Formy reprezentacji grafów.

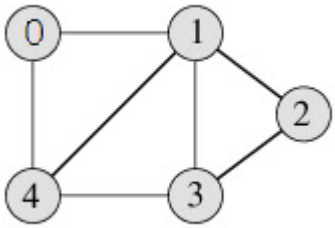
Podstawy

Podstawowe pojęcia:

G(V,E)

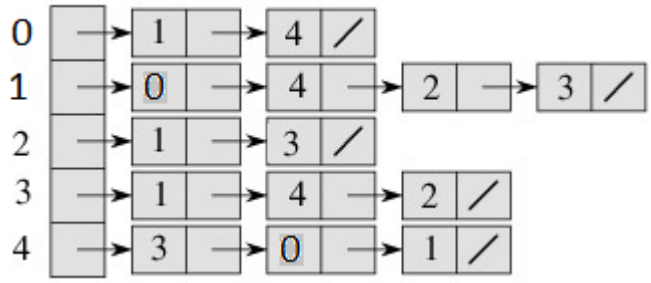
V = [v_1…v_n]– wierzchołki (vertex)

E =[e_1…e_m] – krawędzie (edges)



[1]

Lista sąsiedztwa - Dane zapisywane są w postaci listy obiektów zawierających wierzchołek grafu,wraz z listą listę wierzchołków sąsiednich.



[2]

Macierz sąsiedztwa - dane zapisane są w postaci macierzy, która ma wymiar |V| x |V|, a wartości reprezentują wagę połączeń pomiędzy wierzchołkami.

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

[3]

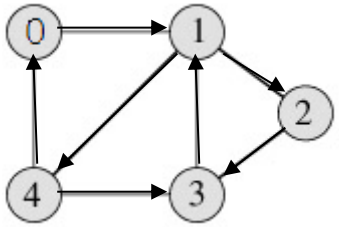
---

[1]

[2] Obrazki pochodzą ze strony: http://www.geeksforgeeks.org/graph-and-its-representations/

Macierz incydencji – macierz o wymiarze |V|x|E| (grafy skierowane) w liczba wierszy odpowiada liczbie wierzchołków, a liczbie kolumn odpowiada liczba krawędzi. Zasada wstawiania wartości można opisać wg. zależności:

$$m_{ij} = \begin{cases} 1 \text{ jeśli krawędź } e_j \text{ wychodzi z wierzchołka } v_i \\ -1 \text{ jeśli krawędź } e_j \text{ wchodzi z wierzchołka } v_i \\ 0 \text{ jeśli } e_j \text{i} v_i \textit{ nie są incydentne} \end{cases}$$

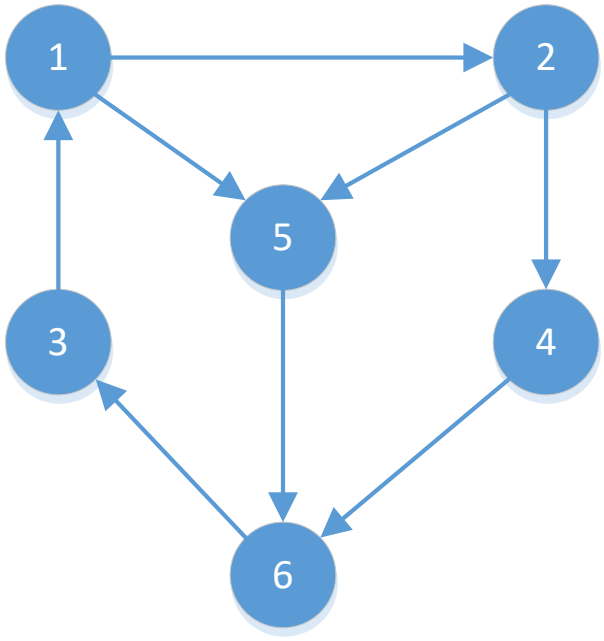


	1	2	3	4	5	6	7
0	1	-1					
1	-1		1	-1	1		
2					-1	1	
3				1		-1	-1
4		1	-1				1

	Lista sąsiedztwa	**Macierz sąsiedztwa**	**Macierz incydencji**
Przechowywanie grafu	O(|V|+|E|)	O(|V|^2)	O(|V||E|)
Dodanie wierzchołka	O(1)	O(|V|^2)	O(|V||E|)
Dodanie krawędzi	O(1)	O(1)	O(|V||E|)
Usunięcie wierzchołka	O(|E|)	O(|V|^2)	O(|V||E|)
Usunięcie krawędzi	O(|E|)	O(1)	O(|V||E|)
Zapytanie: czy wierzchołki x i y są sąsiadujące	O(|V|)	O(1)	O(|V||E|)

Dla grafów warzonych zamiast 1/-1 można wprowadzić wartości opisujące wagę połączenia

Zadania



1)
W przypadku języków Java/C# stwórz interfejs:

```
public interface IGraf
{
    //zwraca id wierzcholka
    int dodajWierzcholek();
    //usuwa wierzcholek o okreslonym id
    void usunWierzcholek(int id);
    //dodaje krawedz o wadze "waga" pomiedzy wierzcholkami start i koniec
    void dodajKrawedz(int start, int koniec, double waga);
    //usuwa krawedz
    void usunKrawedz(int start, int koniec);
    //zwraca stopien wierzcholka
    double stopienWierzcholka(int id);
    //zwraca liczbe weirzcholkow w grafie
    int rzadGrafu();
    //zwraca liczbe krawedzi w grafie
    int rozmiarGrafu();
    //zwraca graf transponowany
    IGraf transponuj();
    //zwraca wagę opisującą krawędź łączącą węzły x i y,
    //jeśli brak krawędzi zwraca NaN
    double wagaPolaczenia(int start, int koniec);
}
```

2)
Dokonaj implementacji trzech poznanych form reprezentacji grafu. Powstałe klasy, osobno dla każdej formy reprezentacji grafu, powinny rozszerzać interface IGraf

3)
Dla podanego grafu skierowanego zapisz go w postaci listy sąsiedztwa i napisz program który wypisuje dla każdego z wierzchołków listę wierzchołków które na niego wskazują. (czyli te wierzchołki, których krawędzie kończą się w danym wierzchołku).
Wyznacz stopień każdego z wierzchołków i to zarówno stopień wychodzący jak i przychodzący.

4)
Podany graf zapisz w postaci macierzy incydencji.
Napisz program który dla danego wierzchołka wyznacza wszystkich jego sąsiadów.
Napisz program który dla danego wierzchołka wyznacza wszystkie wierzchołki które sa jego sąsiadami.
Napisz program wyznaczający stopień wierzchołka osobny wychodzący i przychodzący

5)
Podany graf zapisz w postaci macierzy sąsiedztwa.
Napisz program który dla danego wierzchołka wyznacza wszystkich jego sąsiadów.
Napisz program który dla danego wierzchołka wyznacza wszystkie wierzchołki które sa jego sąsiadami.
Napisz program wyznaczający stopień wierzchołka osobny wychodzący i przychodzący