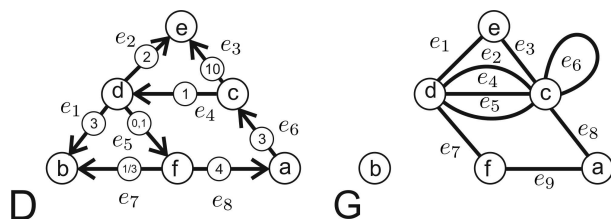


# Algorytmy grafowe 01: Wprowadzenie.

## A Zadania na rozgrzewkę - nie obowiązkowe

**Zadanie A.1.** Dla (multi)grafu  $G$  i digrafu  $D$  z ilustracji wyznacz (jeśli można):

- listy następników;
- macierz wag, macierz przyległości, macierz następników;
- macierz incydencji.



## B Program do napisania

Proszę o przesłanie

- do nocy z soboty na niedzielę (**14/15 marca**);
- w mailu o tytule **AGR01** (WAŻNE: Nie będę czytała tych maili, więc z istotnymi sprawami proszę się zgłaszać w osobnych mailach.);
- plików o zindywidualizowanej nazwie **01NazwiskoImie.py** (nazwisko pierwsze, bez polskich znaków, proszę);
- na adres: kryba@amu.edu.pl.

UWAGA: Przypominam, że piszemy w Pythonie3

**Zadanie B.1.** W pliku graph.txt zapisane są w kolejnych liniach:

- liczba wierzchołków grafu  $G$ ;
- w kolejnych liniach macierz wag grafu  $G$  z wagami całkowitoliczbowymi (wagi mogą być wielocyfrowe, znak '-' oznacza ' $\infty$ ');
- w kolejnych liniach pary liczb (o wartościach nie większych niż liczba wierzchołków w grafie) – liczba tych linii nie jest określona.

Napisz program, który kolejno:

- Odczytuje z pliku tę macierz wag grafu  $G$  i zapisuje w postaci listy list ('-' zastępuje przez float('inf'))
- Odczytuje krawędzie i zapisuje w Państwa ulubionej formie;
- Zapisuje listę następników, cały graf z wagami, listę krawędzi z wagami jako słowniki (patrz notatki z wykładu) i wypisuje uzyskane słowniki. (**Wierzchołki numerujemy liczbami naturalnymi od 1**);
- Wypisuje „ładnie” listę następników grafu  $G$ ;
- Dla krawędzi odczytanych z pliku, kolejno, jeśli krawędź była w grafie – usuwa ją, jeśli nie było jej w grafie dodaje ją do grafu z wagą 3 (do macierzy i do wszystkich zapisanych słowników).
- wypisuje uzyskane słowniki i „ładnie” listę następników uzyskanego grafu.

UWAGA: fragmenty tego zadania będą Państwu przydatne do kolejnych programów. Dlatego warto je zrobić „porządnie”.

**Zadanie B.2. (Dla chętnych zamiast zadania B.1)** Stwórz obiekt Graph, w którym:

- będą wierzchołki i krawędzie z wagami odpowiednio ze sobą powiązane;
- będzie można dodawać i usuwać wierzchołki;
- będzie można dodawać i usuwać krawędzie z wagami;
- będzie można odczytać wagę krawędzi;
- będzie można „pobrać” sąsiadów wierzchołka w postaci listy i w postaci słownika (z zapisanymi wagami krawędzi).

Następnie zrób wersję zadania B.1 z wykorzystaniem tego obiektu (bez wypisywania słowników).

## Przykładowe rozwiązanie zadania B.1

Uwaga: warto pobrać plik txt ze strony a nie kopiować to poniżej do pliku txt.

WEJŚCIE:

```
8
- 16 - - - 8 2 -
16 - - - 2 - 3 -
- - - -4 - 5 - 3
- - -4 - - 5 - -
- 2 - - - - 6 -
8 - 5 5 - - - 90
2 3 - - 6 - - -
- - 3 - - 90 - -
1 2
2 6
8 7
```

WYJŚCIE:

```
{1: [2, 6, 7], 2: [1, 5, 7], 3: [4, 6, 8], 4: [3, 6], 5: [2, 7], 6: [1, 3, 4, 8], 7: [1, 2, 5], 8: [3, 6]}
{1: {2: 16, 6: 8, 7: 2}, 2: {1: 16, 5: 2, 7: 3}, 3: {4: -4, 6: 5, 8: 3}, 4: {3: -4, 6: 5}, 5: {2: 2, 7: 6}, 6: {1: 8, 3: 5, 4: 5, 8: 90},
7: {1: 2, 2: 3, 5: 6}, 8: {3: 3, 6: 90}}
{(1, 2): 16, (1, 6): 8, (1, 7): 2, (2, 5): 2, (2, 7): 3, (3, 4): -4, (3, 6): 5, (3, 8): 3, (4, 6): 5, (5, 7): 6, (6, 8): 90}
```

Lista następników:

```
1 : 2 6 7
2 : 1 5 7
3 : 4 6 8
4 : 3 6
5 : 2 7
6 : 1 3 4 8
7 : 1 2 5
8 : 3 6
```

```
{1: [6, 7], 2: [5, 6, 7], 3: [4, 6, 8], 4: [3, 6], 5: [2, 7], 6: [1, 2, 3, 4, 8], 7: [1, 2, 5, 8], 8: [3, 6, 7]}
{1: {6: 8, 7: 2}, 2: {5: 2, 7: 3, 6: 3}, 3: {4: -4, 6: 5, 8: 3}, 4: {3: -4, 6: 5}, 5: {2: 2, 7: 6}, 6: {1: 8, 3: 5, 4: 5, 8: 90, 2: 3}, 7:
{1: 2, 2: 3, 5: 6, 8: 3}, 8: {3: 3, 6: 90, 7: 3}}
{(1, 6): 8, (1, 7): 2, (2, 5): 2, (2, 7): 3, (3, 4): -4, (3, 6): 5, (3, 8): 3, (4, 6): 5, (5, 7): 6, (6, 8): 90, (2, 6): 3, (8, 7): 3}
```

Lista następników:

```
1 : 6 7
2 : 5 6 7
3 : 4 6 8
4 : 3 6
5 : 2 7
6 : 1 2 3 4 8
7 : 1 2 5 8
8 : 3 6 7
```