



POLITECHNIKA RZESZOWSKA

im. Ignacego Łukasiewicza

WYDZIAŁ MATEMATYKI I FIZYKI STOSOWANEJ

Tomasz Nowak

Grupa P05

**Projekt Algorytmy i struktury danych nr. 3**

Rzeszów 2023

## 1. Wstęp

Na repozytorium trzymam swoją całą pracę: <https://github.com/Tnovyloo/Projekt-Studia>

Założeniem zadania projektu było stworzenie grafu w języku programowania oraz stworzenia funkcji do odczytywania informacji z grafu. Podpunkty zadania były następujące:

- Wypisz wszystkich sąsiadów dla każdego wierzchołka grafu (sąsiad wierzchołka  $w_1$  to ten wierzchołek do którego prowadzi krawędź z  $w_1$ )
- Wypisz wszystkie wierzchołki, które są sąsiadami każdego wierzchołka
- Wypisz stopnie wychodzące wszystkich wierzchołków
- Wypisz stopnie wchodzące wszystkich wierzchołków
- Wypisz wszystkie wierzchołki izolowane
- Wypisz wszystkie pętle
- Wypisz wszystkie krawędzie dwukierunkowe

## 2. Pseudokod dla funkcji

Funkcja\_zadanie1 (graf → lista krawędzi grafu)

Sasiedzi = []

Dla K w graf

Lista = []

W1 = K1[0]

Dla K1 w graf

Jeśli W1 w K1[0]

Lista += K1[1]

Funkcja\_zadanie2(graf → lista krawędzi grafu)

K\_ilość = ilość krawędzi

W\_ilość = []

Dla K w graf

W1 = K[0]

W2 = K[1]

Jeśli W1 nie jest w W\_ilość

W\_ilość += W1

Jeśli W2 nie jest w W\_ilość

Lista= []

Dla K w graf

Lista1 = []

W1 = K[1]

Dla K1 w graf

Jeśli K1[1] == W1

Jeśli K1[0] nie ma w Lista1

Lista1 += K1[0]

Lista += Lista1

Wypisz Lista

Funkcja\_zadanie3(graf → lista krawędzi grafu)

Lista = []

Dla K w graf

W1 = K[0]

Lista1 = []

Dla K1 w graf

Jeśli W1 w K1[0]

Lista1 += K1

Lista += Lista1

Wypisz Lista

Funkcja\_zadanie4(graf → lista krawędzi grafu)

Lista = []

Dla K w graf

W1 = K[1]

Lista1 = []

Dla K1 w graf

Jeśli W1 w K1[1]

Lista1 += K1

Lista += Lista1

Wypisz Lista

Funkcja\_zadanie5(graf -> lista krawędzi grafu)

Funkcja\_zadanie5(graf → lista krawędzi grafu)

Lista = []

Dla K w graf

Lista1 = []

W1 = K[0]

W2 = K[1]

Jeśli W1 == W2

Dla K1 w graf

Jeśli K1[0] == W1 lub K1[1] == W2 to Fałsz

Lista1 += K

Lista += Lista1

Funkcja\_zadanie6(graf → słownik)

Lista = []

Dla K w graf

Jeśli K[0] == K[1]

Lista += K

Funkcja\_zadanie7(graf → słownik)

Lista = []

Dla K w graf

K\_odw = [K[1], K[0]]

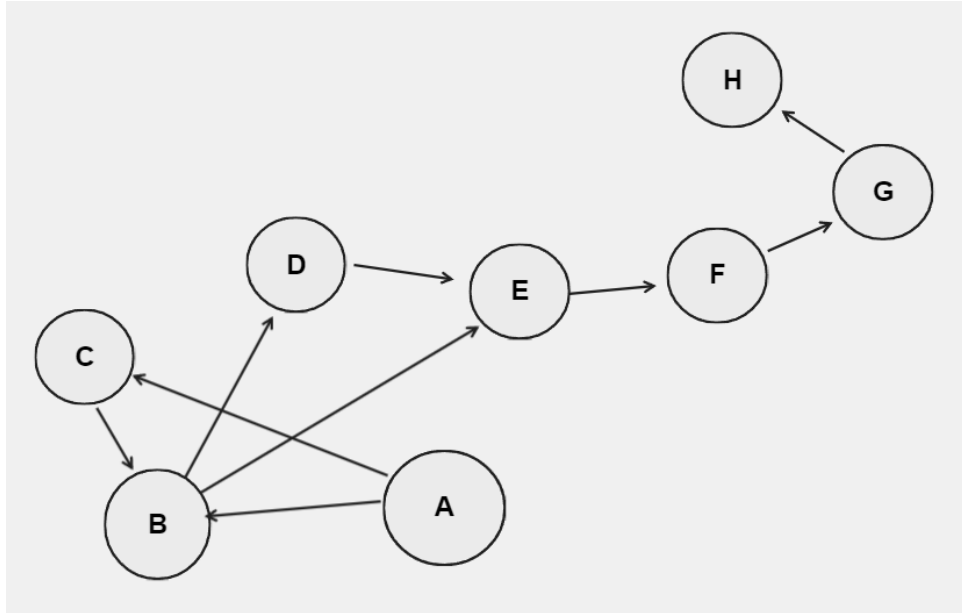
Jeśli K\_odw w graf

Lista += K

### 3. Program

Na repozytorium trzymam swoją całą pracę: <https://github.com/Tnovyloo/Projekt-Studia>

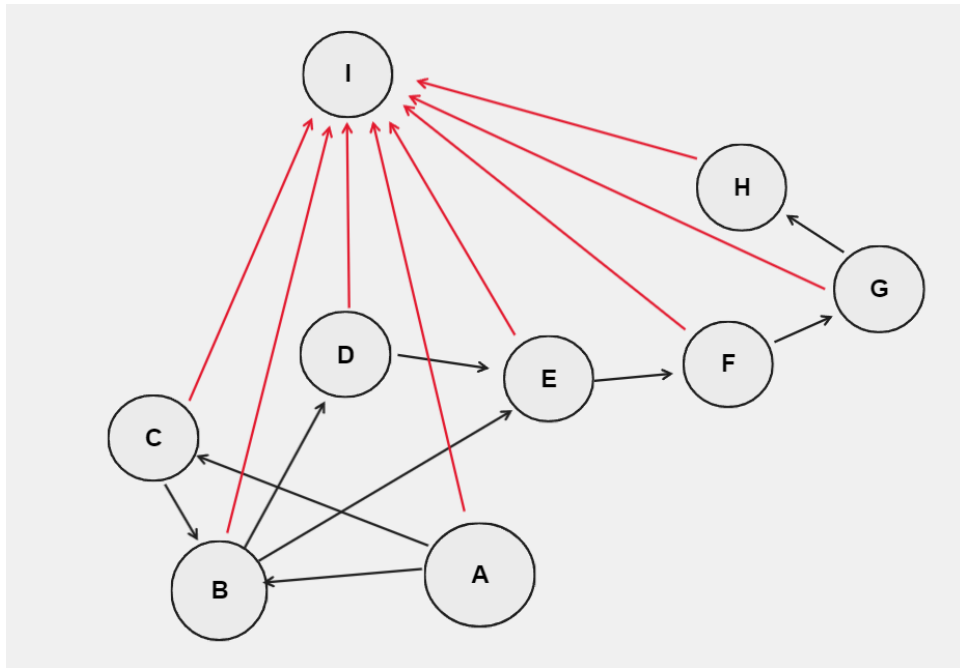
Zadanie 1 dla Grafu:



#### Zadanie 1

```
['Wierzchołek A ma sąsiadów:', ['B', 'C']]
['Wierzchołek B ma sąsiadów:', ['D', 'E']]
['Wierzchołek C ma sąsiadów:', ['B']]
['Wierzchołek D ma sąsiadów:', ['E']]
['Wierzchołek E ma sąsiadów:', ['F']]
['Wierzchołek F ma sąsiadów:', ['G']]
['Wierzchołek G ma sąsiadów:', ['H']]
```

Zadanie 2 dla grafu:



```
Zadanie 2
['Wierzchołek I ma sąsiadów', ['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H']]
```

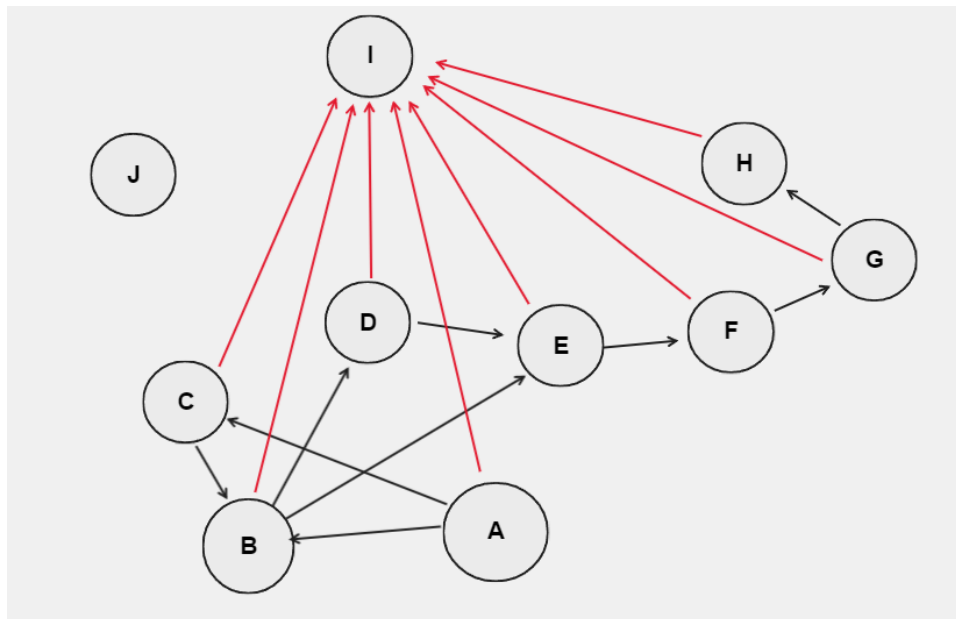
Zadanie 3 dla powyższego grafu:

```
Zadanie 3
['Stopnie wychodzące dla A:', [['A', 'B'], ['A', 'C'], ['A', 'I']]]
['Stopnie wychodzące dla B:', [['B', 'D'], ['B', 'E'], ['B', 'I']]]
['Stopnie wychodzące dla C:', [['C', 'B'], ['C', 'I']]]
['Stopnie wychodzące dla D:', [['D', 'E'], ['D', 'I']]]
['Stopnie wychodzące dla E:', [['E', 'F'], ['E', 'I']]]
['Stopnie wychodzące dla F:', [['F', 'G'], ['F', 'I']]]
['Stopnie wychodzące dla G:', [['G', 'H'], ['G', 'I']]]
['Stopnie wychodzące dla H:', [['H', 'I']]]
```

Zadanie 4 dla powyższego grafu:

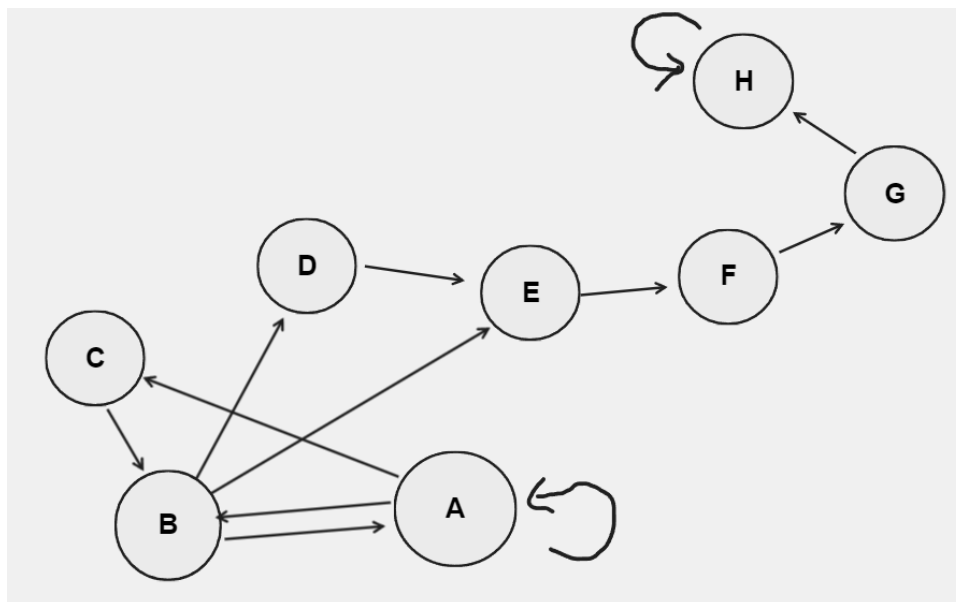
```
Zadanie 4
Wchodzace stopnie do wierzchołka A to []
Wchodzace stopnie do wierzchołka B to ['A', 'C']
Wchodzace stopnie do wierzchołka C to ['A']
Wchodzace stopnie do wierzchołka D to ['B']
Wchodzace stopnie do wierzchołka E to ['B', 'D']
Wchodzace stopnie do wierzchołka F to ['E']
Wchodzace stopnie do wierzchołka G to ['F']
Wchodzace stopnie do wierzchołka H to ['G']
Wchodzace stopnie do wierzchołka I to ['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H']
```

Zadanie 5 dla grafu:



Zadanie 5  
Wierzchołki izolowane to: J

Zadanie 6 dla grafu:



Zadanie 6  
Pętle ['A', 'A']  
Pętle ['H', 'H']



Zadanie 7 dla powyższego grafu:

Zadanie 7

Krawędź ['A', 'B'] i ['B', 'A']

Krawędź ['B', 'A'] i ['A', 'B']