### **LUCRAREA DE LABORATOR Nr. 1**

# TEMA: PĂSTRAREA GRAFURILOR ÎN MEMORIA CALCULATORULUI

1. **SCOPUL LUCRĂRII:**

* Studierea metodelor de definire a unui graf: matrice de incidenţă, matrice de adiacenţă, liste;
* Elaborarea unor proceduri de introducere, extragere şi transformare a diferitor forme de reprezentare internă a grafurilor cu scoaterea rezultatelor la display şi imprimantă.

**2. NOTE DE CURS**

## Metode de reprezentare a grafului

Există trei metode de bază de definire a unui graf:

1. *Matricea de incidenţă;*
2. *Matricea de adiacenţă;*
3. *Lista de adiacenţă (incidenţă).*

Vom face cunoştinţă cu fiecare dintre aceste metode.

## Matricea de incidenţă

Este o matrice de tipul *m*x*n*, în care *m* este numărul de muchii sau arce (pentru un graf orientat), iar *n* este numărul vârfurilor. La intersecţia liniei *i* cu coloana *j* se vor considera valori de *0* sau *1* în conformitate cu următoarea regulă:

* *1* - dacă muchia *i* este incidentă cu vârful *j* (dacă arcul *i* "intră" în vârful *j* în cazul unui graf orientat);
* *0* - dacă muchia (arcul) *i* şi vârful *j* nu sunt incidente;
* *-1* - numai pentru grafuri orientate, dacă arcul *i* "iese" din vârful *j*.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | *x1* | *x2* | *x3* | *x4* | *x5* | *x6* | *x7* |
| *u1* | -1 | 0 | 1 | 0 | 0 | 0 | 0 |
| *u2* | -1 | 0 | 0 | 1 | 0 | 0 | 0 |
| *u3* | 0 | 0 | 0 | -1 | 0 | 0 | 1 |
| *u4* | 0 | 0 | -1 | 0 | 0 | 0 | 1 |
| *u5* | 0 | -1 | 1 | 0 | 0 | 0 | 0 |
| *u6* | 0 | -1 | 0 | 0 | 1 | 0 | 0 |
| *u7* | 0 | -1 | 0 | 0 | 0 | 1 | 0 |
| *u8* | 0 | 0 | -1 | 0 | 0 | 1 | 0 |

Fig. 3. Exemplu de matrice de incidenţă (v.fig.1)

Este uşor de observat că această metodă este de o eficacitate mică în sensul utilizării memoriei calculatorului: fiecare linie conţine doar două elemente diferite de zero (o muchie poate fi incidentă cu nu mai mult de două vârfuri).

În limbajul C matricea de incidenţă poate fi redată printr-un tablou bidimensional *m*x*n*:

## Matricea de adiacenţă

Este o matrice pătrată *n*x*n*, aici *n* este numărul de vârfuri. Fiecare element poate fi *0*, dacă vârfurile respective nu sunt adiacente, sau 1, în caz contrar. Pentru un graf fără bucle putem observa următoarele:

* diagonala principală este formată numai din zerouri;
* pentru grafuri neorientate matricea este simetrică faţă de diagonala principală.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | *x1* | *x2* | *x3* | *x4* | *x5* | *x6* | *x7* |
| *x1* | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| *x2* | 0 | 0 | 1 | 0 | 1 | 1 | 0 |
| *x3* | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| *x4* | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| *x5* | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| *x6* | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| *x7* | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Fig. 4. Exemplu de matrice de adiacenţă (v.fig.1)

După cum este lesne de observat şi în acest caz memoria calculatorului este utilizată nu prea eficace din care cauză matricea de adiacenţă ca şi matricea de incidenţă se vor utiliza de obicei doar în cazul în care se va rezolva o problemă concretă pentru care reprezentarea grafului în această formă aduce unele facilităţi algoritmului respectiv.

Pentru păstrarea grafurilor în memoria calculatorului (în deosebi, memoria externă) se va utiliza una din posibilităţile de mai jos.

## Lista de adiacenţă şi lista de incidenţă

**Lista de adiacenţă** este o listă cu *n* linii (după numărul de vârfuri *n*), în linia cu numărul *i* vor fi scrise numerele vârfurilor adiacente cu vârful *i*.

**Lista de incidenţă** se defineşte analogic cu deosebirea că în linia *i* vor fi scrise numerele muchiilor (arcelor) incidente cu vârful *i*.

Reprezentarea grafurilor prin intermediul acestor liste permite utilizarea mai eficace a memoriei calculatorului, însă aceste forme sunt mai complicate atât în realizare, cât şi în timpul procesării. Pentru a lua în consideraţie lungimea variabilă a liniilor vor fi utilizate variabile dinamice şi pointeri.Vom exemplifica pentru un graf cu *n* vârfuri. Deoarece fiecare element al listei conţine numere de vârfuri este evident să considerăm că vom avea un şir de variabile dinamice de tip INTEGER care se vor afla în relaţia respectivă de precedare (succedare). Această relaţie se va realiza prin pointeri, uniţi împreună cu variabila de tip întreg. Pentru a păstra indicatorii de intrare în aceste şiruri se va folosi un tablou unidimensional de indicatori de lungime *n*. În calitate de simbol de terminare a şirului se va utiliza un simbol care nu a fost folosit la numeraţia vârfurilor (de exemplu *0*), care va fi introdus în calitate de variabilă de tip întreg al ultimului bloc.

**De exemplu**, lista de adiacenţă (fig.1.1):

1 - 3, 4, 0

2 - 3, 5, 6, 0

3 - 6, 7, 0

4 – 7, 0

5 - 0

6 - 0

7 - 0

va avea următoarea reprezentare internă:

…

2

3

4

…

3

^

5

^

6

^

0

variabile dinamice (pointer şi variabilă de tip întreg)

masiv de indicatori

Fig. 5. Reprezentarea internă a listei de adiacenţă

**3. SARCINA DE BAZĂ**

1. Elaboraţi procedura introducerii unui graf în memoria calculatorului în formă de matrice de incidenţă, matrice de adiacenţă şi listă de adiacenţă cu posibilităţi de analiză a corectitudinii.

2. Elaboraţi proceduri de transformare dintr-o formă de reprezentare în alta.

3. Folosind procedurile menţionate elaboraţi programul care va permite:

* introducerea grafului reprezentat sub oricare din cele trei forme cu posibilităţi de corecţie a datelor;
* păstrarea grafului în memoria externă în formă de listă de adiacenţă;
* extragerea informaţiei într-una din cele trei forme la imprimantă şi display.

**4. ÎNTREBĂRI DE CONTROL**

1. Care sunt metodele de bază de reprezentare a unui graf?

2. Descrieţi fiecare din aceste metode.

3. Cum se vor realiza aceste metode în limbajul C?