

Cấu trúc dữ liệu cho đồ thị

❖ Ma trận kề:

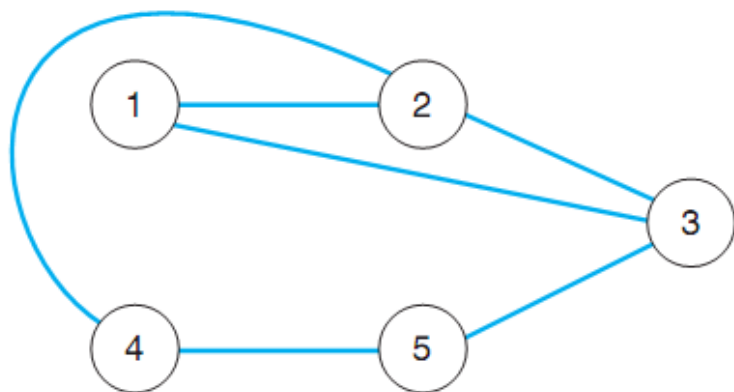
- Biểu diễn đồ thị $G=(V,E)$ bằng ma trận kề $|V|$ với N hàng, N cột với các giá trị 0,1

$$|V| = \begin{cases} 0 & \text{Nếu không tồn tại cạnh giữa } v_i v_j \\ 1 & \text{Nếu tồn tại cạnh giữa } v_i v_j \end{cases}$$

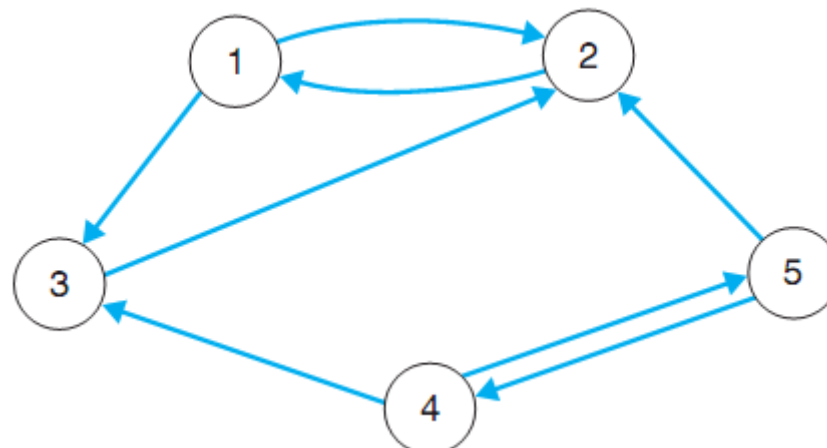
- Đồ thị có thị có trọng số: Giá trị của ma trận kề gồm trọng số của các cạnh

Cấu trúc dữ liệu cho đồ thị

❖ Ví dụ: Biểu diễn ma trận kề cho các đồ thị sau



	1	2	3	4	5
1	0	1	1	0	0
2	1	0	1	1	0
3	1	1	0	0	1
4	0	1	0	0	1
5	0	0	1	1	0

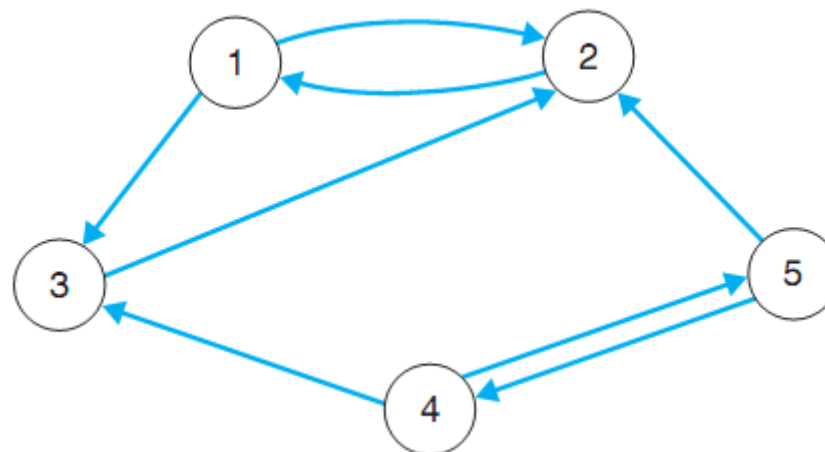
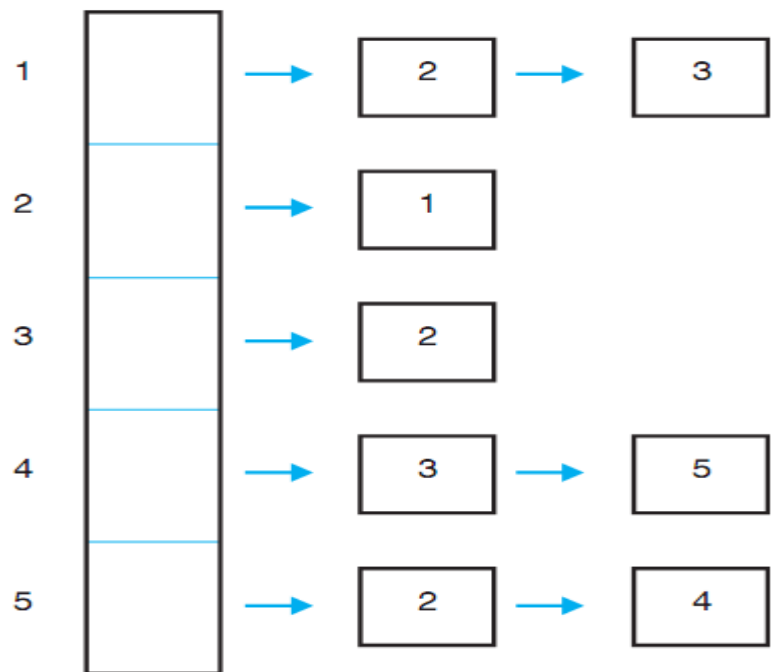


	1	2	3	4	5
1	0	1	1	0	0
2	1	0	0	0	0
3	0	1	0	0	0
4	0	0	1	0	1
5	0	1	0	1	0

Cấu trúc dữ liệu cho đồ thị

❖ Danh sách kề:

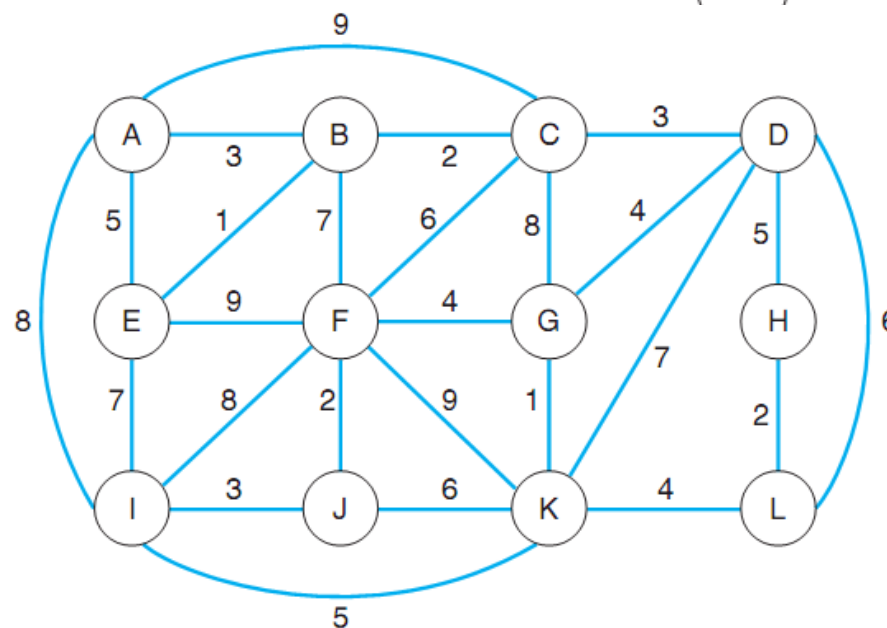
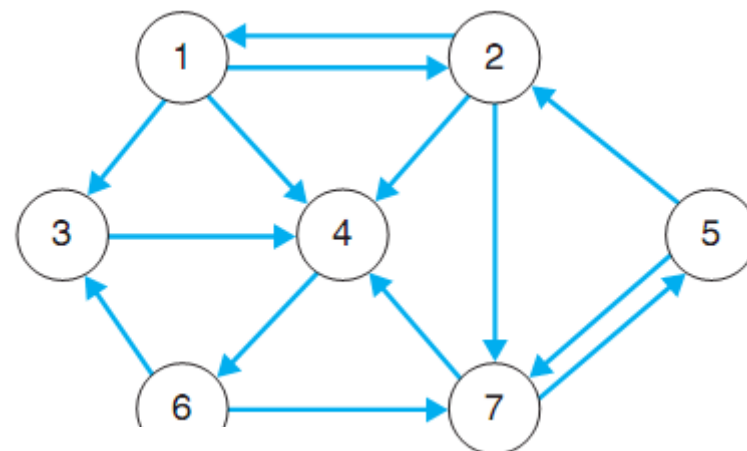
- Biểu diễn đồ thị $G=(V,E)$ bằng danh sách kề $|V|$ là một mảng 1 chiều có size N , trong đó mỗi đỉnh tương đương 1 danh sách liên kết



Cấu trúc dữ liệu cho đồ thị

❖ Bài tập:

1. Biểu diễn danh sách kề cho đồ thị 1
2. Biểu diễn ma trận kề cho đồ thị



❖ Cài đặt ma trận kề:

```
#define max 100  
struct Graph  
{  
    int n;  
    int a[max][max];  
};
```

- ## ❖ Định dạng dữ liệu:
- Dữ liệu vào ma trận kề được lưu ở file:
1. Dòng đầu tiên: số đỉnh của đồ thị
 2. Mỗi dòng chứa n số nguyên ứng với giá trị trong ma trận kề

❖ Đọc ma trận kề từ file:

```
void Matranke (Graph &g)
{ char file[128];
  printf("Tap tin nguon (Dothi.txt)");
  gets(file);
  if (strcmp(file, "")==0)
    strcpy(file, "Dothi.txt");
  FILE *f;
  f = fopen(file, "rt");
```

```
if ( f==NULL)
{   printf(“Khong mo duoc file”);
    exit(0);
}
fscanf(f,”%d”,&g.n);
for (int i=0;i<g.n;i++)
    for (int j =0;j<g.n;j++)
        fscanf (f,”%d”,&g.a[i][j]);
fclose(f);
}
```