

Szukanie cyklu w grafie skierowanym

Wygenerowano przez Doxygen 1.9.6



<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 struktury krawedz	5
3.1.1 Opis szczegółowy	5
3.1.2 Dokumentacja atrybutów składowych	5
3.1.2.1 wierz_koncowy	5
3.2 Dokumentacja struktury wierzcholek	6
3.2.1 Opis szczegółowy	6
3.2.2 Dokumentacja atrybutów składowych	6
3.2.2.1 krawedzie	6
<b>4 Dokumentacja plików</b>	<b>7</b>
4.1 Dokumentacja pliku Cykl.cpp	7
4.1.1 Dokumentacja funkcji	7
4.1.1.1 main()	8
4.2 Dokumentacja pliku Funkcje.cpp	8
4.2.1 Dokumentacja funkcji	9
4.2.1.1 czy_wektor_zawiera()	9
4.2.1.2 drukuj_graf()	10
4.2.1.3 int_ze_stringa()	10
4.2.1.4 odczytanie_z_pliku()	11
4.2.1.5 stworz_graf_ze_stringa()	12
4.2.1.6 szukanie_cykli()	12
4.2.1.7 wypisz_vector()	13
4.2.1.8 zapisz_cykle()	14
4.3 Dokumentacja pliku Funkcje.h	14
4.3.1 Dokumentacja funkcji	16
4.3.1.1 czy_wektor_zawiera()	16
4.3.1.2 drukuj_graf()	16
4.3.1.3 int_ze_stringa()	17
4.3.1.4 odczytanie_z_pliku()	18
4.3.1.5 stworz_graf_ze_stringa()	18
4.3.1.6 szukanie_cykli()	19
4.3.1.7 wypisz_vector()	20
4.3.1.8 zapisz_cykle()	20
4.4 Funkcje.h	21
4.5 Dokumentacja pliku Struktury.h	22
4.5.1 Dokumentacja definicji typów	23

4.5.1.1 Graf . . . . .	23
4.6 Struktury.h . . . . .	23
<b>Skorowidz</b>	<b>25</b>

# Rozdział 1

## Indeks klas

### 1.1 Lista klas

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

<a href="#">krawedz</a>	Typ przechowujący informacje o krawedzi grafu . . . . .	5
<a href="#">wierzcholek</a>	Typ przechowujący wczytane krawędzie grafu . . . . .	6



## Rozdział 2

# Indeks plików

### 2.1 Lista plików

Tutaj znajduje się lista wszystkich plików z ich krótkimi opisami:

<a href="#">Cykl.cpp</a>	7
<a href="#">Funkcje.cpp</a>	8
<a href="#">Funkcje.h</a>	14
<a href="#">Struktury.h</a>	22





## Rozdział 3

# Dokumentacja klas

### 3.1 Dokumentacja struktury krawedz

Typ przechowujący informacje o krawedzi grafu.

```
#include <Struktury.h>
```

#### Atrybuty publiczne

- int [wierz\\_koncowy](#)

#### 3.1.1 Opis szczegółowy

Typ przechowujący informacje o krawedzi grafu.

#### 3.1.2 Dokumentacja atrybutów składowych

##### 3.1.2.1 wierz\_koncowy

```
int krawedz::wierz_koncowy
```

Przechowuje wierzcholek koncowy krawedzi

Dokumentacja dla tej struktury została wygenerowana z pliku:

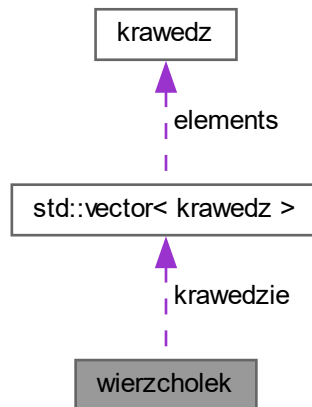
- [Struktury.h](#)

## 3.2 Dokumentacja struktury wierzcholek

Typ przechowujący wczytane krawędzie grafu.

```
#include <Struktury.h>
```

Diagram współpracy dla wierzcholek:



### Atrybuty publiczne

- `std::vector< krawedz > krawedzie`

### 3.2.1 Opis szczegółowy

Typ przechowujący wczytane krawędzie grafu.

### 3.2.2 Dokumentacja atrybutów składowych

#### 3.2.2.1 krawedzie

```
std::vector<krawedz> wierzcholek::krawedzie
```

Przechowuje krawedz z wierzchołkiem końcowym

Dokumentacja dla tej struktury została wygenerowana z pliku:

- [Struktury.h](#)

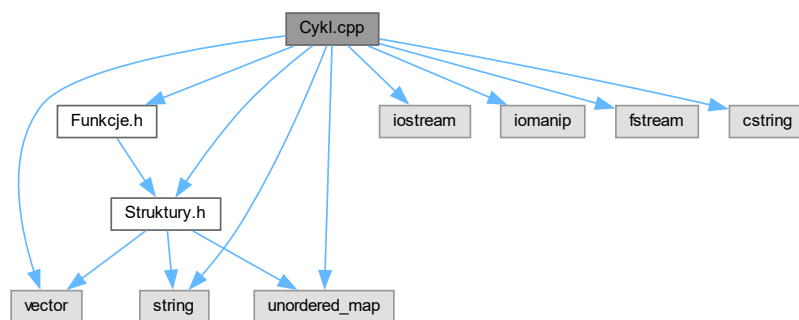
## Rozdział 4

# Dokumentacja plików

### 4.1 Dokumentacja pliku Cykl.cpp

```
#include <vector>
#include <string>
#include <iostream>
#include <iomanip>
#include <fstream>
#include <unordered_map>
#include "Funkcje.h"
#include "Struktury.h"
#include <cstring>
```

Wykres zależności załączania dla Cykl.cpp:



### Funkcje

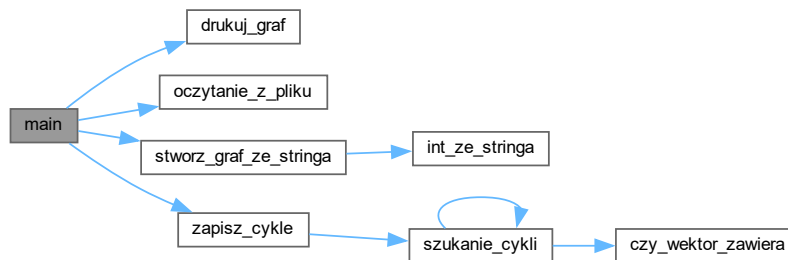
- int `main` (int argc, char \*argv[])

#### 4.1.1 Dokumentacja funkcji

#### 4.1.1.1 main()

```
int main (
    int argc,
    char * argv[] )
```

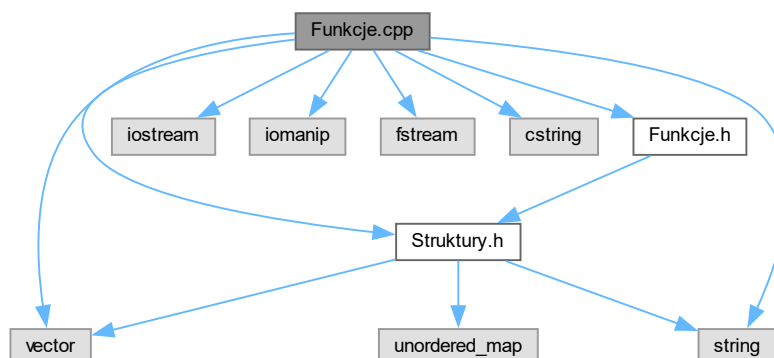
Oto graf wywołań dla tej funkcji:



## 4.2 Dokumentacja pliku Funkcje.cpp

```
#include <vector>
#include <string>
#include <iostream>
#include <iomanip>
#include <fstream>
#include <cstring>
#include "Funkcje.h"
#include "Struktury.h"
```

Wykres zależności załączania dla Funkcje.cpp:



## Funkcje

- `std::string odczytanie_z_pliku` (`const std::string &plik_z_grafem`)  
*Funkcja odczytuje dane z pliku.*
- `int int_ze_stringa` (`std::string &str`)  
*Funkcja przesuwa parametr typu string w celu odnalezienia ciągu znaków które zostaną przekształcone w typ int i usuwa go.*
- `void wypisz_vector` (`std::vector< int > &vector`)  
*Funkcja wypisuje zawartość wektora.*
- `bool czy_wektor_zawiera` (`const std::vector< int > &vector, int wartosc`)  
*Funkcja sprawdza czy podany wektor zawiera cykl do wypisania.*
- `void szukanie_cykli` (`const Graf &graf, std::vector< int > &cykl, const int wierz_poczatkowy, const int sasiad, std::ofstream &zapis`)  
*Funkcja szuka cykli w grafie.*
- `void stworz_graf_ze_stringa` (`std::string str, Graf &graf`)  
*Funkcja do pierwszego znalezionej wierzchołka grafu przypisuje jego sąsiada.*
- `void drukuj_graf` (`Graf &graf`)  
*Wypisuje w konsoli tekstową reprezentację grafu.*
- `void zapisz_cykle` (`Graf &graf, const std::string &plik_z_cyklami`)  
*Funkcja wywołuje funkcję szukanie\_cykli a po znalezieniu cykli zapisuje je do pliku.*

### 4.2.1 Dokumentacja funkcji

#### 4.2.1.1 czy\_wektor\_zawiera()

```
bool czy_wektor_zawiera (  
    const std::vector< int > & vector,  
    int wartosc )
```

Funkcja sprawdza czy podany wektor zawiera cykl do wypisania.

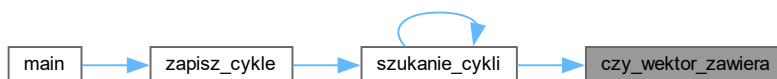
#### Parametry

<i>vector</i>	Przeszukiwany wektor
<i>wartosc</i>	Cykl do wypisania którego szukamy w wektorze

**Zwraca**

Funkcja zwraca true lub false w zależności od tego czy znaleziono cykl do wypisania z wektora

Oto graf wywołań tej funkcji:

**4.2.1.2 drukuj\_graf()**

```
void drukuj_graf (
    Graf & graf )
```

Wypisuje w konsoli tekstową reprezentację grafu.

**Parametry**

out	<i>graf</i>	Graf do wypisania
	<i>graf</i>	Graf który wypisujemy

Oto graf wywołań tej funkcji:

**4.2.1.3 int\_ze\_stringa()**

```
int int_ze_stringa (
    std::string & str )
```

Funkcja przeszkuje parametr typu string w celu odnalezienia ciągu znaków które zostaną przekształcone w typ int i usuwa go.

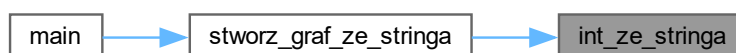
**Parametry**

<i>str</i>	Lancuch znakow w ktorym szukamy ciagow znakow który możemy przemienic na int
------------	--

**Zwraca**

Zwraca wierzcholki grafu zamienione int

Oto graf wywoływań tej funkcji:

**4.2.1.4 odczytanie\_z\_pliku()**

```
std::string odczytanie_z_pliku (  
    const std::string & plik_z_grafem )
```

Funkcja odczytuje dane z pliku.

**Parametry**

<i>plik_z_grafem</i>	Plik zawiera wierzcholki grafu połączone strzałkami
----------------------	---

**Zwraca**

Zwraca odczytane z pliku liczby typu string

Oto graf wywoływań tej funkcji:



#### 4.2.1.5 stworz\_graf\_ze\_stringa()

```
void stworz_graf_ze_stringa (
    std::string str,
    Graf & graf )
```

Funkcja do pierwszego znalezionej wierzchołka grafu przypisuje jego sąsiada.

##### Parametry

out	cykl	Wektor zawierający cykle
	str	Lancuch znakow ktory zostaje wykorzystany w celu wygenerowania grafu
	graf	Graf ktory generujemy

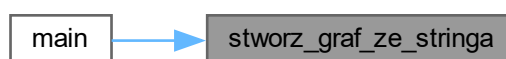
##### Zwraca

Funkcja nic nie zwraca

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



#### 4.2.1.6 szukanie\_cykli()

```
void szukanie_cykli (
    const Graf & graf,
    std::vector< int > & cykl,
    const int wierz_poczatkowy,
    const int sasiad,
    std::ofstream & zapis )
```

Funkcja szuka cykli w grafie.



## Parametry

	<i>graf</i>	Graf który przeszukujemy w celu znalezienia w nim cykli
out	<i>cykl</i>	Znaleziony cykl w grafie
	<i>wierz_poczatkowy</i>	Wierzcholek początkowy krawedzi grafu
	<i>sasiad</i>	Wierzcholek z którym wierzcholek początkowy posiada połączenie w grafie
	<i>zapis</i>	

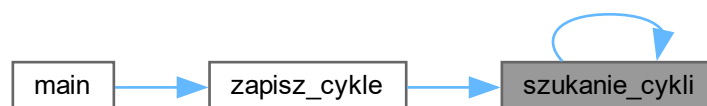
## Zwraca

Funkcja nic nie zwraca

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



## 4.2.1.7 wypisz\_vector()

```
void wypisz_vector (
    std::vector< int > & vector )
```

Funkcja wypisuje zawartosc wektora.

## Parametry

out	<i>vector</i>	Wektor z zapisanymi cyklami
	<i>vector</i>	Wektor który zostanie wypisany

#### 4.2.1.8 zapisz\_cykle()

```
void zapisz_cykle (
    Graf & graf,
    const std::string & plik_z_cyklami )
```

Funkcja wywołuje funkcję szukanie\_cykli a po znalezieniu cykli zapisuje je do pliku.

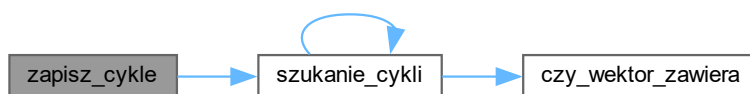
##### Parametry

<i>graf</i>	Graf w którym szukamy cykli
<i>plik_z_cyklami</i>	Plik do którego zostaną zapisane znalezione cykle

##### Zwraca

Funkcja nic nie zwraca

Oto graf wywołań dla tej funkcji:



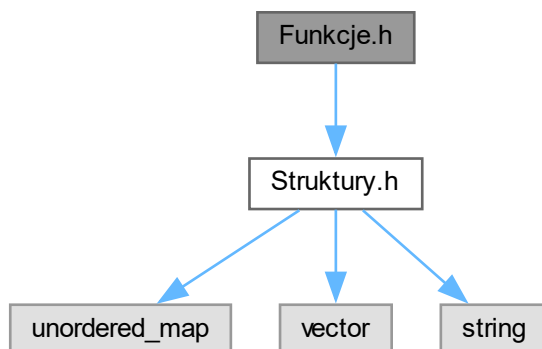
Oto graf wywoływań tej funkcji:



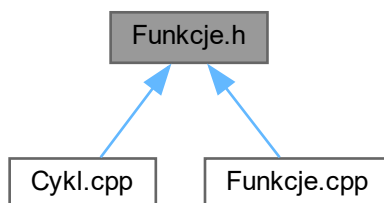
## 4.3 Dokumentacja pliku Funkcje.h

```
#include "Struktury.h"
```

Wykres zależności załączania dla Funkcje.h:



Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:



## Funkcje

- `std::string oczytanie\_z\_pliku (const std::string &plik_z_grafem)`  
*Funkcja odczytuje dane z pliku.*
- `int int\_ze\_stringa (std::string &str)`  
*Funkcja przeszkuje parametr typu string w celu odnalezienia ciągu znaków które zostaną przekształcone w typ int i usuwa go.*
- `void wypisz\_vector (std::vector< int > &vector)`  
*Funkcja wypisuje zawartosc wektora.*
- `bool czy\_wektor\_zawiera (const std::vector< int > &vector, int wartosc)`  
*Funkcja sprawdza czy podany wektor zawiera cykl do wypisania.*
- `void szukanie\_cykli (const Graf &graf, std::vector< int > &cykl, const int wierz_poczatkowy, const int sasiad, std::ofstream &zapis)`  
*Funkcja szuka cykli w grafie.*
- `void stworz\_graf\_ze\_stringa (std::string str, Graf &graf)`  
*Funkcja do pierwszego znalezionej wierzcholka grafu przypisuje jego sasiada.*

- void `drukuj_graf` (`Graf` &graf)  
*Wypisuje w konsoli tekstowa reprezentację grafu.*
- void `zapisz_cykle` (`Graf` &graf, const std::string &plik\_z\_cyklami)  
*Funkcja wywołuje funkcję `szukanie_cykli` a po znalezieniu cykli zapisuje je do pliku.*

### 4.3.1 Dokumentacja funkcji

#### 4.3.1.1 czy\_wektor\_zawiera()

```
bool czy_wektor_zawiera (
    const std::vector< int > & vector,
    int wartosc )
```

Funkcja sprawdza czy podany wektor zawiera cykl do wypisania.

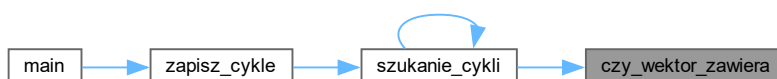
##### Parametry

<code>vector</code>	Przeszukiwany wektor
<code>wartosc</code>	Cykl do wypisania którego szukamy w wektorze

##### Zwraca

Funkcja zwraca true lub false w zależności od tego czy znaleziono cykl do wypisania z wektora

Oto graf wywoływań tej funkcji:



#### 4.3.1.2 drukuj\_graf()

```
void drukuj_graf (
    Graf & graf )
```

Wypisuje w konsoli tekstowa reprezentację grafu.

## Parametry

out	graf	Graf do wypisania
	graf	Graf ktory wypisujemy

Oto graf wywoływań tej funkcji:



## 4.3.1.3 int\_ze\_stringa()

```
int int_ze_stringa (  
    std::string & str )
```

Funkcja przeszkuje parametr typu string w celu odnalezienia ciagu znakow ktore zostane przekształcone w typ int i usuwa go.

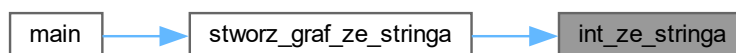
## Parametry

str	Lancuch znakow w ktorym szukamy ciagow znakow ktory mozemy przemienic na int
-----	--

## Zwraca

Zwraca wierzcholki grafu zamienione int

Oto graf wywoływań tej funkcji:



#### 4.3.1.4 odczytanie\_z\_pliku()

```
std::string odczytanie_z_pliku (
    const std::string & plik_z_grafem )
```

Funkcja odczytuje dane z pliku.

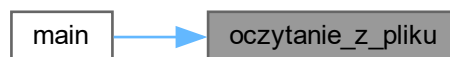
##### Parametry

<i>plik_z_grafem</i>	Plik zawiera wierzchołki grafu połączone strzałkami
----------------------	---

##### Zwraca

Zwraca odczytane z pliku liczby typu string

Oto graf wywoływań tej funkcji:



#### 4.3.1.5 stworz\_graf\_ze\_stringa()

```
void stworz_graf_ze_stringa (
    std::string str,
    Graf & graf )
```

Funkcja do pierwszego znalezionej wierzchołka grafu przypisuje jego sąsiada.

##### Parametry

out	<i>cykl</i>	Wektor zawierający cykle
	<i>str</i>	Lancuch znakow ktory zostaje wykorzystany w celu wygenerowania grafu
	<i>graf</i>	Graf ktory generujemy

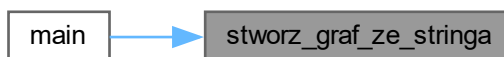
**Zwraca**

Funkcja nic nie zwraca

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:

**4.3.1.6 szukanie\_cykli()**

```
void szukanie_cykli (
    const Graf & graf,
    std::vector< int > & cykl,
    const int wierz_poczatkowy,
    const int sasiad,
    std::ofstream & zapis )
```

Funkcja szuka cykli w grafie.

**Parametry**

	<i>graf</i>	Graf który przeszukujemy w celu znalezienia w nim cykli
out	<i>cykl</i>	Znaleziony cykl w grafie
	<i>wierz_poczatkowy</i>	Wierzcholek początkowy krawedzi grafu
	<i>sasiad</i>	Wierzcholek z którym wierzcholek początkowy posiada połączenie w grafie
	<i>zapis</i>	

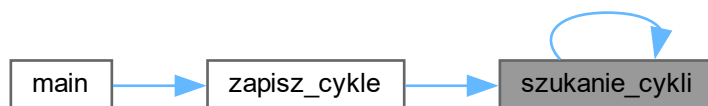
**Zwraca**

Funkcja nic nie zwraca

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:

**4.3.1.7 wypisz\_vector()**

```
void wypisz_vector (
    std::vector< int > & vector )
```

Funkcja wypisuje zawartosc wektora.

**Parametry**

out	<i>vector</i>	Wektor z zapisanymi cyklami
	<i>vector</i>	Wektor ktory zostanie wypisany

**4.3.1.8 zapisz\_cykle()**

```
void zapisz_cykle (
    Graf & graf,
    const std::string & plik_z_cyklami )
```



Funkcja wywołuje funkcje szukanie\_cykli a po znalezieniu cykli zapisuje je do pliku.

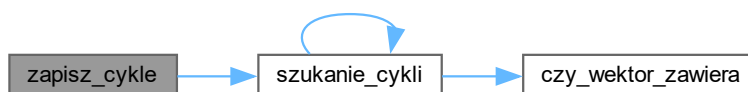
#### Parametry

<i>graf</i>	Graf w którym szukamy cykli
<i>plik_z_cyklami</i>	Plik do ktorego zostana zapisane znalezione cykle

#### Zwraca

Funkcja nic nie zwraca

Oto graf wywołań dla tej funkcji:



Oto graf wywoływań tej funkcji:



## 4.4 Funkcje.h

[Idź do dokumentacji tego pliku.](#)

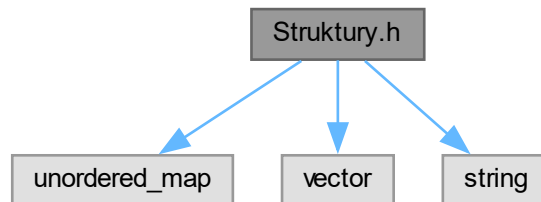
```

00001
00003 // DECLARATIONS (HEADERS) OF FUNCTIONS
00004
00005 #ifndef FUNKCJE_H
00006 #define FUNKCJE_H
00007
00008 #include "Struktury.h"
00009
00014 std::string odczytanie_z_pliku(const std::string& plik_z_grafem);
00015
00020 int int_ze_stringa(std::string& str);
00021
00026 void wypisz_vector(std::vector<int>& vector);
00027
00033 bool czy_wektor_zawiera(const std::vector<int>& vector, int wartosc);
00034
00043 void szukanie_cykli(const Graf& graf, std::vector<int>& cykl, const int wierz_poczatkowy, const int
    sasiad, std::ofstream& zapis);
00044
00051 void stworz_graf_ze_stringa(std::string str, Graf& graf);
00052
00057 void drukuj_graf(Graf& graf);
00058
00064 void zapisz_cykle(Graf& graf, const std::string& plik_z_cyklami);
00065
00066
00067 #endif
  
```

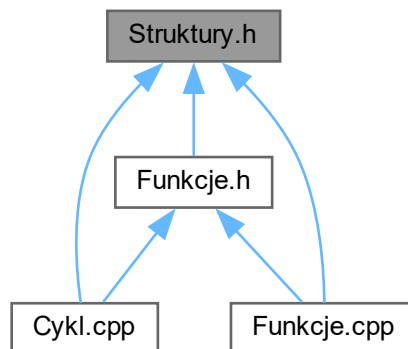
## 4.5 Dokumentacja pliku Struktury.h

```
#include <unordered_map>
#include <vector>
#include <string>
```

Wykres zależności załączania dla Struktury.h:



Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:



### Komponenty

- struct [krawedz](#)  
*Typ przechowujący informacje o krawedzi grafu.*
- struct [wierzcholek](#)  
*Typ przechowujący wczytane krawędzie grafu.*

### Definicje typów

- typedef std::unordered\_map< int, [wierzcholek](#) > [Graf](#)  
*Typedef mapy nieuporządkowanej aby uniknąć zmiennej globalnej.*

## 4.5.1 Dokumentacja definicji typów

### 4.5.1.1 Graf

```
typedef std::unordered_map<int, wierzcholek> Graf
```

Typedef mapy nieuporządkowanej aby uniknąć zmiennej globalnej.

## 4.6 Struktury.h

[Idź do dokumentacji tego pliku.](#)

```
00001
00003 #ifndef STRUKTURY_H
00004 #define STRUKTURY_H
00005
00006 #include <unordered_map>
00007 #include <vector>
00008 #include <string>
00009
00010
00014 struct krawedz
00015 {
00017     int wierz_koncowy;
00018 };
00019
00020
00024 struct wierzcholek
00025 {
00027     std::vector<krawedz> krawedzie;
00028 };
00029
00030
00031
00035 typedef
00036 std::unordered_map<int, wierzcholek> Graf;
00037
00038
00039 #endif
```



# Skorowidz

Cykl.cpp, [7](#)  
    main, [7](#)  
czy\_wektor\_zawiera  
    Funkcje.cpp, [9](#)  
    Funkcje.h, [16](#)  
  
drukuj\_graf  
    Funkcje.cpp, [10](#)  
    Funkcje.h, [16](#)  
  
Funkcje.cpp, [8](#)  
    czy\_wektor\_zawiera, [9](#)  
    drukuj\_graf, [10](#)  
    int\_ze\_stringa, [10](#)  
    oczytanie\_z\_pliku, [11](#)  
    stworz\_graf\_ze\_stringa, [11](#)  
    szukanie\_cykli, [12](#)  
    wypisz\_vector, [13](#)  
    zapisz\_cykle, [14](#)  
Funkcje.h, [14](#)  
    czy\_wektor\_zawiera, [16](#)  
    drukuj\_graf, [16](#)  
    int\_ze\_stringa, [17](#)  
    oczytanie\_z\_pliku, [17](#)  
    stworz\_graf\_ze\_stringa, [18](#)  
    szukanie\_cykli, [19](#)  
    wypisz\_vector, [20](#)  
    zapisz\_cykle, [20](#)  
  
Graf  
    Struktury.h, [23](#)  
  
int\_ze\_stringa  
    Funkcje.cpp, [10](#)  
    Funkcje.h, [17](#)  
  
krawedz, [5](#)  
    wierz\_koncowy, [5](#)  
krawedzie  
    wierzcholek, [6](#)  
  
main  
    Cykl.cpp, [7](#)  
  
oczytanie\_z\_pliku  
    Funkcje.cpp, [11](#)  
    Funkcje.h, [17](#)  
  
Struktury.h, [22](#)  
    Graf, [23](#)  
stworz\_graf\_ze\_stringa  
    Funkcje.cpp, [11](#)  
    Funkcje.h, [18](#)  
    szukanie\_cykli, [12](#)  
    Funkcje.cpp, [12](#)  
    Funkcje.h, [19](#)  
  
wierz\_koncowy  
    krawedz, [5](#)  
wierzcholek, [6](#)  
    krawedzie, [6](#)  
wypisz\_vector  
    Funkcje.cpp, [13](#)  
    Funkcje.h, [20](#)  
  
zapisz\_cykle  
    Funkcje.cpp, [14](#)  
    Funkcje.h, [20](#)