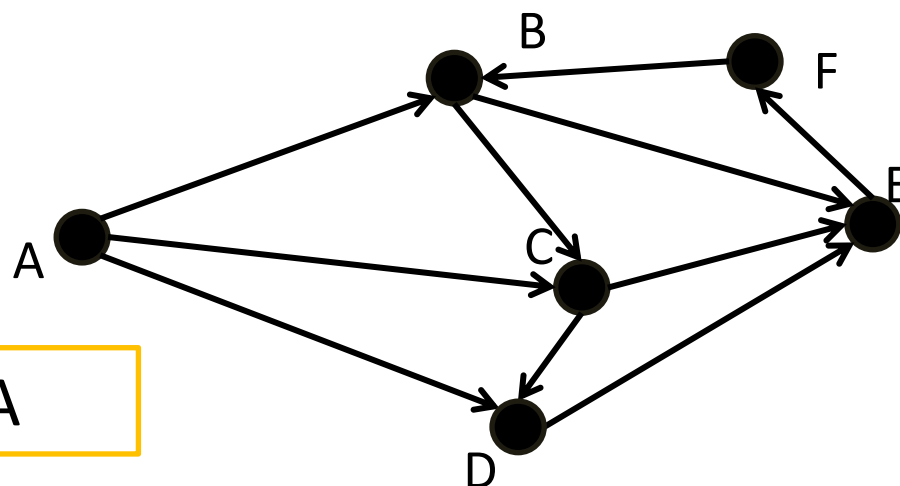
MỘT SỐ KHÁI NIỆM LIÊN QUAN

ĐỒ THỊ CÓ HƯỚNG



Cho đồ thị $G = (V, E)$, nếu mỗi cạnh $e \in E$ được cho tương ứng với một **cặp thứ tự** (u, v) của 2 đỉnh $u, v \in V$ thì ta nói e là một cạnh có hướng từ u đến v , và G được gọi là đồ thị có hướng.

Ví dụ: 5.6



A kề B

B không kề A

MỘT SỐ KHÁI NIỆM LIÊN QUAN

Một số loại đồ thị khác



Đồ thị đơn giản (Simple graph)



Đa đồ thị (Multiple graph)



Đồ thị có hướng (Directed Graph)



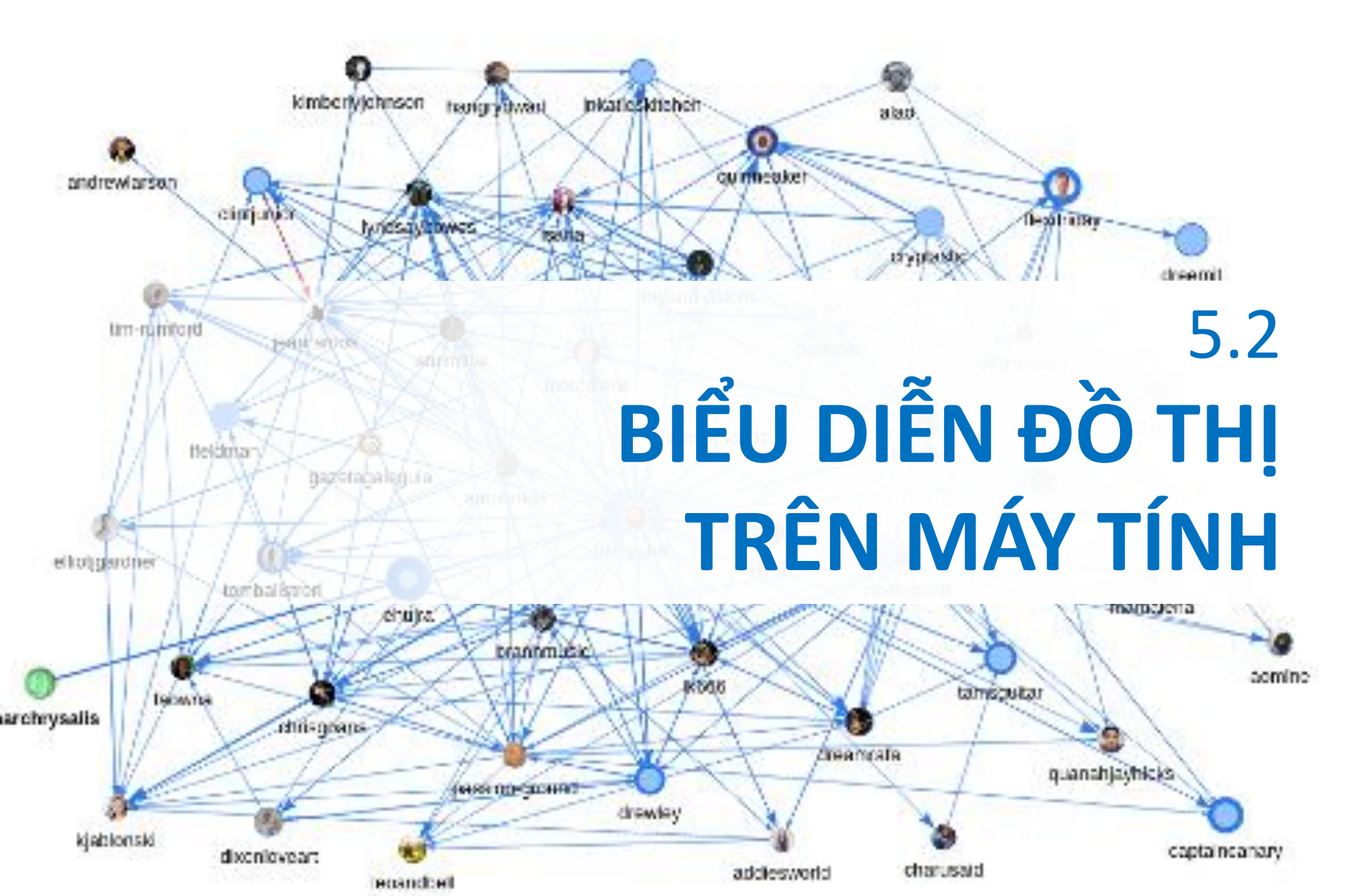
Đồ thị có hướng có trọng số (Weighted Directed Graph)



Đồ thị hỗn hợp (Mixed Graph)



....



5.2 – BIỂU DIỄN ĐỒ THỊ





Ma Trận kể



Danh sách kể

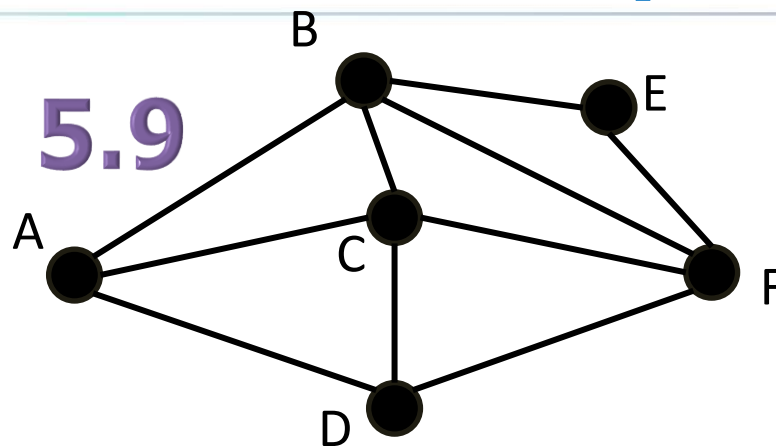
5.2 – BIỂU DIỄN ĐỒ THỊ MA TRẬN KỀ

-  Cho đồ thị $G = (V, E)$ vô hướng không có trọng số, ta đánh các số các đỉnh của đồ thị bằng một số tự nhiên: $1, 2, \dots, n$.
Xây dựng ma trận vuông biểu diễn đồ thị như sau:
-  Ma trận vuông $A_{n \times n}$ được gọi là ma trận kề của G sao cho

$$A_{[i,j]} = \begin{cases} 1 & (\text{nếu } i \text{ kề với } j) \\ 0 & (i \text{ không kề } j) \end{cases}$$

5.2 – BIỂU DIỄN ĐỒ THỊ MA TRẬN KẼ

Ví dụ: 5.9



	A	B	C	D	E	F
A	0	1	1	1	0	0
B	1	0	1	0	1	1
C	1	1	0	1	0	1
D	1	0	1	0	0	1
E	0	1	0	0	0	1
F	0	1	1	1	1	0

5.2 – BIỂU DIỄN ĐỒ THỊ **MA TRẬN KỂ**

CÀI ĐẶT

5.2 – BIỂU DIỄN ĐỒ THỊ MA TRẬN KỀ

KHAI BÁO CẤU TRÚC MA TRẬN KỀ

```
#define MAX 20  
int A[MAX][MAX]; // mảng hai chiều  
int n; // số đỉnh của đồ thị
```

5.2 – BIỂU DIỄN ĐỒ THỊ MA TRẬN KẼ

KHỞI TẠO MẢNG RỖNG

```
void init()  
{  
    n=0;  
}
```

5.2 – BIỂU DIỄN ĐỒ THỊ MA TRẬN KẼ

NHẬP MA TRẬN

```
void input()
{
    cout<<"nhap so dinh do thi n: ";
    cin>>n;
    for(int i=0;i<n;i++)
    {
        cout<<"nhap vao dong thu "<<i+1<<": ";
        for(int j=0;j<n;j++)
            cin>>A[i][j];
    }
}
```





5.2 – BIỂU DIỄN ĐỒ THỊ MA TRẬN KẼ

XUẤT MA TRẬN

```
void output()  
{  
    for(int i=0;i<n;i++)  
    {  
        for(int j=0;j<n;j++)  
            cout<<A[i][j]<<" ";  
        cout<<endl;  
    }  
}
```

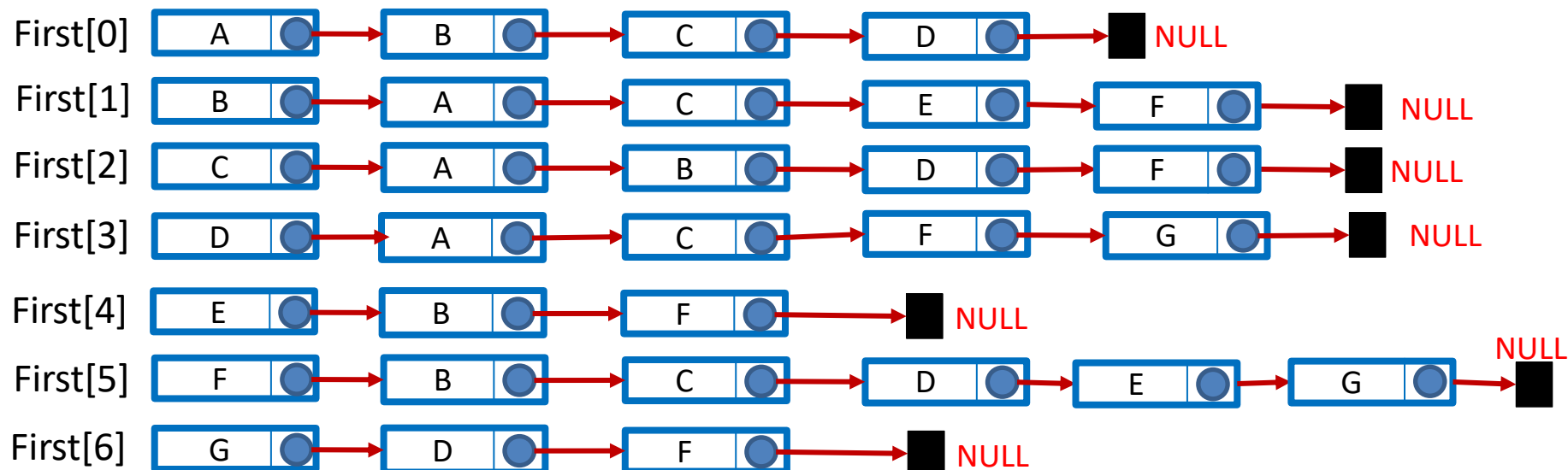
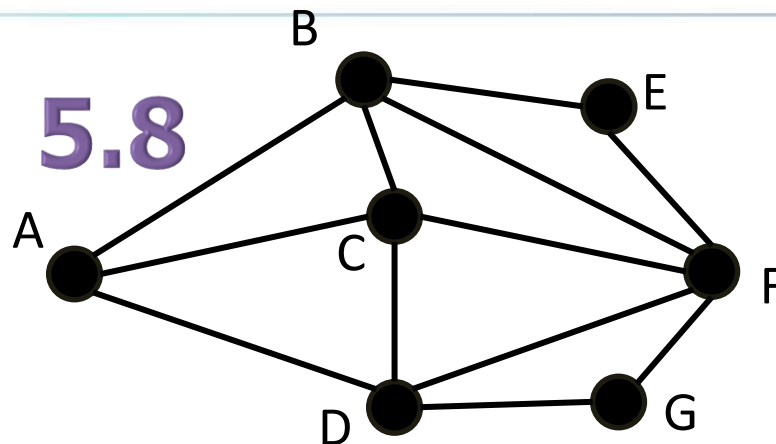
5.2 – BIỂU DIỄN ĐỒ THỊ DANH SÁCH KẼ

DANH SÁCH KẼ

-  Với mỗi đỉnh **u** của đồ thị ta sẽ xây dựng **một danh sách** (danh sách liên kết đơn).
-  Mỗi danh sách gồm phần tử đầu tiên là các đỉnh **u** (các đỉnh đồ thị), các phần tử trong danh sách là các đỉnh **v** (**u** kề **v**).
-  **Một đồ thị** được biểu diễn bằng một mảng danh sách kề.

5.2 – BIỂU DIỄN ĐỒ THỊ DANH SÁCH KÈ

Ví dụ: 5.8



5.2 – BIỂU DIỄN ĐỒ THỊ **DANH SÁCH KẼ**

CÀI ĐẶT

5.2 – BIỂU DIỄN ĐỒ THỊ DANH SÁCH KÈ

KHAI BÁO CẤU TRÚC CHO MỘT MẢNG DANH SÁCH KÈ

```
#define MAX 20
struct Node
{
    int info;
    Node *link;
};
Node *first[MAX]; // mảng danh sách
int n; // số đỉnh trên đồ thị
```

5.2 – BIỂU DIỄN ĐỒ THỊ DANH SÁCH KÈ

KHỞI TẠO MẢNG DANH SÁCH

```
void init()  
{  
    for(int i=0;i<n;i++)  
        first[i] = NULL;  
}
```

5.2 – BIỂU DIỄN ĐỒ THỊ DANH SÁCH KÈ

THÊM MỘT PHẦN TỬ VÀO DANH SÁCH

```
void insert_first(Node *&f, int x)
{
    Node *p;
    p = new Node;
    p->info = x;
    p->link = f;
    f = p;
}
```

5.2 – BIỂU DIỄN ĐỒ THỊ DANH SÁCH KÈ

NHẬP MẢNG DANH SÁCH GỒM 'n' DANH SÁCH

```
void input()
{
    int d,x, m;
    cout<<"nhap so dinh do thi n: ";
    cin>>n;
    for(int i=0;i<n;i++)
    {
        cout<<"\nnhap dinh thu
        "<<i+1<<" : ";
        cin>>d;
        insert_first(first[i],d);
    }
}
```

```
    cout<<"nhap vao so dinh ke cua
    "<<d<<" : ";
    cin>>m;
    for(int j=0;j<m;j++)
    {
        cin>>x;
        insert_first(first[i],x);
    }
}
```


5.2 – BIỂU DIỄN ĐỒ THỊ DANH SÁCH KÈ

XUẤT THÔNG TIN CỦA MỘT DANH SÁCH

```
void output_list(Node *f)
{
    if(f!=NULL)
    {
        Node * p=f;
        while (p != NULL)
        {
            cout<<p->info<<" ";
            p=p->link;
        }
    }
}
```

5.2 – BIỂU DIỄN ĐỒ THỊ DANH SÁCH KÈ

XUẤT THÔNG TIN CỦA MẢNG DANH SÁCH

```
void output()  
{  
    if(n>0)  
        for(int i=0;i<n;i++)  
        {  
            cout<<endl<<"Danh sach thu "<<i+1<<": ";  
            output_list(first[i]);  
        }  
    else  
        cout<<"rong";  
}
```




5.3

DUYỆT ĐỒ THỊ TRÊN MÁY TÍNH

5.3 – DUYỆT ĐỒ THỊ

 Breadth First Search – BFS


(Duyệt theo chiều rộng)

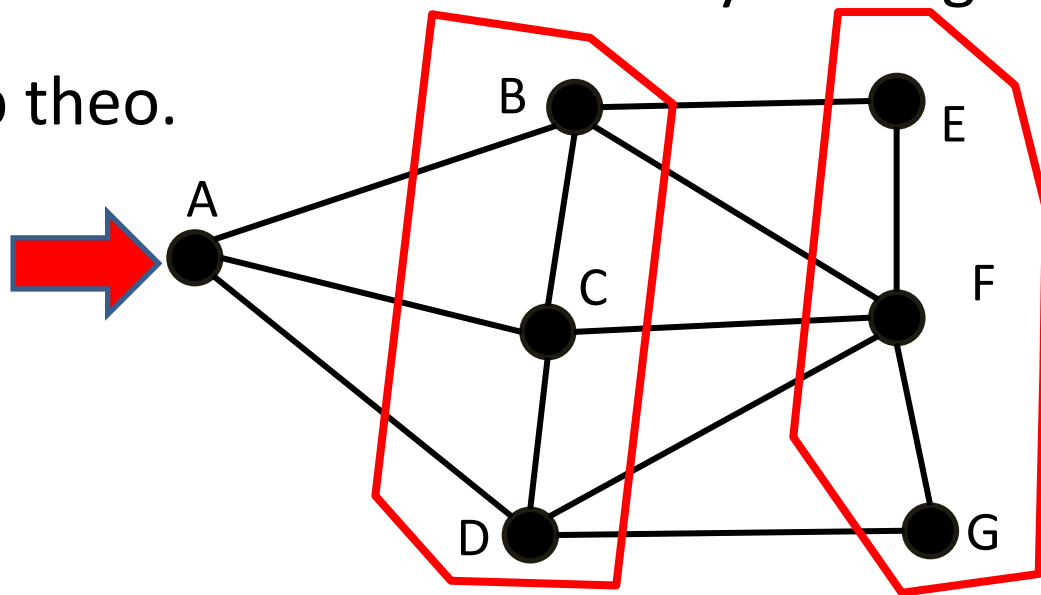
 Depth First Search – DFS (duyet
theo chiều sâu)

5.3 – DUYỆT ĐỒ THỊ **THEO CHIỀU RỘNG - BFS**

5.3 – DUYỆT ĐỒ THỊ THEO CHIỀU RỘNG - BFS

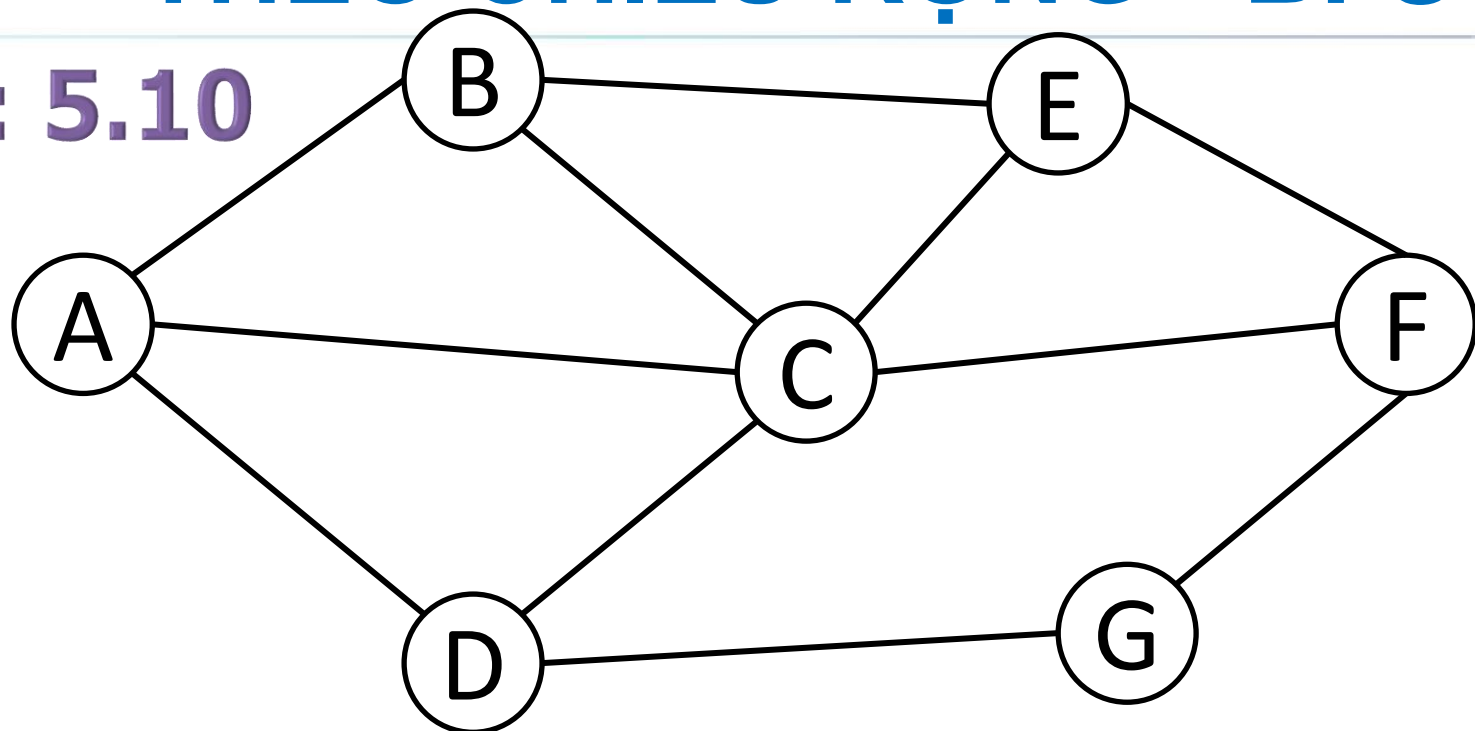
BFS (Breadth First Search)

 Duyệt trên tất cả các nút của một mức trong không gian bài toán trước khi chuyển sang các nút ở mức tiếp theo.



5.3 – DUYỆT ĐỒ THỊ THEO CHIỀU RỘNG - BFS

Ví dụ: 5.10



Kết Quả duyệt theo BFS:






5.3 – DUYỆT ĐỒ THỊ **THEO CHIỀU RỘNG - BFS**

CÀI ĐẶT

5.3 – DUYỆT ĐỒ THỊ THEO CHIỀU RỘNG - BFS

THUẬT GIẢI BFS

-  **Bước 0:** Giả sử s là đỉnh **bắt đầu**, cho s vào trong Q (Queue)
-  **Bước 1:** Nếu Q khác **rỗng**, gọi u là phần tử được lấy từ Q , và xuất u ra ngoài màn hình;
-  **Bước 2:** Tìm các đỉnh v kề với u , cho v vào trong Q quay lại bước 1.

5.3 – DUYỆT ĐỒ THỊ THEO CHIỀU RỘNG - BFS

CÀI ĐẶT PHÉP DUYỆT BFS SỬ DỤNG QUEUE

```
int C[100]; // lưu trữ đỉnh chưa xét;
// 1 là chưa xét; 0 là đã xét
int bfs[100]; // lưu danh sách phần tử đã duyệt
int nbfs=0; // chỉ số lưu đỉnh đã xét
Queue Q; // chỉ số lưu đỉnh đã xét
void BFS(int v) // v là đỉnh bắt đầu
{
    int w, p;
    push(v);
    C[v]=0;
    while(front!=NULL)
```

```
{
    pop(p);
    bfs[nbfs]=p;
    nbfs++;
    for(w=0; w<n; w++)
        if(C[w]&&a[p][w]==1)
        {
            push(w);
            C[w]=0;
        }
}
```

5.3 – DUYỆT ĐỒ THỊ THEO CHIỀU RỘNG - BFS

HÀM KHỞI TẠO ĐỈNH CHƯA XÉT

```
void khoitaochuaxet()  
{  
    for(int i=0; i<n; i++) // n là số đỉnh của đồ thị  
        C[i]=1;  
}
```

5.3 – DUYỆT ĐỒ THỊ THEO CHIỀU RỘNG - BFS




HÀM XUẤT CÁC ĐỈNH TRONG TẬP **bfs**

```
void output()  
{  
    for(int i=0; i<nbfs; i++) // n là số đỉnh của đồ thị  
        cout<<bfs[i]<<" ";  
}
```

5.3 – DUYỆT ĐỒ THỊ **THEO CHIỀU SÂU - DFS**

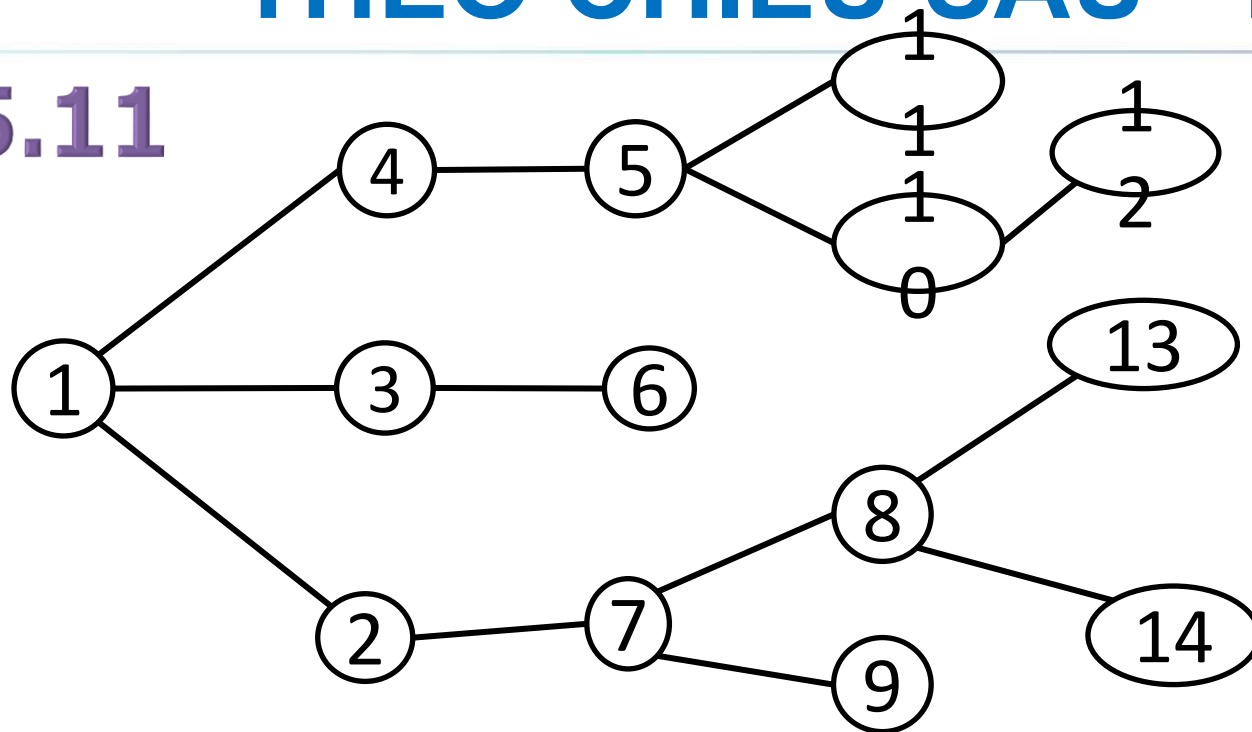
5.3 – DUYỆT ĐỒ THỊ THEO CHIỀU SÂU - DFS

Ý TƯỞNG DFS

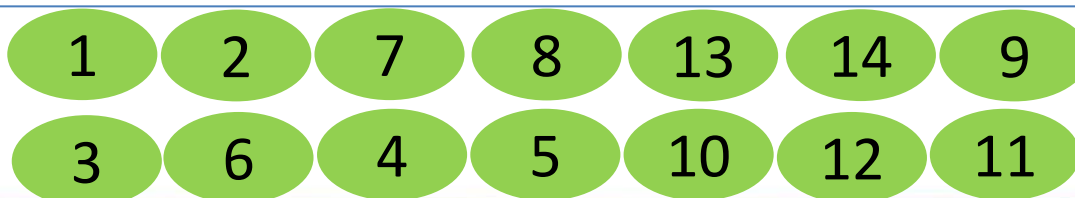
-  Bắt đầu từ một đỉnh **u**
-  Từ **u** đi theo cạnh (cung) xa nhất có thể
-  Nếu hết đường đi, trở lại đỉnh trước của cạnh xa nhất tiếp tục duyệt như trước, cho đến đỉnh cuối cùng.

5.3 – DUYỆT ĐỒ THỊ THEO CHIỀU SÂU - DFS

Ví dụ: 5.11



Kết Quả duyệt theo DFS:






5.3 – DUYỆT ĐỒ THỊ THEO CHIỀU SÂU - DFS

CÀI ĐẶT

5.3 – DUYỆT ĐỒ THỊ THEO CHIỀU SÂU - DFS

THUẬT GIẢI DFS

-  **Bước 0:** Giả sử **v** là đỉnh bắt đầu, cho v vào Stack và cho v vào tập DFS;
-  **Bước 1:** Nếu Stack chưa rỗng, thì lấy x từ trong Stack ra;
-  **Bước 2:** Tìm **u (gần nhất)** kề với **x** và **u chưa** có trong tập DFS;
 - 2.1. cho x vào trong Stack,
 - 2.2. u vào trong Stack;
 - 2.3. Cho u vào tập DFS;
 - 2.4. Sau đó quay lại bước 1;

5.3 – DUYỆT ĐỒ THỊ THEO CHIỀU SÂU - DFS

CÀI ĐẶT PHÉP DUYỆT DFS DÙNG STACK

```
int C[100]; // lưu trữ đỉnh chưa xét;  
// 1 là chưa xét; 0 là đã xét  
int dfs[100]; // lưu danh sách phần tử đã  
duyet  
int ndfs=0; // chỉ số lưu đỉnh đã xét  
STACK sta=0; // chỉ số lưu đỉnh đã xét  
void DFS(int s)  
{  
    push(s);  
    dfs[ndfs]=s;  
    ndfs++;  
    C[s]=0;  
    int v=-1, u=s;  
    while(!isEmpty())  
    {
```

```
        if(v==n)  
            pop(u);  
        for(v=0; v<n; v++)  
            if(a[u][v]!=0 &&  
C[v]==1)  
            {  
                push(u);  
                push(v);  
                dfs[ndfs]=v;  
                ndfs++;  
                C[v] = 0;  
                u=v;  
                break;  
            }  
        }  
    }
```

5.3 – DUYỆT ĐỒ THỊ THEO CHIỀU SÂU - DFS

HÀM XUẤT TẬP **dfs**

```
void output()  
{  
    for(int i=0;i<ndfs;i++)  
        cout<<dfs[i]<<" ";  
}
```

5.3 – DUYỆT ĐỒ THỊ THEO CHIỀU SÂU - DFS

HÀM KHỞI TẠO ĐỈNH CHƯA XÉT

```
void khoitaochuaxet()  
{  
    for(int i=0; i<n; i++) // n là số đỉnh của đồ thị  
        C[i]=1;  
}
```


An aerial photograph of a university campus with green lawns, trees, and buildings. Overlaid on the image is a network diagram consisting of a central light blue circle connected by white lines to five peripheral circles: orange (top-left), pink (top-right), green (left), magenta (bottom-left), and yellow (bottom-right).

5.4

TÌM KIẾM TRÊN ĐỒ THỊ

5.4 – TÌM KIẾM TRÊN ĐỒ THỊ




Tìm kiếm dựa vào Breadth First Search – BFS (Duyệt theo chiều rộng)



Tìm kiếm dựa vào Depth First Search – DFS (duyet theo chiều sâu)

5.4 – TÌM KIẾM TRÊN ĐỒ THỊ

PHÁT BIỂU BÀI TOÁN

 Cho đồ thị vô hướng đơn giản $G = (V, E)$, và một đỉnh X được nhập vào từ bàn phím. Hãy kiểm tra X có tồn tại trong G hay không?

5.4 – TÌM KIẾM TRÊN ĐỒ THỊ

MÔ HÌNH HÓA BÀI TOÁN



Đầu vào: một đồ thị G , và một đỉnh X ;



Đầu ra: TRUE/FALSE




BIỂU DIỄN BÀI TOÁN LÊN MÁY TÍNH



Ta biểu diễn đồ thị G lên máy tính bằng Ma trận kề (hoặc danh sách kề).

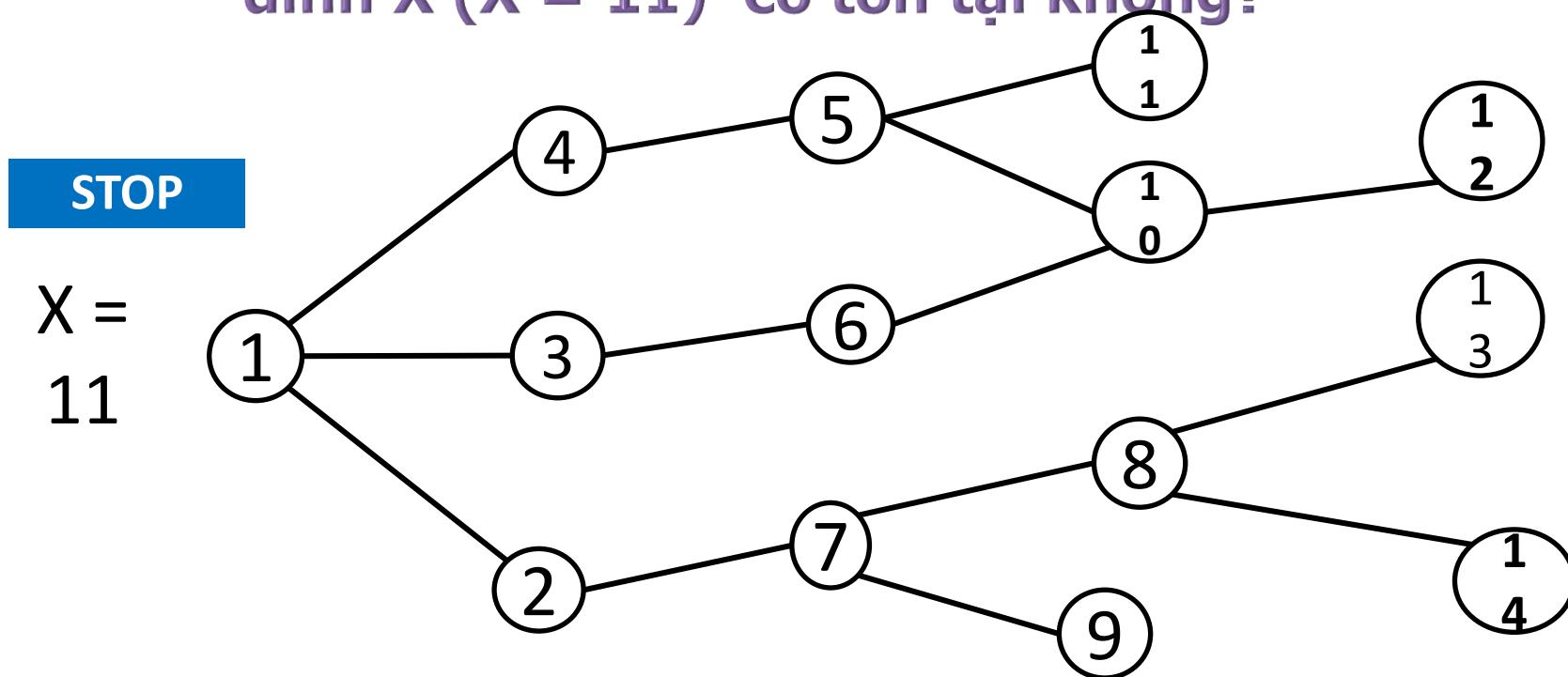
5.4 – TÌM KIẾM TRÊN ĐỒ THỊ

Ý TƯỞNG TÌM KIẾM

-  Ta thấy rằng, phương pháp duyệt đồ thị DFS, BFS sẽ cho phép ta duyệt qua tất cả các đỉnh của đồ thị.
-  Vì vậy, thay vì ta chỉ xuất ra ngoài màn hình đỉnh của đồ thị được duyệt. Mà ta có thể thực hiện lệnh kiểm X với đỉnh được duyệt.
-  Nếu đỉnh được duyệt bằng (giống giá trị) X thì ta thực hiện dừng phép duyệt và thông báo rằng X tìm thấy.

5.4 – TÌM KIẾM TRÊN ĐỒ THỊ

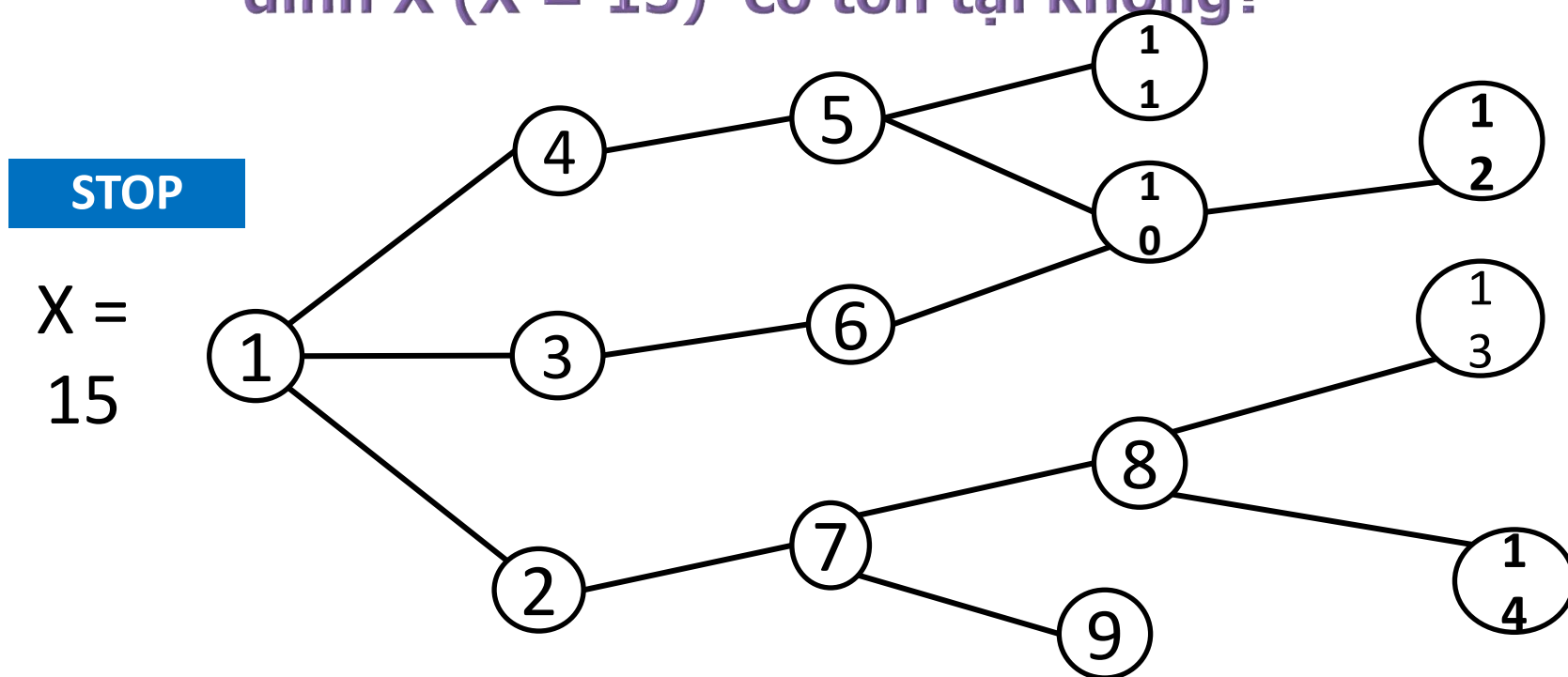
Ví dụ 5.12 cho đồ thị như hình bên dưới, hãy kiểm tra đỉnh X ($X = 11$) có tồn tại không?



Minh họa thuật giải tìm kiếm bằng phương pháp duyệt theo chiều rộng

5.4 – TÌM KIẾM TRÊN ĐỒ THỊ

Ví dụ 5.13 cho đồ thị như hình bên dưới, hãy kiểm tra đỉnh X ($X = 15$) có tồn tại không?



Minh họa thuật giải tìm kiếm bằng phương pháp duyệt theo chiều rộng

5.4 – TÌM KIẾM TRÊN ĐỒ THỊ

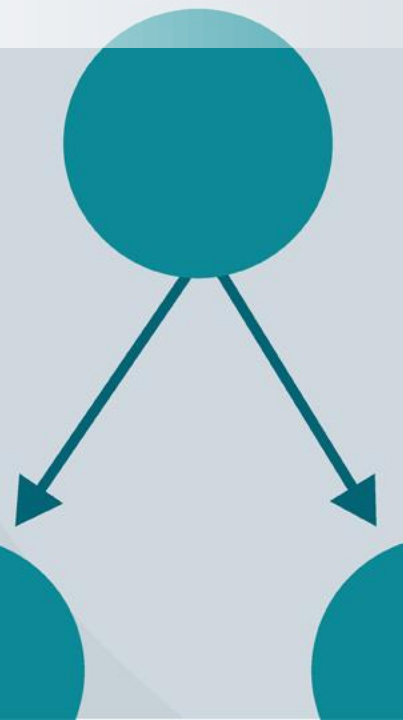
CÀI ĐẶT THUẬT TOÁN TÌM KIẾM DỰA TRÊN PHÉP DUYỆT BFS

```
int C[100]; // lưu trữ đỉnh chưa xét;





void Search_by_BFS(int x, int v) // v là
đỉnh bắt đầu
{
    int w, p;
    push(v);
    C[v]=0;
    while(front!=NULL)
    {
        pop(p);
        If (x == p)
```

```
{
    cout<<x<<"ton tai" ;
    return;
}
for(w=0; w<n; w++)
    if(C[w]&&a[p][w]==1)
    {
        push(w);
        C[w]=0;
    }
}
```

5.5 – TỔNG KẾT CHƯƠNG 5



5.5 – Tổng kết chương 5

-  **Đồ thị** và một số khái niệm liên quan
-  Một số **phương pháp biểu diễn đồ thị**
-  Một số **phương pháp duyệt đồ thị**
-  Bài toán **tìm kiếm trên đồ thị**

A person wearing a blue and white checkered shirt is sitting at a wooden desk, working on a silver laptop. The laptop screen displays a website with a car image. To the left of the laptop is an open notebook with a pen resting on it. To the right is a light blue cup of coffee on a saucer. In the background, there is a stack of books and a black mug. The background wall is made of white bricks.

5.6- Bài tập rèn luyện **CHƯƠNG 5**

5.6 - Bài tập chương 5

CÂU HỎI



Câu 1: Đồ thị là gì? Cho ví dụ? Có các loại đồ thị gì? Ngoài những đồ thị đã học, hãy tìm hiểu thêm còn có những loại đồ thị nào?



Câu 2: Để biểu diễn đồ thị trên máy tính, ta có mấy cách?

5.6 - Bài tập chương 5

BÀI TẬP LÝ THUYẾT

Bài 1: Thực hiện đếm bậc các đỉnh tại ví dụ 5.1, 5.2, 5.3, 5.6

Bài 2: Cho biết ma trận kề của đồ thị tại ví dụ 5.1, 5.6, 5.7, 5.11

Bài 3: Cho biết danh sách kề của đồ thị tại ví dụ 5.1, 5.7, 5.8, 5.11

5.6 - Bài tập chương 5

BÀI TẬP THỰC HÀNH

Bài 4: Viết chương trình với các yêu cầu sau:

- 4.1 Viết hàm nhập ma trận kề của đồ thị 5.1
- 4.2 Viết hàm xuất ma trận kề này ra ngoài màn hình.
- 4.3 Viết hàm duyệt đồ thị theo chiều rộng (dựa trên Queue bằng kỹ thuật cài đặt danh sách liên kết đơn)
- 4.4 Viết hàm duyệt đồ thị theo chiều sâu (dựa trên Stack bằng kỹ thuật cài đặt danh sách liên kết đơn)
- 4.5 Nhập vào một đỉnh X, hãy kiểm tra X có tồn tại trên đồ thị hay không? (dựa trên phép duyệt BFS)

5.6 - Bài tập chương 5

BÀI TẬP THỰC HÀNH

Bài 5: Viết chương trình với các yêu cầu sau:

- 5.1 Viết hàm nhập danh sách kề của đồ thị tại ví dụ 5.1;
- 5.2 Viết hàm xuất danh sách kề này ra ngoài màn hình.
- 5.3 Viết hàm duyệt đồ thị theo chiều rộng (dựa trên Queue bằng kỹ thuật cài đặt danh sách liên kết đơn)
- 5.4 Viết hàm duyệt đồ thị theo chiều sâu (dựa trên Stack bằng kỹ thuật cài đặt danh sách liên kết đơn)
- 5.5 Nhập vào một đỉnh X, hãy kiểm tra X có tồn tại trên đồ thị hay không? (dựa trên phép duyệt BFS)

5.6 - Bài tập chương 5

BÀI TẬP LÀM THÊM

Bài 6: Viết chương trình với các yêu cầu sau:

- 6.1 Viết hàm nhập ma trận kề của đồ thị 5.1,
- 6.2 Viết hàm xuất ma trận kề này ra ngoài màn hình.
- 6.3 Viết hàm duyệt đồ thị theo chiều rộng (dựa trên Queue bằng kỹ thuật cài đặt danh sách đặc)
- 6.4 Viết hàm duyệt đồ thị theo chiều sâu (dựa trên Stack bằng kỹ thuật cài đặt danh sách đặc)
- 6.5 Nhập vào một đỉnh X, hãy kiểm tra X có tồn tại trên đồ thị hay không? (dựa trên phép duyệt DFS)

5.6 - Bài tập chương 5

BÀI TẬP LÀM THÊM

Bài 7: Viết chương trình với các yêu cầu sau:

- 7.1 Viết hàm nhập danh sách kề của đồ thị tại ví dụ 5.1;
- 7.2 Viết hàm xuất danh sách kề này ra ngoài màn hình.
- 7.3 Viết hàm duyệt đồ thị theo chiều rộng (dựa trên Queue bằng kỹ thuật cài đặt danh sách đặc)
- 7.4 Viết hàm duyệt đồ thị theo chiều sâu (dựa trên Stack bằng kỹ thuật cài đặt danh sách đặc)
- 7.5 Nhập vào một đỉnh X, hãy kiểm tra X có tồn tại trên đồ thị hay không? (dựa trên phép duyệt DFS)

5.6 - Bài tập chương 5

BÀI TẬP LÀM THÊM

Bài 8: Biểu diễn G ở ví dụ 5.11 (bảng ma trận kề) lên trên máy tính và đặt tên là do_thi_1.txt (dạng file TEXT) sau đó thực hiện viết chương trình với các hàm sau:

8.1. Hàm đọc file do_thi_1.txt và lưu vào mảng hai chiều;

8.2. Thực hiện xuất các đỉnh đồ thị theo phép duyệt DFS (dung Queue với kỹ thuật cài đặt bằng danh sách liên kết);

8.3. Thực hiện xuất các đỉnh đồ thị theo phép duyệt BFS (dung Stack với kỹ thuật cài đặt bằng danh sách liên kết);

8.3. Thực hiện xuất các đỉnh đồ thị theo phép duyệt BFS (dung kỹ thuật đệ quy)

8.4. Thực hiện nhập vào một đỉnh x, kiểm tra x có phải là một đỉnh của đồ thị hay không?

Hướng dẫn

- Tất cả sinh viên phải trả lời các **câu hỏi** của chương, làm **bài tập thực hành** tại phòng máy (**bài làm thêm** ở nhà, và **bài nâng cao** khuyến khích hoàn tất) và nộp bài qua LMS của trường.
- Câu hỏi và bài tập lý thuyết chương 5 làm trên file WORD; trong bài làm ghi rõ họ tên, lớp, bài tập chương và các thông tin cần thiết.
- Khuyến khích sử dụng tiếng Anh trong bài tập.

⇒ **Ngày nộp:** trước khi kết thúc môn học.

⇒ **Cách nộp:** sử dụng **github** để nộp bài, sau đó nộp lên LMS của trường.

Tài liệu tham khảo

- **Thomas H.Cormen, Charles E.Leiserson, Ronald L. Rivest, Clifford Stein**, (*Chapter 22*) *Introduction to Algorithms*, Third Edition, 2009.

Phụ lục – Thuật ngữ tiếng Anh

#	Tiếng Anh	Phiên Âm	Tiếng Việt
1	Graph	/ græf /	Đồ thị
2	Representation	/ reprɪzen'teɪʃn /	Biểu diễn
3	Connectivity	/ kə:nek'tɪvɪti /	Liên thông
4	Directed Undirected	/ daɪ'rektɪd / / ʌn'daɪ'rektɪd /	Có hướng Vô hướng
5	Depth	/ depθ /	Chiều sâu
6	Breadth	/ bredθ /	Chiều rộng
7	Vertex	/ 'vɜ:teks /	Đỉnh
8	Edge	/ edʒ /	Cạnh

KẾT THÚC CHƯƠNG 5



Trường Đại học Mở TP.HCM

Khoa Công Nghệ Thông Tin