Lab 01: Biểu diễn Đồ thị trên máy tính

I. Ma trận kề

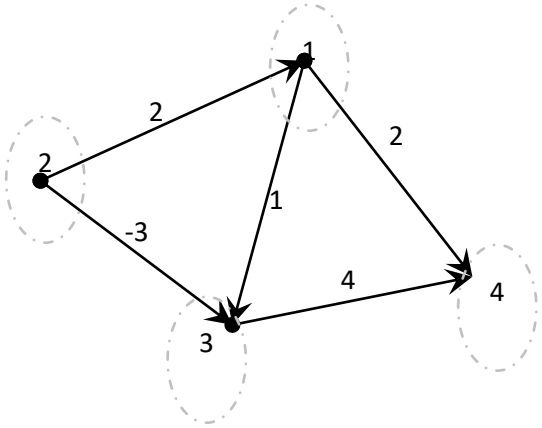
Định nghĩa:

Giả sử G = (V, E) là một đơn đồ thị có số đỉnh là n. ***Ma trận kề*** là ma trận vuông $A = (a_{ij})$ cấp n x n với n là số đỉnh của đồ thị. Trong đó: $a_{ij} = \{1\ nếu\ (i,j) \in E\ 0\ nếu\ (i,j) \notin E$

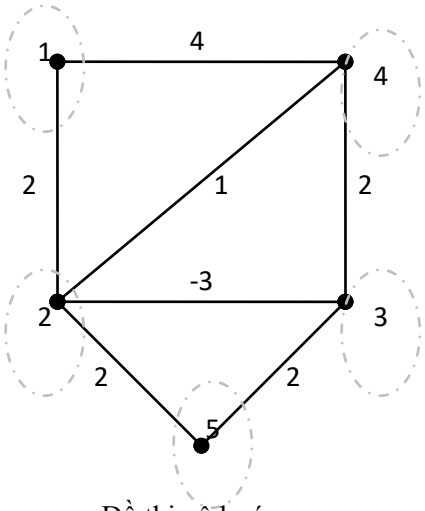
Một số tính chất:

- Đồ thị vô hướng $a_{ij} = a_{ji}$ (ma trận đối xứng qua đường chéo chính)
- Đường chéo chính $a_{ii} = 0$ (do không có khuyên).

Ví dụ:



Đồ thị có hướng



Đồ thị vô hướng

Ma trận kề tương ứng 2 đồ thị trên:

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

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

Cài đặt:

Thông thường ma trận kề được lưu trữ trên tập tin và được chương trình đọc lên để thực hiện các thuật toán trên đồ thị tương ứng.

- Dòng đầu tiên chứa số đỉnh của đồ thị
- n dòng tiếp theo, mỗi dòng chứa n số nguyên (0 hoặc 1) tương ứng với giá trị của các phân tử trong ma trận kề.

Ví dụ: đồ thị có hướng ở trên sẽ được lưu trong tập tin DOTHI.TXT như sau:

```
4
0   0   1   1
0   0   1   1
1   1   0   1
1   1   1   0
```

II. Một số gợi ý

Khai báo cấu trúc dữ liệu

```
const int MAX = 100; //so dinh toi da cua do thi

//Do thi gom co so dinh va ma tran ke.
struct GRAPH
{
	int sodinh;//so dinh do thi
	int a[MAX][MAX];//ma tran ke
};
```

Hàm đọc dữ liệu lên từ file

```
void readGRAPH (string fn, GRAPH &g)
{
	ifstream f;
	f.open(fn);
	if(f.is_open())
	{
		f>>g.sodinh;
		for(int i = 0;i < g.sodinh; i++)
			for(int j = 0;j < g.sodinh; j++)
				f>>g.a[i][j];
		f.close();
	}
	else
		cout<<"Khong mo duoc file!!!";
}
```

Hàm xuất ma trận kề

```
void printGRAPH (GRAPH g)
{
	cout<<"So dinh cua do thi: "<<g.sodinh<<endl;
	for(int i=0;i<g.sodinh;i++)
	{
		for(int j=0;j<g.sodinh;j++)
			cout<<setw(4)<<g.a[i][j];
		cout<<endl;
	}
}
```

III. Bài tập

Viết chương trình đọc ma trận kề của đồ thị. Xác định và in ra:

a) Kiểm tra tính hợp lệ của đồ thị (giá trị trên đường chéo chính bằng 0).

```
int KiemTraMaTranKeHopLe(GRAPH &g)
{
	// kiểm tra các giá trị a[0][0], a[1][1], … xem có giá trị khác 0 hay không
	// nếu có, nghĩa là ma trận kề không hợp lệ
	int i;
	for (i=0; i<g.sodinh; i++)
	if (g.a[i][i] != 0)
		return 0;
	return 1;
}

// cách sử dụng trong hàm main như sau
//	if (!KiemTraMaTranKeHopLe(g)
//	{
//		printf("Ma tran ke khong hop le");
//		exit(0);
//	}
```

b) Cho biết đồ thị có hướng hay vô hướng?

```
int KiemTraDoThiVoHuong(GRAPH &g)
{
	// kiểm tra xem các giá trị a[i][j] có bằng với a[j][i] hay không
	// nếu có, nghĩa là đồ thị không đối xứng
	int i, j;
	for (i=0; i<g.sodinh; i++)		// có thể giảm bớt các bước kiểm tra thừa với
	for (j=0; j<g.sodinh; j++)		// for (j=i+1; j<n; j++)
	if (g.a[i][j] != g.a[j][i])
		return 0;
```

```
        return 1;
} //thuat toan nua tam giac tren cheo chinh.
```

a) Số cạnh, số đỉnh của đồ thị? (*Phân biệt 2 trường hợp đồ thị có hướng và vô hướng*)

b) Xuất bậc của tất cả các đỉnh

Hướng dẫn:

- *Bậc của 1 đỉnh trong đồ thị vô hướng là tổng số các cạnh kề với đỉnh đó. Khuyên được tính 2 lần.*
- *Bậc nửa ngoài/trong của 1 đỉnh trong trong đồ thị hữu hướng là số cạnh đi ra/vào đỉnh đó. Tổng nửa bậc ngoài và trong của đỉnh là bậc của đỉnh.*

c) Các đỉnh có số bậc lớn nhất/nhỏ nhất, đỉnh bậc chẵn, đỉnh bậc lẻ?

d) Các định cô lập, đỉnh treo?

Đỉnh treo là đỉnh có bậc bằng 1, Đỉnh cô lập là đỉnh có bậc bằng 0.

IV. Tài liệu tham khảo

[1]. *Bài tập thực hành Lý thuyết đồ thị*, Khoa CNTT, ĐH Khoa học Tự Nhiên, ĐHQG TpHCM.