

Politechnika Rzeszowska
Wydział Matematyki i Fizyki Stosowanej

Projekt III
GRAF SKIEROWANY REPREZENTOWANY PRZY
POMOCY MACIERZY SĄSIEDZTWA

Krzysztof Motas
Grupa projektowa nr 5

Sokolniki, 9 stycznia 2022

1 Opis problemu

Treść problemu:

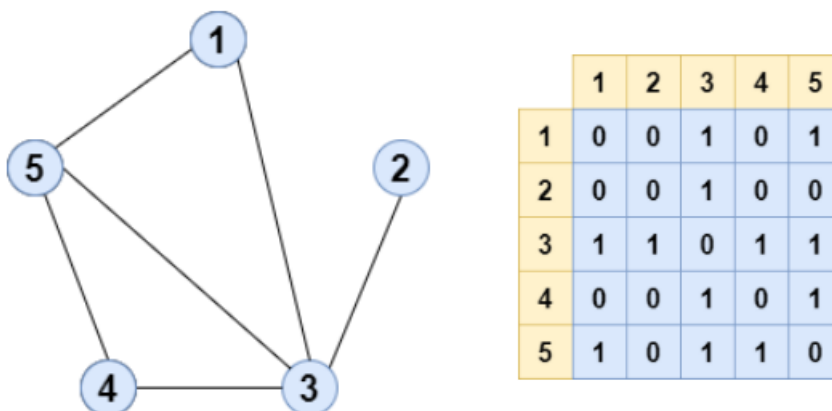
Napisz program, który dla zadanego grafu skierowanego reprezentowanego przy pomocy macierzy sąsiedztwa wyznaczy i wypisze następujące informacje:

1. wszystkich sąsiadów dla każdego wierzchołka grafu
2. wszystkie wierzchołki, które są sąsiadami każdego wierzchołka
3. stopnie wychodzące wszystkich wierzchołków
4. stopnie wchodzące wszystkich wierzchołków
5. wszystkie wierzchołki izolowane
6. wszystkie pętle
7. wszystkie krawędzie dwukierunkowe

Każdy z powyższych podpunktów powinien być realizowany jako oddzielna funkcja. W funkcji `main()` należy przedstawić działanie napisanej przez siebie biblioteki na reprezentatywnym przykładzie. Kod powinien być opatrzony stosownymi komentarzami.

2 Teoretyczne podstawy macierzy sąsiedztwa

Graf reprezentowany jest za pomocą macierzy kwadratowej A o stopniu n , gdzie n oznacza liczbę wierzchołków w grafie. Macierz sąsiedztwa odwzorowuje połączenia wierzchołków krawędziami. Wiersze macierzy sąsiedztwa odwzorowują zawsze wierzchołki startowe krawędzi, a kolumny odwzorowują wierzchołki końcowe krawędzi. Komórka $A[i, j]$, która znajduje się w i -tym wierszu oraz j -tej kolumnie odwzorowuje krawędź łączącą wierzchołek startowy $v[i]$ z wierzchołkiem końcowym $v[j]$. Jeśli $A[i, j]$ ma wartość 1 to dana krawędź istnieje. Jeżeli $A[i, j]$ ma wartość 0, to wierzchołek $v[i]$ nie łączy się krawędzią z wierzchołkiem $v[j]$.



Zaletą takiej reprezentacji grafu jest to, że potrafimy określić czy dwa wierzchołki sąsiadują ze sobą zawsze w takim samym czasie. Podobnie jest w przypadku dodawania lub usuwania krawędzi. Niepodważalną zaletą jest także prostota implementacji. Natomiast wadą jest złożoność pamięciowa. Rozmiar macierzy sąsiedztwa to V^2 . Załóżmy, że istnieje graf z 1000 wierzchołków i jedną krawędzią. Do przedstawienia tego grafu w wersji macierzowej, używamy tablicy o rozmiarze 1000^2 do przechowywania jednej krawędzi, co jest dosyć dużym marnowaniem pamięci.

3 Działanie biblioteki

3.1 Funkcja wyznaczająca wszystkich sąsiadów dla każdego wierzchołka grafu.

Należy wyznaczyć indeksy sąsiadów każdego wierzchołka i -tego, znajdziemy w i -tym wierszu, jeśli wartość pola jest niezerowa. (GetNeighborsOfAllVertices)

```
wczytaj(adjacencyMatrix)
wczytaj(numVertices)

i <- 0

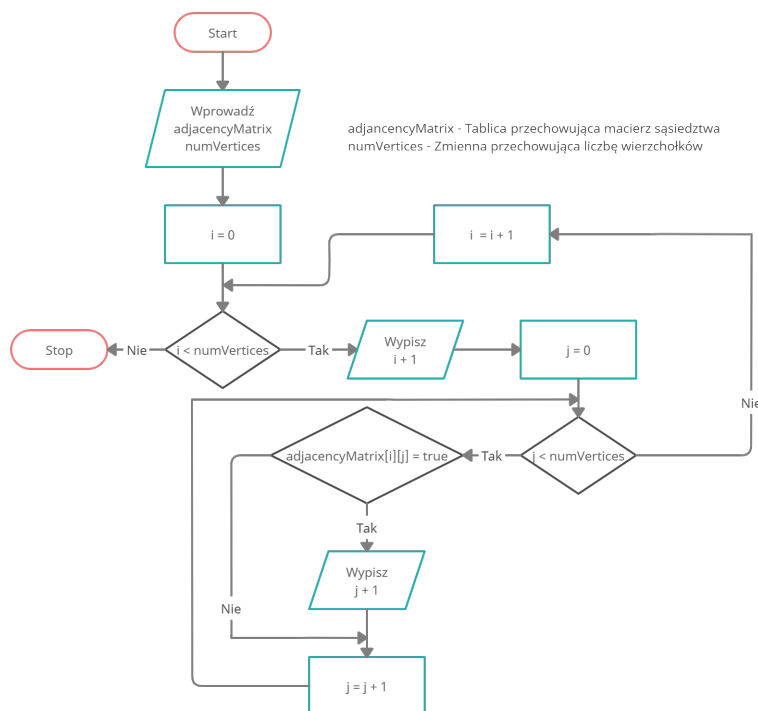
dopoki i < numVertices wykonuj
  wypisz(i + 1, ":")

  j <- 0

  dopoki j < numVertices wykonuj
    jezeli adjacencyMatrix[i][j] == true wykonaj
      wypisz(j + 1)

    j <- j + 1

  i <- i + 1
```



3.2 Funkcja wyznaczająca wszystkie wierzchołki, które są sąsiadami każdego wierzchołka.

Należy wyznaczyć liczbę sąsiadów dla każdego wierzchołka, a następnie sprawdzić, czy wartość jest równa liczbie wierzchołków macierzy. (GetVerticesThatAreNeighborsOfEachVertex)

```
wczytaj(adjacencyMatrix)
wczytaj(numVertices)

i <- 0

dopoki i < numVertices wykonuj
  counter <- 0
  j <- 0
```

```

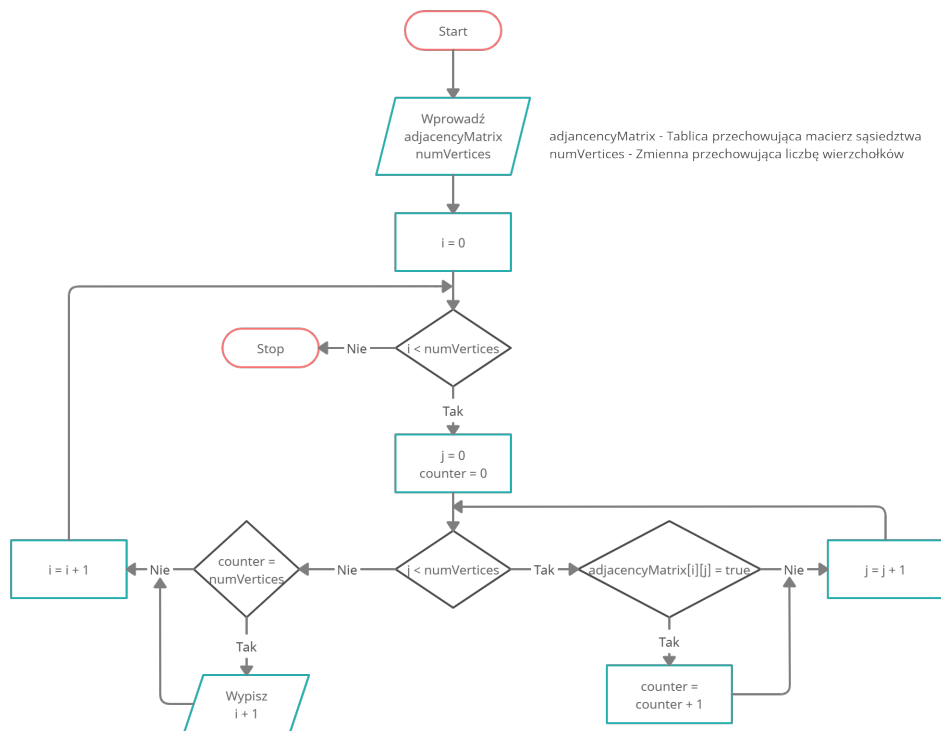
dopoki j < numVertices wykonuj
  jezeli adjacencyMatrix[i][j] wykonaj
    counter <- counter + 1

  j <- j + 1

jezeli numVertices == counter wykonaj
  wypisz(i + 1)

i <- i + 1

```



3.3 Funkcja wyznaczająca stopnie wychodzące wszystkich wierzchołków.

Należy wyznaczyć liczbę krawędzi wychodzących z każdego wierzchołka (suma wyrazów w wierszu macierzy). (GetOutDegrees)

```

wczytaj(adjacencyMatrix)
wczytaj(numVertices)

i <- 0

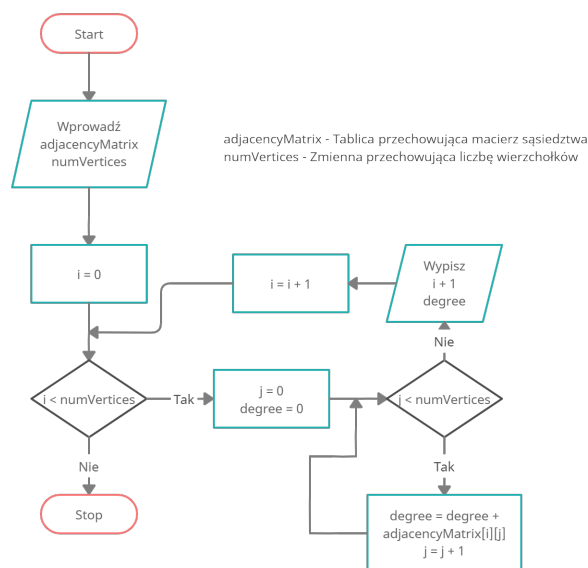
dopoki i < numVertices wykonuj
  degree <- 0
  j <- 0

  dopoki j < numVertices wykonuj
    degree <- degree + adjacencyMatrix[i][j]
    j <- j + 1

  wypisz(i + 1, degree)

i <- i + 1

```



3.4 Funkcja wyznaczająca stopnie wchodzące wszystkich wierzchołków.

Należy wyznaczyć liczbę krawędzi wchodzących do każdego wierzchołka (suma wyrazów w kolumnie macierzy). (GetInDegrees)

```

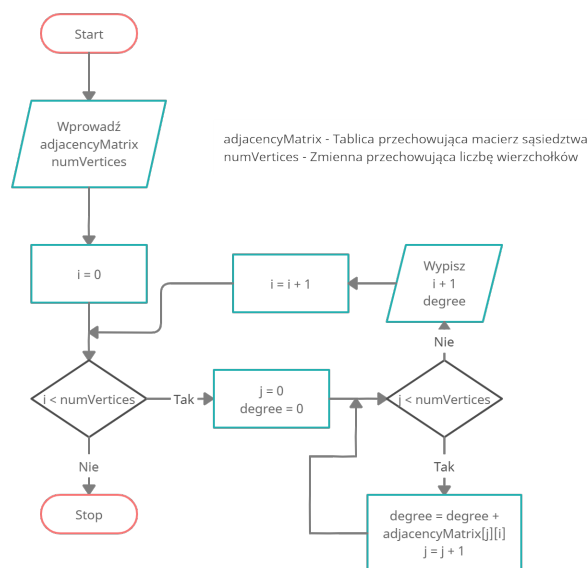
wczytaj(adjacencyMatrix)
wczytaj(numVertices)

i <- 0

dopoki i < numVertices wykonuj
  degree <- 0
  j <- 0
  dopoki j < numVertices wykonuj
    degree <- degree + adjacencyMatrix[j][i]
    j <- j + 1

  wypisz(i + 1, degree)

  i <- i + 1
  
```



3.5 Funkcja wyznaczająca wierzchołki izolowane.

Należy wyznaczyć sumę wyrazów i -tego wiersza i i -tej kolumny. Jeśli jest ona równa zero, to i -ty wierzchołek jest izolowany. Sprawdzenie należy wykonać dla każdego wierzchołka. (GetIsolatedVertices)

```
wczytaj(adjacencyMatrix)
wczytaj(numVertices)

i <- 0

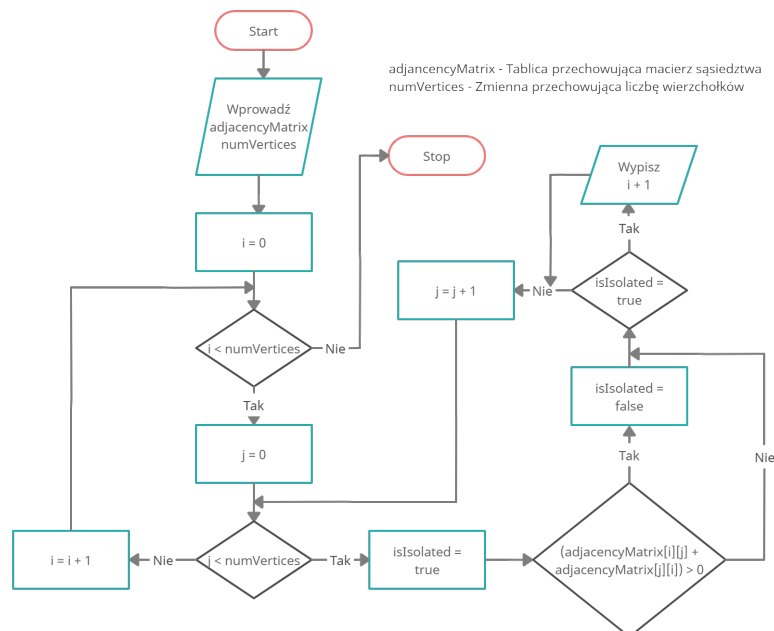
dopoki i < numVertices wykonuj
  isIsolated <- true
  j <- 0

  dopoki j < numVertices wykonuj
    jezeli (adjacencyMatrix[i][j] + adjacencyMatrix[j][i]) > 0 wykonaj
      isIsolated <- false

  j <- j + 1

  jezeli isIsolated == true wykonaj
    wypisz(i + 1)

  i <- i + 1
```



3.6 Funkcja wyznaczająca wszystkie pętle.

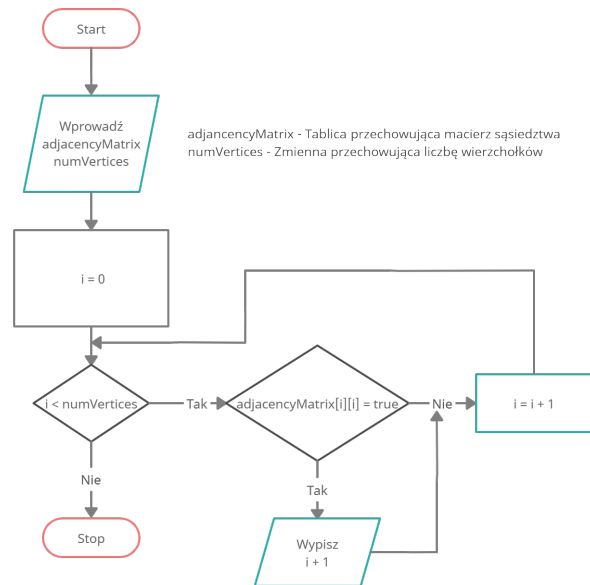
Należy zsumować elementy głównej przekątnej macierzy sąsiedztwa. (GetLoops)

```
wczytaj(adjacencyMatrix)
wczytaj(numVertices)

i <- 0

dopoki i < numVertices wykonuj
  jezeli adjacencyMatrix[i][i] == true wykonaj
    wypisz(i + 1)

  i <- i + 1
```



3.7 Funkcja wyznaczająca wszystkie krawędzie dwukierunkowe.

Krawędzie dwukierunkowe występują, gdy dwa wierzchołki są wzajemnie swoimi sąsiadami. Czyli od jednego biegnie krawędź do drugiego, a od drugiego biegnie krawędź do pierwszego. (GetBiDirectionalEdges)

```

wczytaj(adjacencyMatrix)
wczytaj(numVertices)

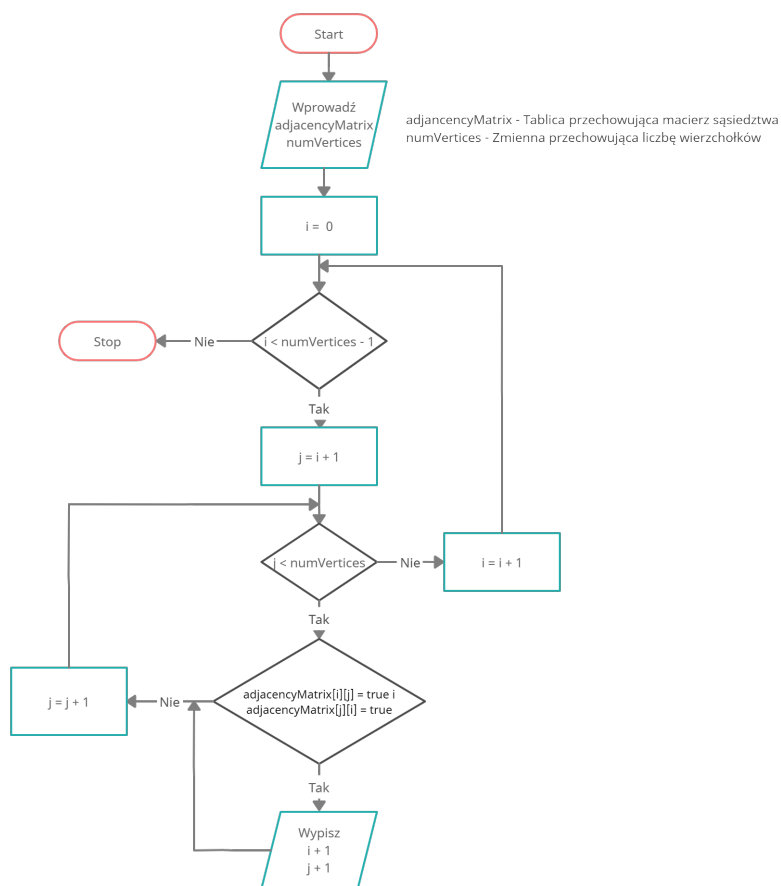
i <- 0

dopoki i < numVertices - 1 wykonuj
  j <- i + 1

  dopoki j < numVertices wykonuj
    jeżeli adjacencyMatrix[i][j] == true && adjacencyMatrix[j][i] == true wykonaj
      wypisz(i + 1, j + 1)

    j <- j + 1

  i <- i + 1
  
```



4 Przykłady działania biblioteki

4.1 Przykład 1

Macierz sąsiedztwa:

$$\begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Wszyscy sąsiedzi dla każdego wierzchołka grafu:

1: 2 3
2: 1 3
3: 1 2 4
4: 3

Nie znaleziono żadnego wierzchołka, który jest sąsiadem każdego wierzchołka!

Stopnie wychodzące wszystkich wierzchołków:

1: 2
2: 2
3: 3
4: 1

Stopnie wchodzące wszystkich wierzchołków:

1: 2
2: 2

3: 3

4: 1

Żaden wierzchołek nie jest izolowany!

Nie znaleziono żadnych pętli!

Lista krawędzi dwukierunkowych:

1: 2

1: 3

2: 3

3: 4

4.2 Przykład 2

Macierz sąsiedztwa:

$$\begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Wszyscy sąsiedzi dla każdego wierzchołka grafu:

1: 3 4 9

2: 6 10

3: 1 6 7 8

4: 1 5 7 9

5: 4 10

6: 2 3

7: 3 4 8 9

8: 3 7 10

9: 1 4 7

10: 2 5 8

Nie znaleziono żadnego wierzchołka, który jest sąsiadem każdego wierzchołka!

Stopnie wychodzące wszystkich wierzchołków:

1: 3

2: 2

3: 4

4: 4

5: 2

6: 2

7: 4

8: 3

9: 3

10: 3

Stopnie wchodzące wszystkich wierzchołków:

1: 3

2: 2

3: 4

4: 4

5: 2

6: 2
7: 4
8: 3
9: 3
10: 3

Żaden wierzchołek nie jest izolowany!

Nie znaleziono żadnych pętli!

Lista krawędzi dwukierunkowych:

1: 3
1: 4
1: 9
2: 6
2: 10
3: 6
3: 7
3: 8
4: 5
4: 7
4: 9
5: 10
7: 8
7: 9
8: 10

5 Linki

5.1 GitHub

https://github.com/kmotas/aisd_projekt3

5.2 Źródła

<https://eduinf.waw.pl>

https://en.wikipedia.org/wiki/Adjacency_matrix