

CHƯƠNG 6: ĐỒ THỊ (GRAPH)

DATA STRUCTURES AND ALGORITHMS

L/O/G/O

Đặt vấn đề



- Bài toán tô màu

Giả sử ta có bản đồ các quốc gia trên thế giới, ta muốn tô màu các quốc gia này sao cho 2 nước có cùng ranh giới được tô khác màu nhau.

Yêu cầu tìm cách tô sao cho số màu sử dụng là ít nhất.

ĐỊNH NGHĨA

Đồ thị (Graph)

- ✓ Là 1 cấu trúc dữ liệu gồm 2 thành phần, $G=(V,E)$:
 - V: Tập hữu hạn các đỉnh (vertex/vertices, node)
 - E: Tập hữu hạn các cạnh (edge), mỗi cạnh là 1 cặp (u,v) . Cạnh có thể chứa trọng số (weight)
- ✓ Cặp (u,v) có thứ tự: đồ thị có hướng (directed graph/di-graph), ngược lại là đồ thị vô hướng (undirected graph)

Nội dung buổi 1

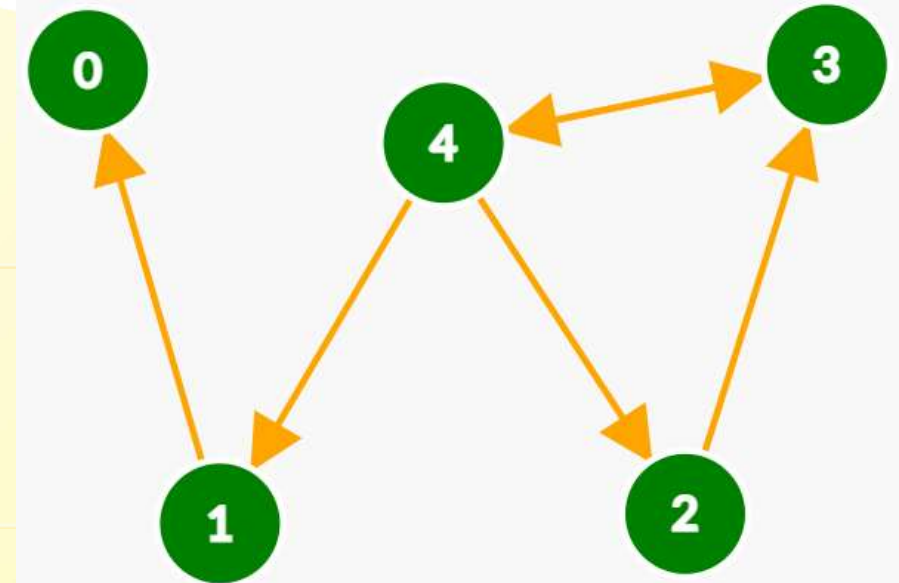
6.1 Các khái niệm trên đồ thị

- Định nghĩa
- Các loại đồ thị
- Khái niệm đường đi, chu trình, liên thông

6.2 Biểu diễn đồ thị trên máy tính

Nhập đơn đồ thị

- Trường hợp 1:
 - Cho biết số đỉnh, số cạnh
 - Quy ước đánh số các đỉnh là số nguyên $(0, 1, \dots)$ hoặc $(1, 2, \dots)$
- Input:
 - Dòng đầu tiên chứa 02 số v, e lần lượt là số đỉnh, số cạnh
 - e dòng tiếp theo, mỗi dòng chứa 02 số nguyên u và i thể hiện có cạnh nối từ đỉnh u sang đỉnh i



5	6
1	0
2	3
4	1
4	3
3	4
4	2

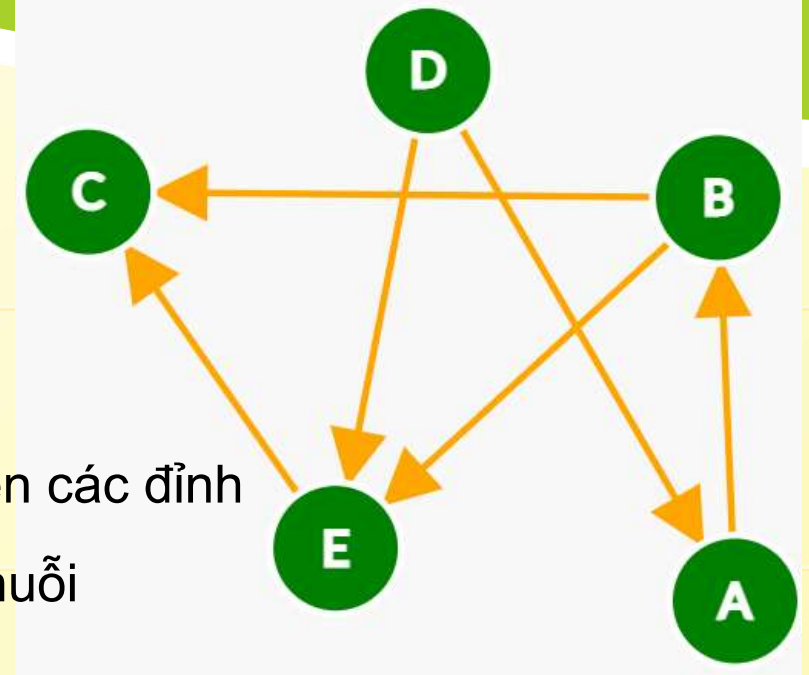
Nhập đơn đồ thị

- Trường hợp 2:

- Cho biết số đỉnh, số cạnh và danh sách tên các đỉnh
- Mỗi đỉnh của đồ thị được đặt tên là một chuỗi

- Input:

- Dòng đầu tiên chứa 02 số v, e lần lượt là số đỉnh, số cạnh
- Dòng tiếp theo chứa v chuỗi (chuỗi không khoảng trắng) là danh sách tên các đỉnh
- e dòng tiếp theo, mỗi dòng chứa 02 chuỗi u và i thể hiện có cạnh nối từ đỉnh u sang đỉnh i



5	6				
A	B	C	D	E	
A	B				
B	C				
D	E				
D	A				
E	C				
B	E				

BIỂU DIỄN ĐỒ THỊ

(Representations of a graph)

2 cách phổ biến nhất

1. Ma trận kề (Adjacency Matrix)

2. Danh sách kề (Adjacency List)

Khác:

3. Ma trận liên thuộc (Incidence Matrix)

4. Danh sách liên thuộc (Incidence List)

5. Danh sách cạnh,

***Việc chọn cách biểu diễn nào là tùy vào tình huống sử dụng*

MA TRẬN KỀ

- ✓ Đơn đồ thị vô hướng $G=(V,E)$,
- Tập đỉnh $V = \{0, 1, \dots, n-1\}$, có n đỉnh
- Tập cạnh $E = \{e_0, e_1, \dots, e_{m-1}\}$, có m cạnh

✓ Ma trận 2 chiều, kích thước $n \times n$

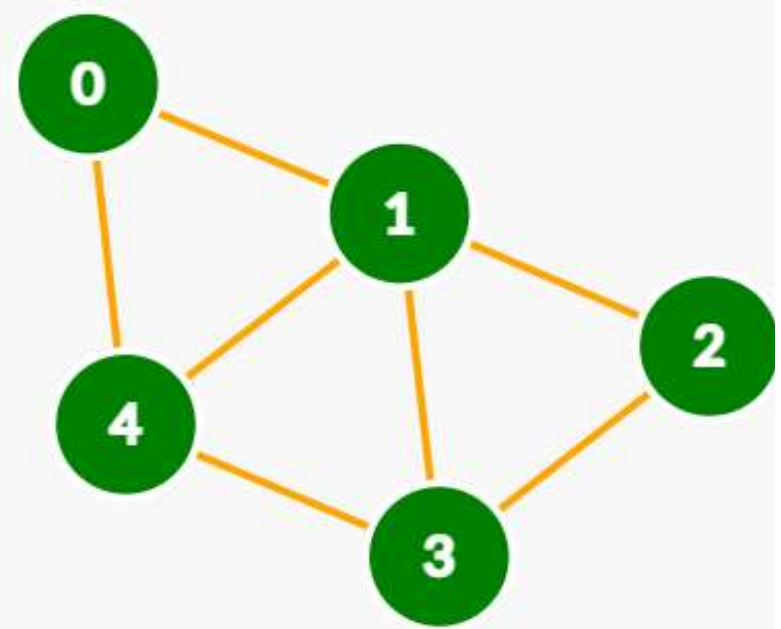
$$a = \{a_{ij} \mid i, j = 0, 1, \dots, n-1\} \rightarrow a[i][j]$$

Trong đó:

- $a_{ij} = 1$, nếu $(i,j) \in E$
- $a_{ij} = 0$, nếu $(i,j) \notin E$

✓ Ưu, nhược điểm?

1. Đối xứng: $a_{ij} = a_{ji}$
2. Tổng dòng (cột) i = bậc của đỉnh i

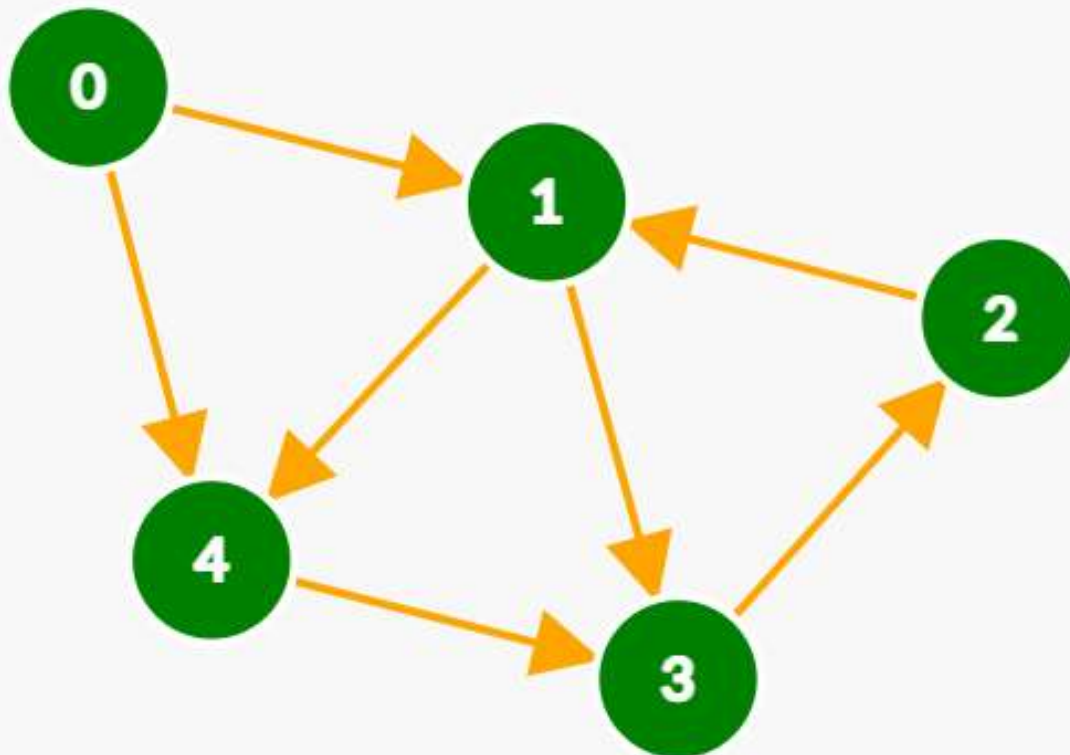


	0	1	2	3	4
0					
1					
2					
3					
4					

MA TRẬN KỀ

✓ Đơn đồ thị có hướng:

– ma trận không đối xứng ($a_{ij} \neq a_{ji}$)

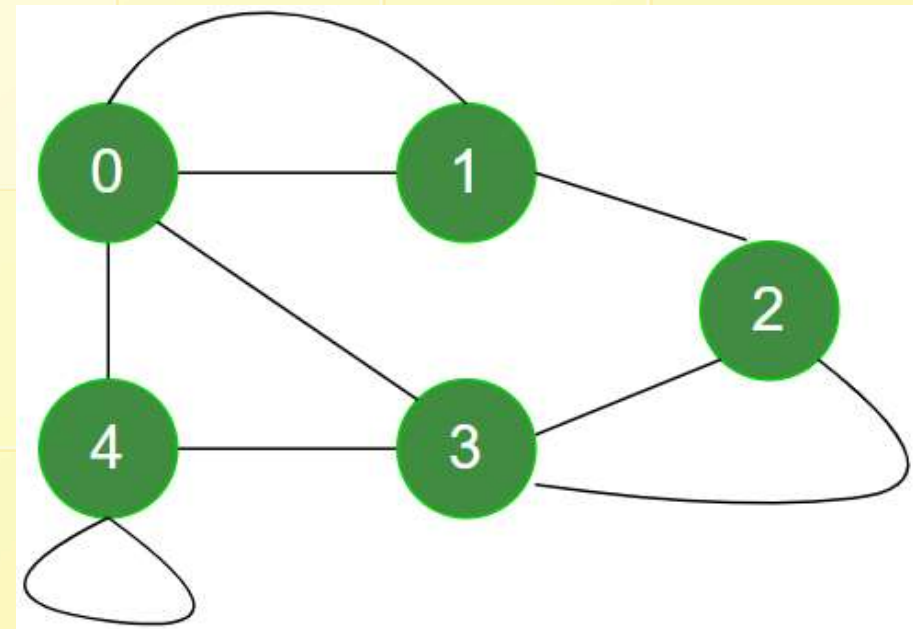
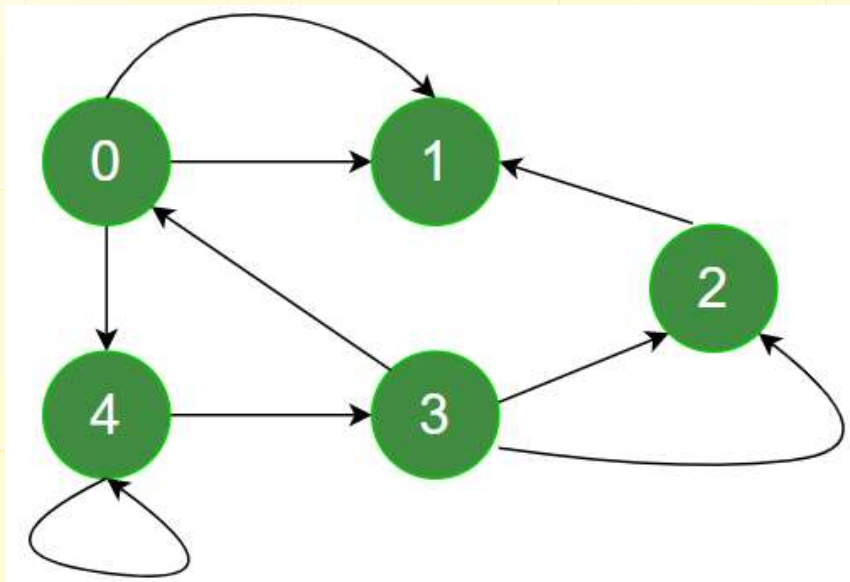


	0	1	2	3	4
0	0	1	0	0	1
1	0	0	0	1	1
2	0	1	0	0	0
3	0	0	1	0	0
4	0	0	0	1	0

MA TRẬN KỀ: dạng đặc biệt

✓ Đa đồ thị:

- $a_{ij} = k$, nếu $(i,j) \in E$, k là số cạnh nối 2 đỉnh i và j



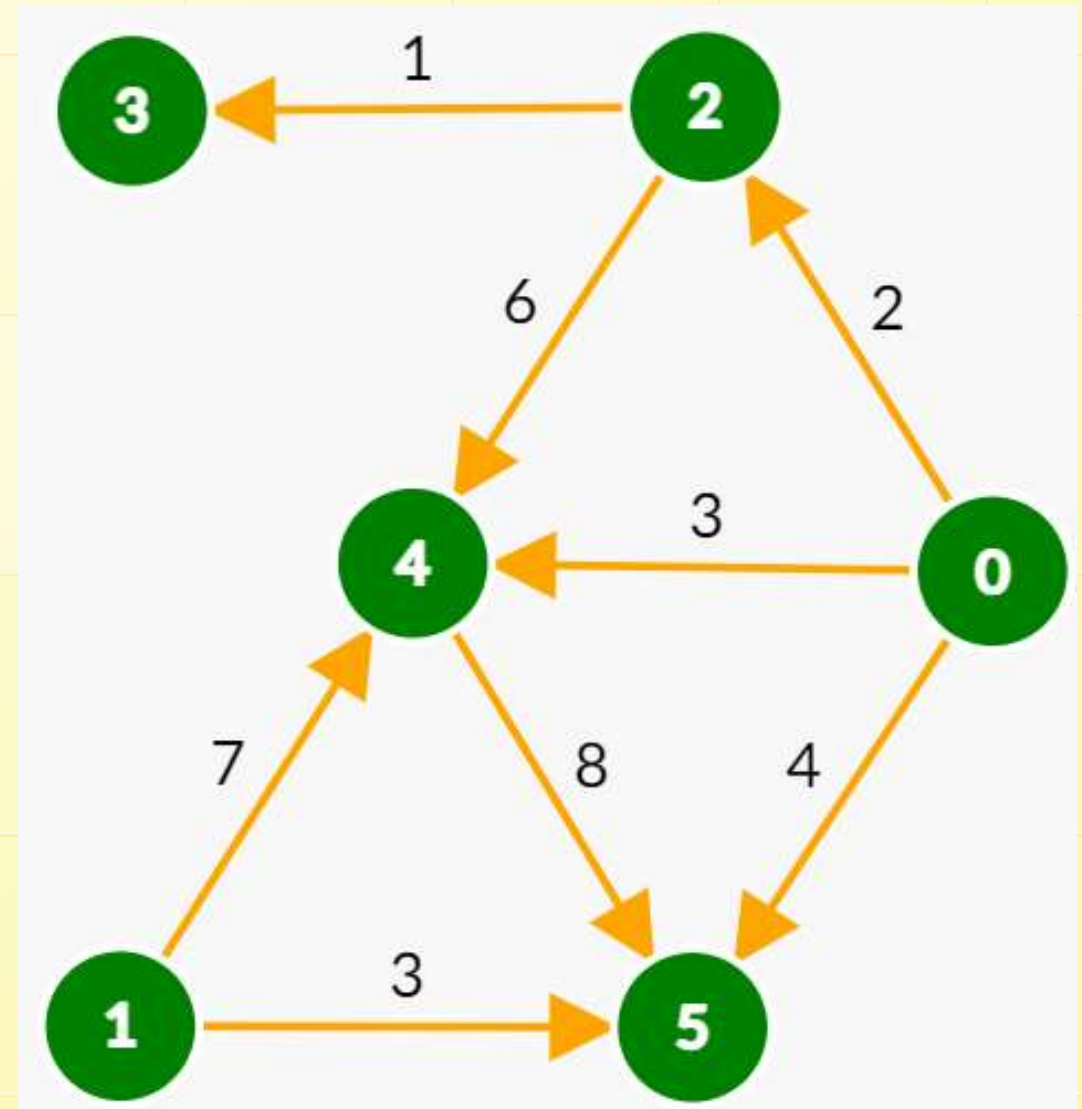
- Câu hỏi: Ma trận sẽ như thế nào?

MA TRẬN TRỌNG SỐ

✓ Đồ thị có trọng số:

- $a_{ij} = w(i,j)$ là trọng số của cạnh nối 2 đỉnh i và j

	0	1	2	3	4	5
0	0	0	2	0	3	4
1	0	0	0	0	7	3
2	0	0	0	1	6	0
3	0	0	0	0	0	0
4	0	0	0	0	0	8
5	0	0	0	0	0	0



CÀI ĐẶT: Ma trận kề/trọng số

Cách 1 (thủ công):

Bước 1: Khai báo và cấp phát động

- `int **a = new int* [n]; //cấp phát động mảng con trỏ`
- `a[i] = new int [n]`

// mỗi con trỏ a[i] quản lý một mảng động ứng với dòng thứ i của ma trận

Bước 2: Khởi tạo ma trận chứa toàn 0

Bước 3: Nhập các cạnh và lưu trữ thông tin cạnh vào ma trận kề

Cách 2: dùng STL

- `vector`

CÀI ĐẶT: Ma trận kề/trọng số

- ✓ Khởi tạo vector (matrix): vừa khai báo vừa khởi tạo

```
vector<vector<int> > G (v, vector<int> (v, 0));
```

- ✓ Khai báo rồi khởi tạo sau

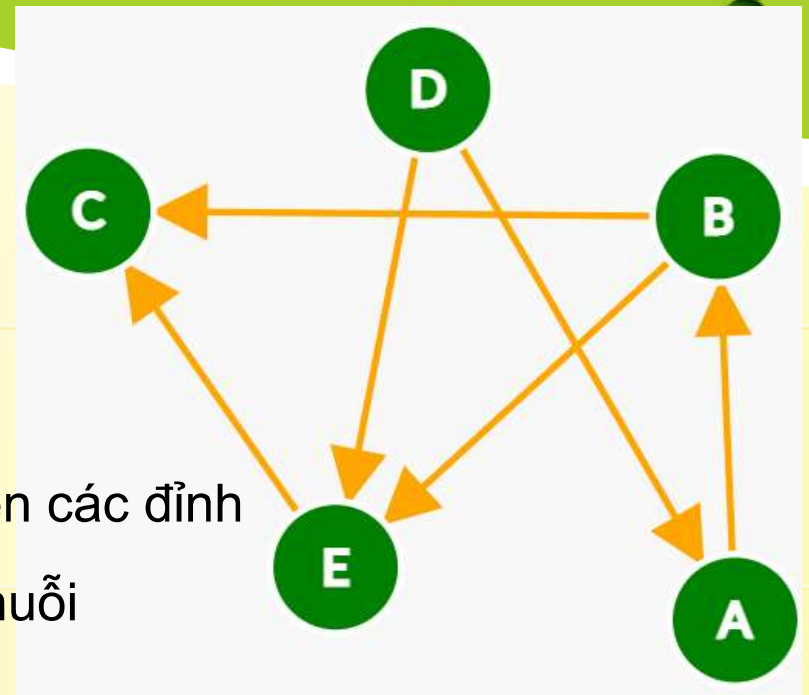
```
vector< vector<int> > a;
```

```
a = vector< vector<int> >(v, vector<int>(v, 0) );
```

Nhập đơn đồ thị

- Trường hợp 2:

- Cho biết số đỉnh, số cạnh và danh sách tên các đỉnh
- Mỗi đỉnh của đồ thị được đặt tên là một chuỗi

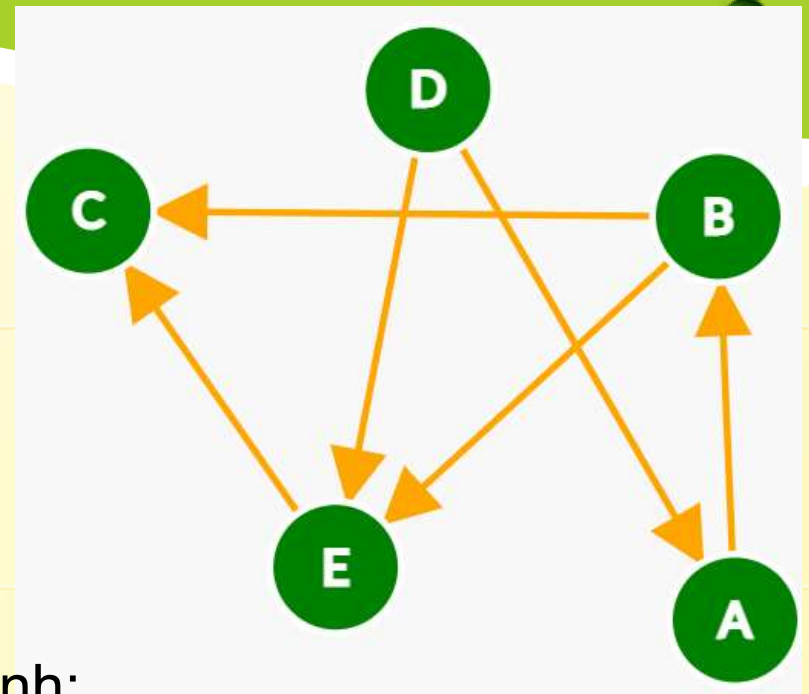


✓ Cần lưu trữ được các thông tin sau:

1. Ma trận kề của đồ thị
2. Lưu danh sách các tên (chuỗi) của đỉnh
3. Ánh xạ từ tên đỉnh (chuỗi) sang index (số nguyên) trong ma trận kề

5	6				
A	B	C	D	E	
A	B				
B	C				
D	E				
D	A				
E	C				
B	E				

Nhập đơn đồ thị



✓ Xây dựng CTDL cho đồ thị:

1. Ma trận kề của đồ thị
2. Cấu trúc lưu danh sách các tên (chuỗi) đỉnh: vector/list/tree....
3. **Ánh xạ từ tên đỉnh (chuỗi) sang index (số nguyên)** cho biết chỉ số dòng/cột tương ứng trong ma trận

Ví dụ:



Gợi ý:

1. `map<string, int> name_to_index;` hoặc
2. dùng mảng lưu các đỉnh (khi muốn biết đỉnh tương ứng với index nào thì phải search)

5	6			
A	B	C	D	E
A	B			
B	C			
D	E			
D	A			
E	C			
B	E			