



TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



Toán rời rạc

Nguyễn Khánh Phương

Bộ môn Khoa học máy tính
E-mail: phuongnk@soict.hust.edu.vn

Phần  thứ hai

LÝ THUYẾT ĐỒ THỊ

Graph Theory



Nguyễn Khánh Phương

Bộ môn Khoa học Máy tính,
Viện CNTT và Truyền thông,
Đại học Bách khoa Hà nội,

E-mail: phuongnk@soict.hust.edu.vn

Nội dung phần 2

Chương 1. Các khái niệm cơ bản

Chương 2. Biểu diễn đồ thị

Chương 3. Duyệt đồ thị

Chương 4. Cây và cây khung của đồ thị

Chương 5. Bài toán đường đi ngắn nhất

Chương 6. Bài toán luồng cực đại



TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

Chương 2

BIỂU DIỄN ĐỒ THỊ

(Representations of Graphs)



Biểu diễn đồ thị

- Có nhiều cách biểu diễn. Việc lựa chọn cách biểu diễn phụ thuộc vào từng bài toán cụ thể cần xét, thuật toán cụ thể cần cài đặt.
- Có hai vấn đề chính cần quan tâm khi lựa chọn cách biểu diễn:
 - Bộ nhớ mà cách biểu diễn đó đòi hỏi
 - Thời gian cần thiết để trả lời các truy vấn thường xuyên đối với đồ thị trong quá trình xử lý đồ thị:
 - Chẳng hạn:
 - Có cạnh nối hai đỉnh u, v ?
 - Liệt kê các đỉnh kề của đỉnh v ?

Nội dung chi tiết

2.1. Ma trận kẻ

2.2. Ma trận trọng số

2.3. Ma trận liên thuộc đỉnh cạnh

2.4. Danh sách kẻ

2.5. Danh sách các cạnh



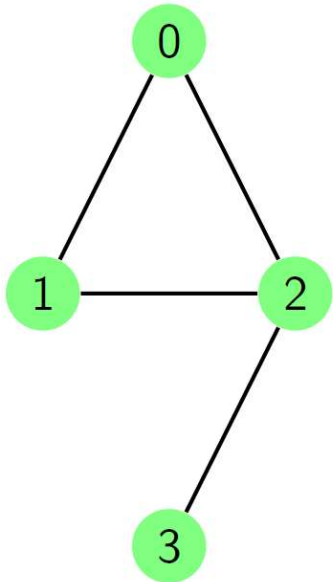
2.1. Ma trận kề (Adjacency Matrix)

- Ma trận kề A kích thước $|V| \times |V|$.
- Các đỉnh được đánh số từ 1 đến $|V|$ theo 1 thứ tự nào đó.
- A xác định bởi:

$$A[i, j] = a_{ij} = \begin{cases} 1 & \text{nếu } (i, j) \in E \\ 0 & \text{nếu trái lại} \end{cases}$$

$$n = |V|; m = |E|$$

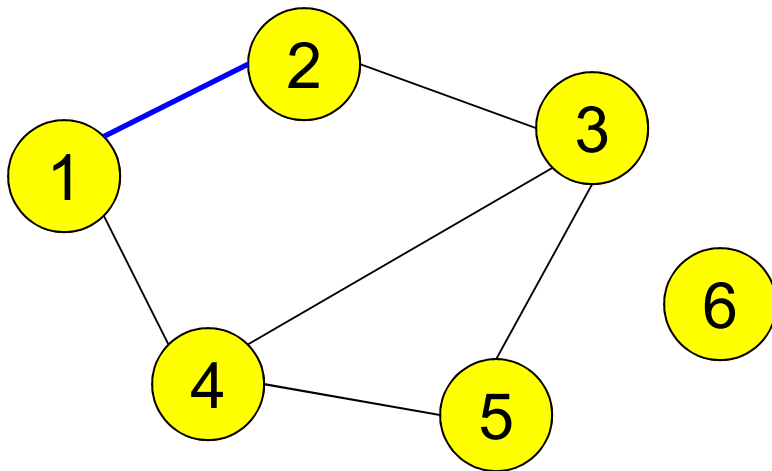
2.1. Biểu diễn đồ thị bởi ma trận kề



0	1	1	0
1	0	1	0
1	1	0	1
0	0	1	0

```
bool adj[4][4];  
adj[0][1] = true;  
adj[0][2] = true;  
adj[1][0] = true;  
adj[1][2] = true;  
adj[2][0] = true;  
adj[2][1] = true;  
adj[2][3] = true;  
adj[3][2] = true;
```

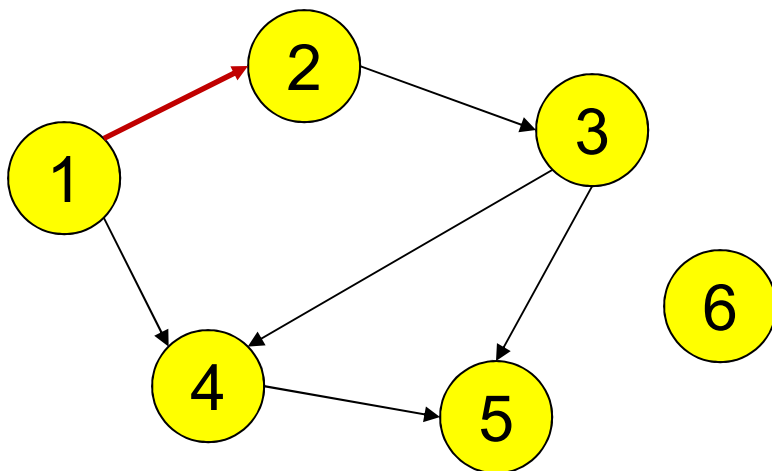

Ma trận kề của đồ thị vô hướng



$$A[u,v] = \begin{cases} 1 \text{ nếu } (u,v) \in E \\ 0 \text{ nếu trái lại} \end{cases}$$

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

Ma trận kề của đồ thị có hướng



$$A[u,v] = \begin{cases} 1 \text{ nếu } (u,v) \in E \\ 0 \text{ nếu trái lại} \end{cases}$$

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

Tính chất của ma trận kề

- Gọi A là ma trận kề của đồ thị vô hướng:
 - A là ma trận đối xứng: $A = A^T$ ($a_{ij} = a_{ji}$)
 - $\deg(v) =$ Tổng các phần tử trên dòng v của A
 - Nếu ký hiệu $A^k = (a^{(k)}[u, v])$ thì $a^{(k)}[u, v]$ là số lượng đường đi từ u đến v đi qua không quá $k-1$ đỉnh trung gian.
- Khái niệm ma trận kề có thể mở rộng để biểu diễn đa đồ thị vô hướng: a_{uv} – số lượng cạnh nối hai đỉnh u và v .

Phân tích chi phí

- Bộ nhớ (Space)
 - $|V|^2$ bits
 - $(|V|^2 + |V|)/2$ (nếu là đồ thị vô hướng, nhưng khó cài đặt).
 - Các thông tin bổ sung, chẳng hạn chi phí trên cạnh, cần được cất giữ dưới dạng ma trận. Một cách làm khác là cất giữ con trỏ đến các thông tin này.
- Thời gian trả lời các truy vấn
 - Hai đỉnh i và j có kề nhau? $O(1)$
 - Bổ sung hoặc loại bỏ cạnh $O(1)$
 - Bổ sung đỉnh: tăng kích thước ma trận
 - Liệt kê các đỉnh kề của v $O(|V|)$ (ngay cả khi v là đỉnh cô lập).

Nội dung chi tiết

2.1. Ma trận kê

2.2. Ma trận trọng số

2.3. Ma trận liên thuộc đỉnh cạnh

2.4. Danh sách kê

2.5. Danh sách các cạnh



2.2. Ma trận trọng số

- Trong trường hợp đồ thị có trọng số trên cạnh, thay vì ma trận kề, để biểu diễn đồ thị ta sử dụng **ma trận trọng số**

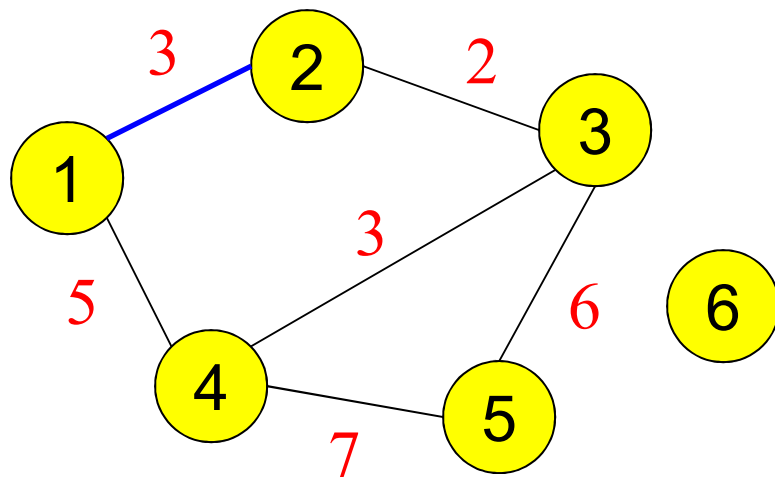
$$C = c[i, j], \quad i, j = 1, 2, \dots, n,$$

với

$$c[i, j] = \begin{cases} c(i, j), & \text{nếu } (i, j) \in E \\ \theta, & \text{nếu } (i, j) \notin E, \end{cases}$$

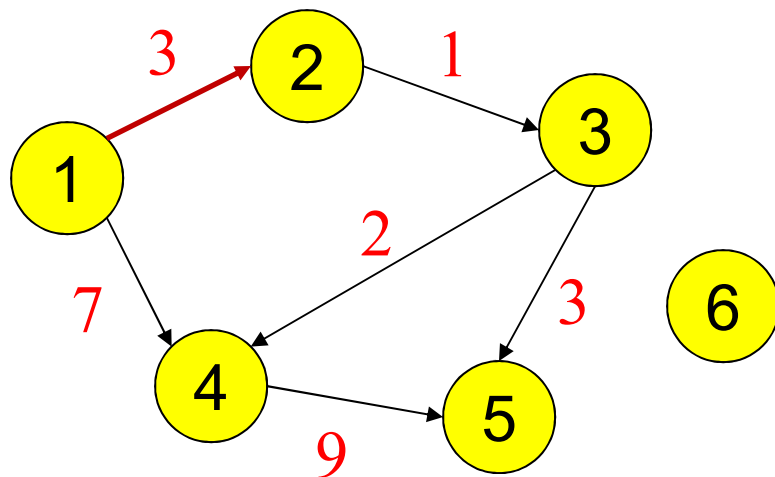
trong đó θ là giá trị đặc biệt để chỉ ra một cặp (i, j) không là cạnh, tùy từng trường hợp cụ thể, có thể được đặt bằng một trong các giá trị sau: $0, +\infty, -\infty$.

Ma trận trọng số của đồ thị vô hướng



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

Ma trận trọng số của đồ thị có hướng



	1	2	3	4	5	6
1	0	3	0	7	0	0
2	0	0	1	0	0	0
3	0	0	0	2	3	0
4	0	0	0	0	9	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0

Nội dung chi tiết

2.1. Ma trận kề

2.2. Ma trận trọng số

2.3. Ma trận liên thuộc đỉnh cạnh

2.4. Danh sách kề

2.5. Danh sách các cạnh



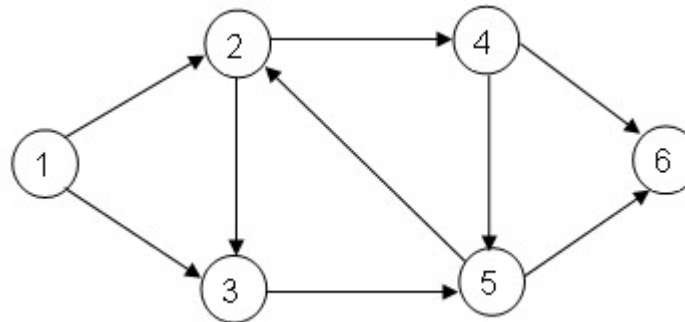
2.3. Ma trận liên thuộc đỉnh cạnh

- Xét $G = (V, E)$, ($V = \{1, 2, \dots, n\}$, $E = \{e_1, e_2, \dots, e_m\}$), là đơn đồ thị có hướng.
- Ma trận liên thuộc đỉnh cạnh $A = (a_{ij}: i = 1, 2, \dots, n; j = 1, 2, \dots, m)$, với

$$a_{ij} = \begin{cases} 1, & \text{nếu đỉnh } i \text{ là đỉnh đầu của cung } e_j, \\ -1, & \text{nếu đỉnh } i \text{ là đỉnh cuối của cung } e_j, \\ 0, & \text{nếu đỉnh } i \text{ không là đầu mút của cung } e_j, \end{cases}$$

- Ma trận liên thuộc đỉnh-cạnh là một trong những cách biểu diễn rất hay được sử dụng trong các bài toán liên quan đến đồ thị có hướng mà trong đó phải xử lý các cung của đồ thị.

2.3. Ma trận liên thuộc đỉnh cạnh



$$A = \begin{matrix} & \begin{matrix} (1,2) & (1,3) & (2,3) & (2,4) & (3,5) & (4,5) & (4,6) & (5,2) & (5,6) \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{matrix} & \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & -1 & -1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 \end{bmatrix} \end{matrix}$$

Nội dung chi tiết

2.1. Ma trận kề

2.2. Ma trận trọng số

2.3. Ma trận liên thuộc đỉnh cạnh

2.4. Danh sách kề

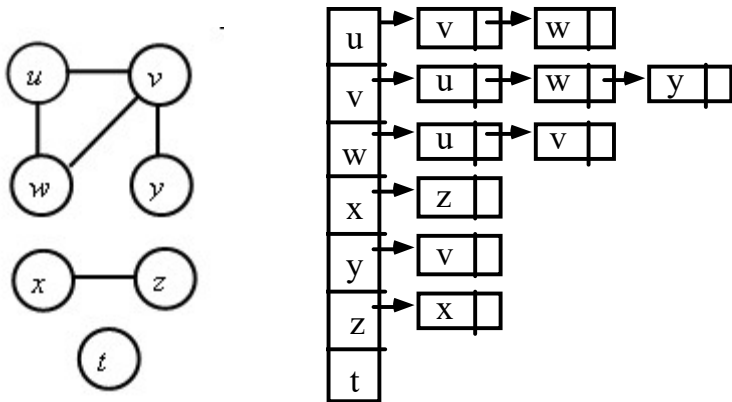
2.5. Danh sách các cạnh



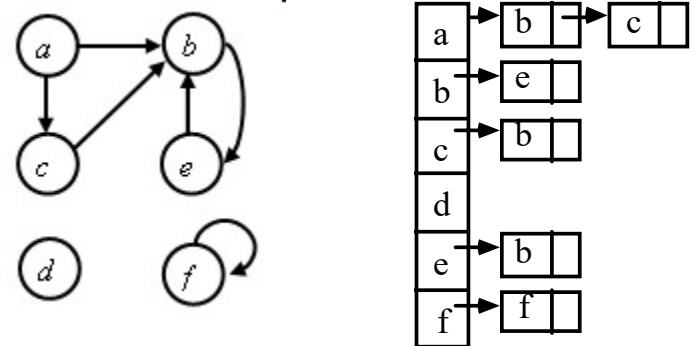
2.4. Danh sách kề

- **Danh sách kề (Adjacency Lists):** Với mỗi đỉnh v cất giữ danh sách các đỉnh kề của nó.
 - Là mảng Ke gồm $|V|$ danh sách.
 - Mỗi đỉnh có một danh sách.
 - Với mỗi $u \in V$, $Ke[u]$ bao gồm tất cả các đỉnh kề của u .
- **Ví dụ:**

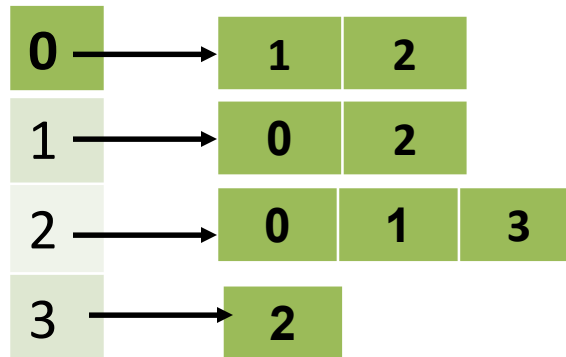
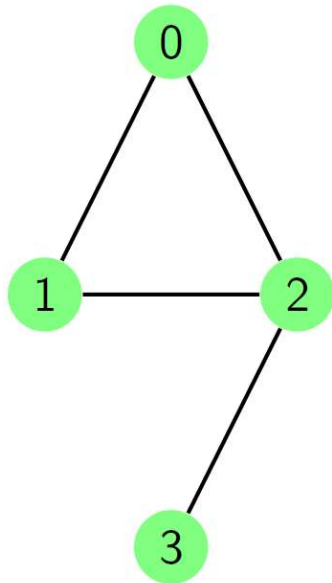
Đồ thị vô hướng



Đồ thị có hướng



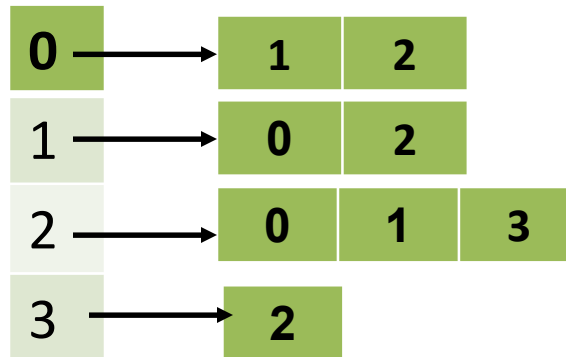
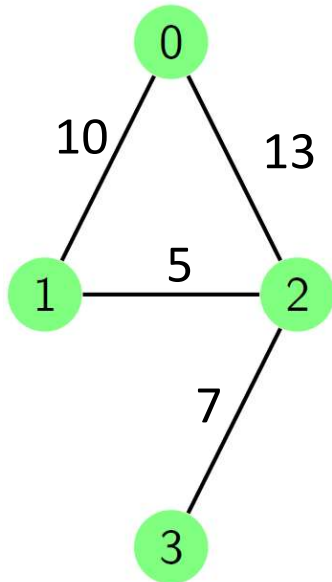
2.4. Biểu diễn đồ thị bởi danh sách kề



```
vector<int> adj[4];  
adj[0].push_back(1);  
adj[0].push_back(2);  
adj[1].push_back(0);  
adj[1].push_back(2);  
adj[2].push_back(0);  
adj[2].push_back(1);  
adj[2].push_back(3);  
adj[3].push_back(2);
```

- **Danh sách kề** (Adjacency Lists): Với mỗi đỉnh v cất giữ danh sách các đỉnh kề của nó.
 - Là mảng Ke gồm $|V|$ danh sách.
 - Mỗi đỉnh có một danh sách.
 - Với mỗi $u \in V$, $Ke[u]$ bao gồm tất cả các đỉnh kề của u .

2.4. Biểu diễn đồ thị bởi danh sách kề

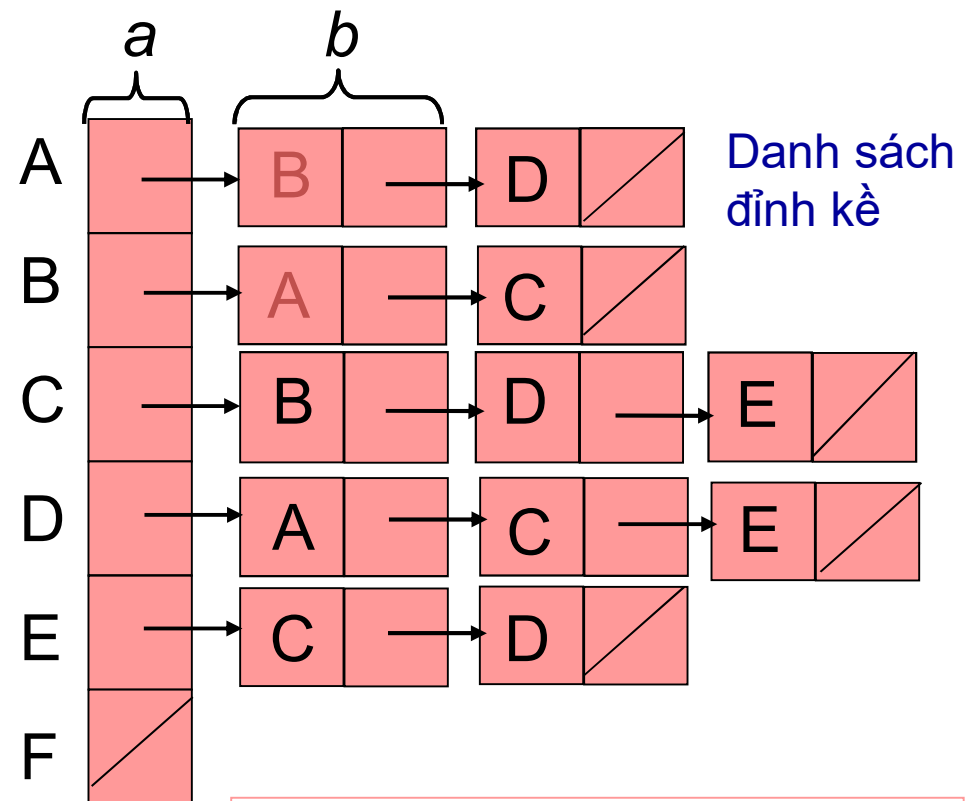
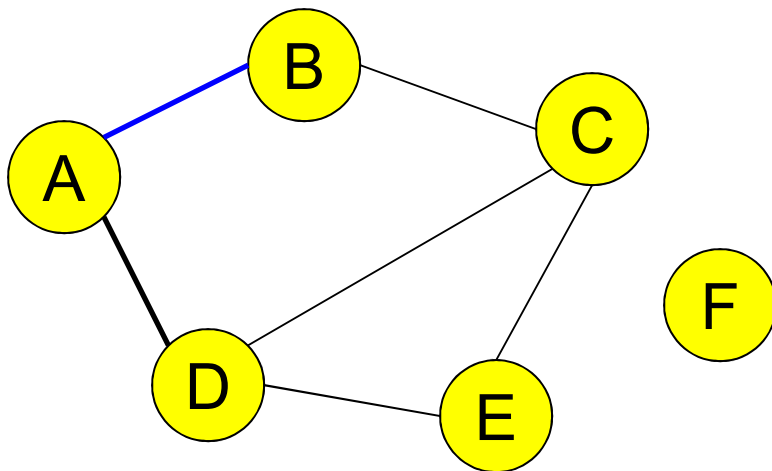


```
vector <pair <int, int> > adj[4];  
//hoặc list <pair <int, int> > adj[4];  
adj[0].push_back(make_pair(1, 10));  
adj[1].push_back(make_pair(0, 10)); //đồ thị vô hướng  
adj[0].push_back(make_pair(2, 13));  
adj[2].push_back(make_pair(0, 13)); //đồ thị vô hướng  
adj[1].push_back(make_pair(2, 5));  
adj[2].push_back(make_pair(1, 5)); //đồ thị vô hướng  
adj[2].push_back(make_pair(3, 7));  
adj[3].push_back(make_pair(2, 7)); //đồ thị vô hướng
```

```
typedef pair <int, int> iPair;  
vector <pair <int, int> > *adj; //hoặc list <pair <int, int> > *adj;  
adj = new vector <iPair> [4]; // hoặc adj = new list <iPair> [4];
```

Danh sách kề của đồ thị vô hướng

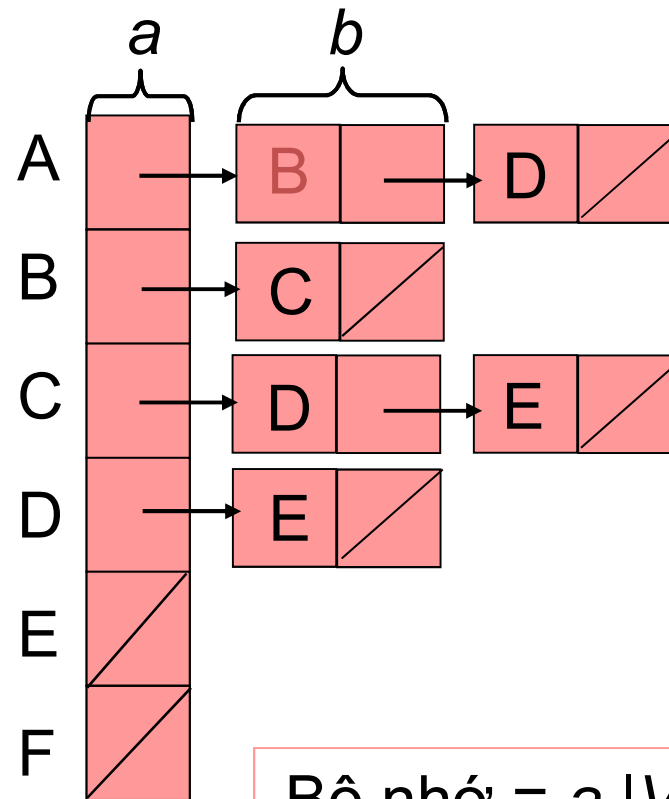
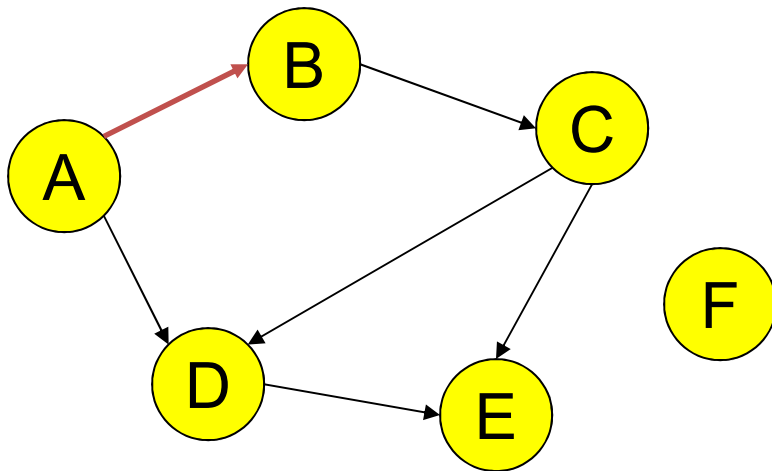
Với mỗi $v \in V$, $\text{Ke}(v)$ = danh sách các đỉnh u : $(v, u) \in E$



$$\text{Bộ nhớ} = a |V| + 2 b |E|$$

Danh sách kề của đồ thị có hướng

Với mỗi $v \in V$, $\text{Ke}(v) = \{ u: (v, u) \in E \}$



Bộ nhớ = $a |V| + b |E|$

Yêu cầu bộ nhớ

- Tổng cộng bộ nhớ: $\Theta(|V|+|E|)$
- Thường là nhỏ hơn nhiều so với $|V|^2$, nhất là đối với đồ thị thưa (sparse graph).
- Đồ thị thưa là đồ thị mà $|E| \leq k |V|$ với $k < 10$.
- *Chú ý:*
 - Phần lớn các đồ thị trong thực tế ứng dụng là đồ thị thưa!
 - Cách biểu diễn này được sử dụng nhiều nhất trong ứng dụng

Biểu diễn đồ thị

- Thời gian trả lời các truy vấn:
 - Thêm cạnh $O(1)$
 - Xoá cạnh Duyệt qua danh sách kề của mỗi đầu mút.
 - Thêm đỉnh Phụ thuộc vào cài đặt.
 - Liệt kê các đỉnh kề của v : $O(<\text{số đỉnh kề}>)$ (tốt hơn ma trận kề)
 - Hai đỉnh i, j có kề nhau?
 - Tìm kiếm trên danh sách: $\Theta(\text{degree}(i))$. Đánh giá trong tình huống tồi nhất là $O(|V|) \Rightarrow$ không hiệu quả (tồi hơn ma trận kề)

Nội dung chi tiết

2.1. Ma trận kẻ

2.2. Ma trận trọng số

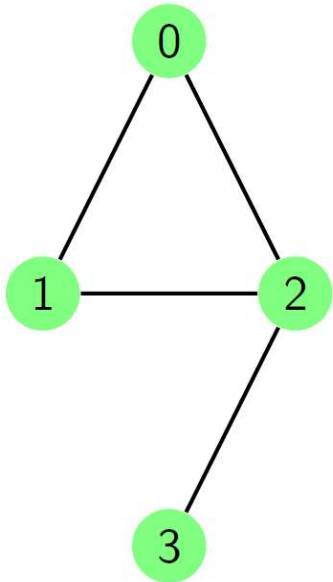
2.3. Ma trận liên thuộc đỉnh cạnh

2.4. Danh sách kẻ

2.5. Danh sách các cạnh



2.5. Biểu diễn đồ thị bởi danh sách cạnh



(0, 1),
(0, 2),
(1, 2),
(2, 3)

```
vector<pair<int, int> > edges;  
edges.push_back(make_pair(0, 1));  
edges.push_back(make_pair(0, 2));  
edges.push_back(make_pair(1, 2));  
edges.push_back(make_pair(2, 3));
```

So sánh phân tích chi phí

	Danh sách kề	Ma trận kề	Danh sách cạnh
Bộ nhớ lưu trữ	$O(V + E)$	$O(V ^2)$	$O(E)$
Thêm đỉnh	$O(1)$	$O(V ^2)$	$O(1)$
Thêm cạnh	$O(1)$	$O(1)$	$O(1)$
Xóa đỉnh	$O(E)$	$O(V ^2)$	$O(E)$
Xóa cạnh	$O(E)$	$O(1)$	$O(E)$
Truy vấn: u, v có kề nhau không	$O(V)$	$O(1)$	$O(E)$