

Politechnika Śląska  
Wydział Informatyki, Elektroniki i Informatyki

**Podstawy Programowania  
Komputerów**

Graf Dwudzielny

autor	Szymon Czech
prowadzący	dr Inż. Marek Kokot
rok akademicki	2020/2021
kierunek	Informatyka
rodzaj studiów	SSI
semestr	1
termin laboratorium	poniedziałek, 12:15 – 14:30 środa, 15:30 – 17:45
sekcja	4-2
termin oddania sprawozdania	2021-02-04

# 1 Treść zadania

Napisać program, do sprawdzania, czy graf nieskierowany jest dwudzielny. Plik z grafem ma następującą postać:

- Każda krawędź jest podana w osobnej linii; podane są dwa wierzchołki, które łączy krawędź.
- W pliku mogą wystąpić puste linie.
- W linii mogą wystąpić dodatkowe (nadmiarowe) znaki białe.

Przykładowy plik z grafem:

```
0 1
2 0
1    3
3 0
```

Program wypisuje do pliku wyjściowego zadany graf i komunikat, czy jest to graf dwudzielny, czy nie. Jeżeli zadany graf jest dwudzielny, program wypisuje wierzchołki obu grup grafu. Program uruchamiany jest z linii poleceń z wykorzystaniem następujących przełączników:

-i plik wejściowy z krawędziami grafu

-o plik wyjściowy z wynikami

Natomiast uruchomienie programu bez odpowiednich przełączników powoduje wypisanie krótkiej instrukcji.

## 2 Analiza zadania

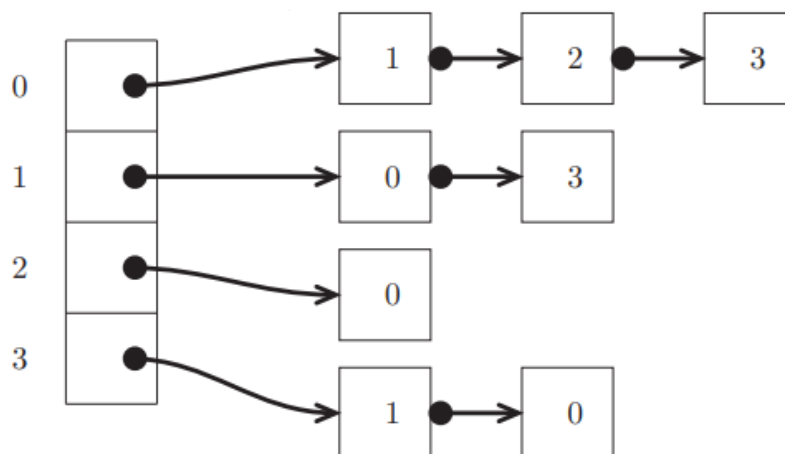
Zagadnienie przedstawia problem przejścia przez wszystkie wierzchołki grafu, które pobieramy z pliku tekstowego i podzielenie ich na dwa rozłączne zbiory tak, aby krawędzie nie łączyły wierzchołków tego samego zbioru.

### 2.1 Struktury danych

W programie została wykorzystana utworzona przeze mnie klasa Graf która przechowuje:

- Wierzchołki grafu reprezentowane jako lista sąsiedztwa
- Metody pozwalające na operowanie na wierzchołkach grafu jak np., czytanie wierzchołków grafu z pliku tekstowego czy sprawdzanie dwudzielności.

Struktura danych przechowująca wierzchołki grafu to lista sąsiedztwa. Do jej utworzenia wykorzystałem w programie wektor list.



## 2.2 Algorytmy

Program wykorzystuje przeszukiwanie grafu w głąb w celu przejścia przez wszystkie jego wierzchołki oraz przypisuje im odpowiednie wartości dzieląc je na dwie grupy w celu sprawdzenia czy graf jest dwudzielny.

## 3 Specyfikacja zewnętrzna

Program jest uruchamiany z linii poleceń. Należy przekazać do programu nazwy plików: wejściowego i wyjściowego po odpowiednich przełącznikach (odpowiednio: -i dla pliku wejściowego i -o dla pliku wyjściowego), np.

```
Zadanie.exe -i wierzchołki.txt -o czydwudzielny.txt
```

Jeśli użytkownik poda błędne parametry lub program nie będzie mógł otworzyć pliku wejściowego to na ekranie wyświetli się komunikat o błędzie oraz instrukcja jak należy wprowadzić dane.

## 4 Specyfikacja wewnętrzna

### 4.1 Ogólna struktura programu

W funkcji głównej na początku program sprawdza czy podane przez użytkownika parametry uruchomienia programu są poprawne. Gdy program nie został wywołany prawidłowo, zostaje wypisany stosowny komunikat oraz instrukcja jak należy uruchomić program. Tworzony jest obiekt klasy *Graf* oraz wektor *kolor* informujący do której grupy należą wierzchołki grafu. Metoda *czytaj\_graf* wczytuje z pliku tekstowego wierzchołki grafu oraz zapisuje je do listy sąsiedztwa i metoda *wypisz\_graf* wypisuje je na ekran oraz do pliku wyjściowego. Następnie metoda *czy\_graf\_dwudzielny* sprawdza czy graf składa się z więcej niż jednego wierzchołka oraz przypisuje odpowiednie wartości *kolor* oraz *odwiedzzone* (informacja czy wierzchołek został już odwiedzony) dla pierwszego z nich. Metoda wywołuje w sobie oraz zwraca wartość prywatnej metody *DFS\_czy\_dwudzielny*. Metoda ta rekurencyjnie

przechodzi po wszystkich wierzchołkach grafu sprawdzając czy da się je podzielić na dwie grupy (dwudzielność) i zwraca „true” jeśli jest dwudzielny albo „false” jeśli nie. Następnie jeśli graf jest dwudzielny funkcja *grupuj\_wierzchołki\_grafu\_dwudzielnego* wpisuje je do dwóch wektorów oraz zwraca je jako parę. Ostatnimi dwoma funkcjami są *pokaż\_wyniki* oraz *zapisz\_wyniki*, które odpowiednio pokazują wyniki wykonanych operacji na ekran oraz zapisują je do pliku wyjściowego.

## 4.2 Szczegółowy opis typów i funkcji

Szczegółowy opis typów i funkcji zawarty jest w dokumentacji i komentarzach (doxygen).

## 5 Testowanie

Program został przetestowany na różnego rodzaju plikach w celu sprawdzenia czy wyniki wykonywanych przez niego operacji są prawidłowe. W przypadku błędów na ekranie pojawiają się stosowne komunikaty.

## 6 Wnioski

Zadanie mimo początkowych dużych trudności zadanie udało mi się zrealizować choć myślę, że można je jeszcze znacząco poprawić. Najtrudniejsze okazało się zrozumienie i napisanie algorytmu przeszukiwania w głąb tak aby poprawnie poruszał się po wierzchołkach i spełniał swoją funkcję. Mimo problemów, po dokładnej analizie oraz informacji z różnych źródeł internetowych oraz książek udało mi się ukończyć zadanie. W trakcie tworzenia programu poszerzyłem swoją wiedzę z zakresu wektorów a także dowiedziałem się co to są klasy i do czego służą oraz poznałem różnice między metodami i atrybutami publicznymi a prywatnymi. Pomyślne ukończenie zadania sprawiło mi wiele satysfakcji.

Graf Dwudzielny

Wygenerowano przez Doxygen 1.9.1



<b>1 Indeks klas</b>	<b>1</b>
1.1 Lista klas	1
<b>2 Indeks plików</b>	<b>3</b>
2.1 Lista plików	3
<b>3 Dokumentacja klas</b>	<b>5</b>
3.1 Dokumentacja klasy Graf	5
3.1.1 Opis szczegółowy	5
3.1.2 Dokumentacja funkcji składowych	6
3.1.2.1 czy_graf_dwudzielny()	6
3.1.2.2 czytaj_graf()	6
3.1.2.3 DFS_czy_dwudzielny()	6
3.1.2.4 liczba_wierzchołkow()	7
<b>4 Dokumentacja plików</b>	<b>9</b>
4.1 Dokumentacja pliku C:/Users/szymo/source/repos/7133f719-gr42-repo/projekt/ZadanieProgramistyczne/↵ ZadanieProgramistyczne/Graf.h	9
4.2 Dokumentacja pliku C:/Users/szymo/source/repos/7133f719-gr42-repo/projekt/ZadanieProgramistyczne/↵ ZadanieProgramistyczne/OperacjeNaWierzchołkach.h	9
4.2.1 Dokumentacja funkcji	9
4.2.1.1 grupuj_wierzchołki_grafu_dwudzielnego()	9
4.3 Dokumentacja pliku C:/Users/szymo/source/repos/7133f719-gr42-repo/projekt/ZadanieProgramistyczne/↵ ZadanieProgramistyczne/OperacjeWyjscia.h	10
4.3.1 Dokumentacja funkcji	10
4.3.1.1 pokaz_wyniki()	10
4.3.1.2 wypisz_graf()	11
4.3.1.3 zapisz_wyniki()	11
<b>Indeks</b>	<b>13</b>





# Rozdział 1

## Indeks klas

### 1.1 Lista klas

Tutaj znajdują się klasy, struktury, unie i interfejsy wraz z ich krótkimi opisami:

<a href="#">Graf</a>	Klasa reprezentująca obiekt <a href="#">Graf</a> . . . . .	<a href="#">5</a>
----------------------	--	-------------------



## Rozdział 2

# Indeks plików

### 2.1 Lista plików

Tutaj znajduje się lista wszystkich udokumentowanych plików z ich krótkimi opisami:

C:/Users/szymo/source/repos/7133f719-gr42-repo/projekt/ZadanieProgramistyczne/ZadanieProgramistyczne/[Graf.h](#)  
9

C:/Users/szymo/source/repos/7133f719-gr42-repo/projekt/ZadanieProgramistyczne/ZadanieProgramistyczne/[OperacjeNaWierchochach.h](#)  
9

C:/Users/szymo/source/repos/7133f719-gr42-repo/projekt/ZadanieProgramistyczne/ZadanieProgramistyczne/[OperacjeWyjscia.h](#)  
10



## Rozdział 3

# Dokumentacja klas

### 3.1 Dokumentacja klasy Graf

Klasa reprezentująca obiekt [Graf](#).

```
#include <Graf.h>
```

#### Metody publiczne

- bool [czytaj\\_graf](#) (std::string plik\_wejscowy)

*Funkcja wczytująca graf z pliku.*

- bool [czy\\_graf\\_dwudzielny](#) (std::vector< int > &kolor)

*Funkcja rozpoczyna sprawdzanie dwudzielności grafu. Funkcja sprawdza czy graf posiada więcej niż jeden wierzchołek. Jeśli nie, operacja sprawdzania dwudzielności jest zakończona. Następnie w funkcji tworzona jest tablica typu vector "odwiedzone" (o wielkości takiej jaka jest liczba wierzchołków) reprezentująca informacje czy wierzchołek został już odwiedzony, oraz pierwszemu wierzchołkowi są przypisywane odpowiednie wartości: odwiedzone = true(wierzchołek został odwiedzony; kolor = 1(należy do pierwszej grupy);.*

- int [liczba\\_wierzchołkow](#) ()

*Funkcja określa liczbę wierzchołków grafu.*

#### Metody prywatne

- bool [DFS\\_czy\\_dwudzielny](#) (int v, std::vector< bool > &odwiedzone, std::vector< int > &kolor)

*Funkcja sprawdzająca dwudzielność grafu poprzez przeszukiwanie w głąb.*

#### Atrybuty prywatne

- std::vector< std::list< int > > [lista\\_sasiedztwa](#)

*Struktura danych zawierająca wierzchołki grafu.*

#### 3.1.1 Opis szczegółowy

Klasa reprezentująca obiekt [Graf](#).

Klasa zawiera dane z wierzchołkami grafu, oraz metody pozwalające na wykonywanie operacji na tych wierzchołkach.

### 3.1.2 Dokumentacja funkcji składowych

#### 3.1.2.1 czy\_graf\_dwudzielny()

```
bool Graf::czy_graf_dwudzielny (
    std::vector< int > & kolor )
```

Funkcja rozpoczyna sprawdzanie dwudzielności grafu. Funkcja sprawdza czy graf posiada więcej niż jeden wierzchołek. Jeśli nie, operacja sprawdzania dwudzielności jest zakończona. Następnie w funkcji tworzona jest tablica typu vector "odwiedzone" (o wielkości takiej jaka jest liczba wierzchołków) reprezentująca informacje czy wierzchołek został już odwiedzony, oraz pierwszemu wierzchołkowi są przypisywane odpowiednie wartości: odwiedzone = true(wierzchołek został odwiedzony; kolor = 1(należy do pierwszej grupy);.

##### Parametry

<i>kolor</i>	tablica określająca "kolory" (grupy) poszczególnych wierzchołków, 1 jeśli należy do grupy pierwszej oraz 0 jeśli do grupy drugiej.
--------------	--

##### Zwraca

Wynik operacji przeszukiwania przeszukiwania w głąb grafu, "true" jeśli dwudzielny i "false" w przeciwnym wypadku.

#### 3.1.2.2 czytaj\_graf()

```
bool Graf::czytaj_graf (
    std::string plik_wejsciowy )
```

Funkcja wczytująca graf z pliku.

##### Parametry

<i>plik_wejsciowy</i>	plik_wejsciowy plik z wierzchołkami grafu.
-----------------------	--

##### Zwraca

"true" jeśli udało się wczytać wierzchołki, w przeciwnym wypadku zwraca "false".

#### 3.1.2.3 DFS\_czy\_dwudzielny()

```
bool Graf::DFS_czy_dwudzielny (
    int v,
```

```
std::vector< bool > & odwiedzone,  
std::vector< int > & kolor ) [private]
```

Funkcja sprawdzająca dwudzielność grafu poprzez przeszukiwanie w głąb.

Funkcja rekurencyjnie przechodzi po kolejnych wierzchołkach grafu, przypisuje im odpowiednie wartości oraz określa czy graf jest dwudzielny czy też nie.

#### Parametry

<i>v</i>	wierzchołek startowy od którego zaczyna się przeszukiwanie.
<i>odwiedzony</i>	odwiedzony tablica określająca czy wierzchołek został już odwiedzony(true) czy też nie(false).
<i>kolor</i>	kolor tablica określająca "kolory" (grupy) poszczególnych wierzchołków, 1 jeśli należy do grupy pierwszej oraz 0 jeśli do grupy drugiej.

#### Zwraca

"true" jeśli graf jest dwudzielny, w przeciwnym wypadku "false".

#### 3.1.2.4 liczba\_wierzchołkow()

```
int Graf::liczba_wierzchołkow ( )
```

Funkcja określa liczbę wierzchołków grafu.

#### Zwraca

zwraca rozmiar tablicy "lista\_sąsiedztwa".

Dokumentacja dla tej klasy została wygenerowana z plików:

- C:/Users/szymo/source/repos/7133f719-gr42-repo/projekt/ZadanieProgramistyczne/ZadanieProgramistyczne/[Graf.h](#)
- C:/Users/szymo/source/repos/7133f719-gr42-repo/projekt/ZadanieProgramistyczne/ZadanieProgramistyczne/Graf.[↔](#)cpp





## Rozdział 4

# Dokumentacja plików

### 4.1 Dokumentacja pliku

**C:/Users/szymo/source/repos/7133f719-gr42-repo/projekt/ZadanieProgramistyczne/ZadanieProgramistyczne/Graf.h**

```
#include <vector>
#include <list>
#include <string>
```

#### Komponenty

- class [Graf](#)

*Klasa reprezentująca obiekt [Graf](#).*

### 4.2 Dokumentacja pliku C:/Users/szymo/source/repos/7133f719-gr42-repo/projekt/ZadanieProgramistyczne/ZadanieProgramistyczne/OperacjeNaWierzchołkach.h

#### Funkcje

- `std::pair< std::vector< int >, std::vector< int > > grupuj_wierzcholki_grafu_dwudzielnego (const std::vector< int > &kolor)`

*Para zawierająca tablice typu wektor "l" zawierająca wierzchołki pierwszej grupy oraz "p" wierzchołki drugiej grupy.*

#### 4.2.1 Dokumentacja funkcji

##### 4.2.1.1 grupuj\_wierzcholki\_grafu\_dwudzielnego()

```
std::pair<std::vector<int>, std::vector<int> > grupuj_wierzcholki_grafu_dwudzielnego (
    const std::vector< int > & kolor )
```

Para zawierająca tablice typu wektor "l" zawierająca wierzchołki pierwszej grupy oraz "p" wierzchołki drugiej grupy.

## Parametry

<i>kolor</i>	tablica określająca "kolory"(grupy) poszczególnych wierzchołków, 1 jeśli należy do grupy pierwszej oraz 0 jeśli do grupy drugiej.
--------------	---

## 4.3 Dokumentacja pliku

**C:/Users/szymo/source/repos/7133f719-gr42-repo/projekt/ZadanieProgramistyczne/ZadanieProgramistyczne/OperacjeWyjscia.h**

### Funkcje

- void `pokaz_wyniki` (bool dwudzielny, std::pair< std::vector< int >, std::vector< int >> wierzcholki)  
*Funkcja pokazuje wyniki wykonywanych operacji na ekran.*
- void `zapisz_wyniki` (std::string plik\_wyjsciowy, bool dwudzielny, std::pair< std::vector< int >, std::vector< int >> wierzcholki)  
*Funkcja zapisuje wyniki wykonanych operacji do pliku wyjściowego.*
- bool `wypisz_graf` (std::string plik\_wejsciowy, std::string plik\_wyjsciowy)  
*Funkcja wypisuje zarówno do pliku jak i na ekran graf na którym operujemy.*
- void `Niewlasciwe_parametry` ()  
*Funkcja wyświetlająca odpowiedni komunikat gdy parametry potrzebne do uruchomienia programu, wprowadzone przez użytkownika są niepoprawne.*
- void `nie_otwarto_pliku` ()  
*Funkcja wyświetlająca podpowiedź dla użytkownika gdy nie udało się otworzyć pliku wejściowego.*

### 4.3.1 Dokumentacja funkcji

#### 4.3.1.1 pokaz\_wyniki()

```
void pokaz_wyniki (
    bool dwudzielny,
    std::pair< std::vector< int >, std::vector< int >> wierzcholki )
```

Funkcja pokazuje wyniki wykonywanych operacji na ekran.

Funkcja wyświetla wyniki sprawdzania dwudzielności grafu. Informuję czy jest on dwudzielny czy też nie i wydaje odpowiedni komunikat. W przypadku gdy graf jest dwudzielny wyświetlane są także wierzchołki należące do poszczególnych grup.

## Parametry

<i>dwudzielny</i>	zmienna przechowująca wyniki badania dwudzielności, "true" jeśli dwudzielny, "false" jeśli nie.
<i>wierzcholki</i>	para tablic zawierających wierzchołki pierwszej i drugiej grupy.

#### 4.3.1.2 wypisz\_graf()

```
bool wypisz_graf (
    std::string plik_wejscowy,
    std::string plik_wyjscowy )
```

Funkcja wypisuje zarówno do pliku jak i na ekran graf na którym operujemy.

Funkcja wyświetla zadany graf oraz zapisuje go pliku wyjściowego.

##### Parametry

<i>plik_wejscowy</i>	Plik wejściowy z którego są pobierane wierzchołki grafu.
<i>plik_wyjscowy</i>	Plik wyjściowy do którego wierzchołki grafu będą zapisywane.

##### Zwraca

#### 4.3.1.3 zapisz\_wyniki()

```
void zapisz_wyniki (
    std::string plik_wyjscowy,
    bool dwudzielny,
    std::pair< std::vector< int >, std::vector< int >> wierzcholki )
```

Funkcja zapisuje wyniki wykonanych operacji do pliku wyjściowego.

Funkcja zapisuje wyniki sprawdzania dwudzielności grafu. Zapisuje informacje czy jest on dwudzielny czy też nie. W przypadku gdy graf jest dwudzielny zapisywane są także wierzchołki należące do poszczególnych grup.

##### Parametry

<i>plik_wyjscowy</i>	plik do którego zostaną zapisane wyniki operacji.
<i>dwudzielny</i>	zmienna przechowująca wyniki badania dwudzielności, "true" jeśli dwudzielny, "false" jeśli nie.
<i>wierzcholki</i>	para tablic zawierających wierzchołki pierwszej i drugiej grupy.



# Indeks

C:/Users/szymo/source/repos/7133f719-gr42-repo/projekt/ZadanieProgramistyczne/ZadanieProgramistyczne/Graf.h,  
9

C:/Users/szymo/source/repos/7133f719-gr42-repo/projekt/ZadanieProgramistyczne/ZadanieProgramistyczne/OperacjeNaWierzchołkach.h,  
9

C:/Users/szymo/source/repos/7133f719-gr42-repo/projekt/ZadanieProgramistyczne/ZadanieProgramistyczne/OperacjeWyjscia.h,  
10

czy\_graf\_dwudzielny  
Graf, 6

czytaj\_graf  
Graf, 6

DFS\_czy\_dwudzielny  
Graf, 6

Graf, 5

    czy\_graf\_dwudzielny, 6

    czytaj\_graf, 6

    DFS\_czy\_dwudzielny, 6

    liczba\_wierzchołkow, 7

grupuj\_wierzchołki\_grafu\_dwudzielnego  
OperacjeNaWierzchołkach.h, 9

liczba\_wierzchołkow  
Graf, 7

OperacjeNaWierzchołkach.h

    grupuj\_wierzchołki\_grafu\_dwudzielnego, 9

OperacjeWyjscia.h

    pokaz\_wyniki, 10

    wypisz\_graf, 10

    zapisz\_wyniki, 11

pokaz\_wyniki  
OperacjeWyjscia.h, 10

wypisz\_graf  
OperacjeWyjscia.h, 10

zapisz\_wyniki  
OperacjeWyjscia.h, 11