

Cykl
1.02

Wygenerowano przez Doxygen 1.9.8

1 Indeks plików	1
1.1 Lista plików	1
2 Dokumentacja plików	3
2.1 Dokumentacja pliku funkcje.cpp	3
2.1.1 Opis szczegółowy	3
2.1.2 Dokumentacja funkcji	4
2.1.2.1 czy_nowy_cykl()	4
2.1.2.2 czytaj_graf()	4
2.1.2.3 isnumber()	4
2.1.2.4 podszukanie()	5
2.1.2.5 porownaj()	5
2.1.2.6 szukaj_cyklow()	6
2.1.2.7 wypisz_graf()	6
2.1.2.8 wyswietl_cykle()	6
2.1.2.9 zapisz_cykle()	6
2.2 Dokumentacja pliku funkcje.h	7
2.2.1 Opis szczegółowy	8
2.2.2 Dokumentacja funkcji	8
2.2.2.1 czy_nowy_cykl()	8
2.2.2.2 czytaj_graf()	8
2.2.2.3 isnumber()	9
2.2.2.4 podszukanie()	9
2.2.2.5 porownaj()	9
2.2.2.6 szukaj_cyklow()	10
2.2.2.7 wypisz_graf()	10
2.2.2.8 wyswietl_cykle()	10
2.2.2.9 zapisz_cykle()	11
2.3 funkcje.h	11
2.4 Dokumentacja pliku main.cpp	11
2.4.1 Opis szczegółowy	12
Skorowidz	13

Rozdział 1

Indeks plików

1.1 Lista plików

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

funkcje.cpp	Plik zawiera zawartosc funkcji	3
funkcje.h	Plik zawiera deklaracje funkcji	7
main.cpp	Plik zawiera funkcje main	11

Rozdział 2

Dokumentacja plików

2.1 Dokumentacja pliku funkcje.cpp

plik zawiera zawartosc funkcji

```
#include "funkcje.h"
```

Funkcje

- bool [isnumber](#) (std::string &text)
funkcja sprawdza czy podany string mozna przekonwertowac na int
- bool [porownaj](#) (std::vector< int > v1, std::vector< int > v2)
funkcja porownuje dwa wektory (uzywa sortowania)
- [Graph](#) [czytaj_graf](#) (const std::string &nazwa_pliku)
funkcja odczytuje zawartosc podanego pliku i konwertuje go na mape
- void [wypisz_graf](#) (const [Graph](#) &graph)
funkcja wypisuje zawartosc mapy w postaci grafu skierowanego (x -> y)
- bool [czy_nowy_cykl](#) (const [Cycles](#) &cycles, const [Cycle](#) &cycle)
funkcja sprawdza czy podany cykl wystepuje w wektorze cyklow
- [Cycles](#) [szukaj_cyklow](#) (const [Graph](#) &graph)
funkcja zwraca znalezione cykle w podanym grafie skierowanym
- void [podszukanie](#) (const [Graph](#) &graph, const int starting, int node, [Cycles](#) &cycles, [Cycle](#) cycle, std::set< int > visited, int depth)
rekurencyjna funkcja znajdujaca cykle w grafie o podanym poczatku
- void [wyswietl_cykle](#) (const [Cycles](#) &cycles)
funkcja wyswietla podany wektor cyklow
- void [zapisz_cykle](#) (const [Cycles](#) &cycles, const std::string filename)
funkcja zapisuje podany wektor cyklow do pliku

2.1.1 Opis szczegółowy

plik zawiera zawartosc funkcji

Autor

Adam Goczol

Data

13.01.2024

2.1.2 Dokumentacja funkcji

2.1.2.1 czy_nowy_cykl()

```
bool czy_nowy_cykl (
    const Cycles & cycles,
    const Cycle & cycle )
```

funkcja sprawdza czy podany cykl występuje w wektorze cykli

Parametry

<i>cycles</i>	- wektor cykli
<i>cycle</i>	- pojedynczy wektor zawierający cykl

Zwraca

true jeśli nie występuje, false jeśli występuje

2.1.2.2 czytaj_graf()

```
Graph czytaj_graf (
    const std::string & nazwa_pliku )
```

funkcja odczytuje zawartość podanego pliku i konwertuje go na mapę

Parametry

<i>nazwa_pliku</i>	- nazwa pliku z którego chcemy czytać
--------------------	---------------------------------------

Zwraca

mapa zawierająca graf skierowany

2.1.2.3 isnumber()

```
bool isnumber (
    std::string & text )
```

funkcja sprawdza czy podany string można przekonwertować na int

Parametry

<i>text</i>	- string do sprawdzenia
-------------	-------------------------

Autor

Adam Goczol

Zwraca

true jesli sie da, false jesli nie

2.1.2.4 podszukanie()

```
void podszukanie (
    const Graph & graph,
    const int starting,
    int node,
    Cycles & cycles,
    Cycle cycle,
    std::set< int > visited,
    int depth )
```

rekurencyjna funkcja znajdujaca cykle w grafie o podanym poczatku

Parametry

<i>graph</i>	- graf skierowany
<i>starting</i>	- poczatek ktory musi byc koncem cyklu
<i>node</i>	- aktualnie sprawdzana krawedz grafu skierowanego
<i>cycles</i>	- wektor zawierajacy znalezione cykle
<i>cycle</i>	- aktualnie sprawdzany cykl
<i>visited</i>	- odwiedzone krawedzie
<i>depth</i>	- glebokosc rekurencji

2.1.2.5 porownaj()

```
bool porownaj (
    std::vector< int > v1,
    std::vector< int > v2 )
```

funkcja porownuje dwa wektory (uzywa sortowania)

Parametry

<i>v1</i>	- pierwszy wektor
<i>v2</i>	- drugi wektor

Zwraca

true jesli sa identyczne, false jesli nie

2.1.2.6 szukaj_cykwlow()

```
Cycles szukaj_cykwlow (
    const Graph & graph )
```

funkcja zwraca znalezione cykle w podanym grafie skierowanym

Parametry

<i>graph</i>	- graf skierowany
--------------	-------------------

Zwraca

vektor zawierający znalezione cykle

2.1.2.7 wypisz_graf()

```
void wypisz_graf (
    const Graph & graph )
```

funkcja wypisuje zawartość mapy w postaci grafu skierowanego (x -> y)

Parametry

<i>graph</i>	- mapa zawierająca graf skierowany
--------------	------------------------------------

2.1.2.8 wyswietl_cykle()

```
void wyswietl_cykle (
    const Cycles & cycles )
```

funkcja wyświetla podany wektor cykli

Parametry

<i>cycles</i>	- wektor zawierający cykle do wyświetlenia
---------------	--

2.1.2.9 zapisz_cykle()

```
void zapisz_cykle (
    const Cycles & cycles,
    const std::string filename )
```

funkcja zapisuje podany wektor cykli do pliku

Parametry

<code>cycles</code>	- wektor zawierający cykle do zapisania
<code>filename</code>	- nazwa pliku

2.2 Dokumentacja pliku funkcje.h

plik zawiera deklaracje funkcji

```
#include <iostream>
#include <fstream>
#include <set>
#include <utility>
#include <vector>
#include <string>
#include <sstream>
#include <map>
#include <algorithm>
```

Definicje typów

- `typedef std::map< int, std::vector< int > > Graph`
mapa zawierająca wierzchołki grafu skierowanego, np 'a' -> 'b' oznacza że 'a' jest połączone z 'b', a 'b' z 'a' już nie
- `typedef std::vector< int > Cycle`
wektor zawierający pojedynczy cykl z grafu skierowanego np ('a' -> 'b' -> 'c', z czego cykl kończy się na 'c', nie na 'a')
- `typedef std::vector< Cycle > Cycles`
wektor zawierający wszystkie cykle występujące w grafie skierowanym

Funkcje

- `bool isnumber (std::string &text)`
funkcja sprawdza czy podany string można przekonwertować na int
- `bool porownaj (std::vector< int > v1, std::vector< int > v2)`
funkcja porównuje dwa wektory (używa sortowania)
- `Graph czytaj_graf (const std::string &nazwa_pliku)`
funkcja odczytuje zawartość podanego pliku i konwertuje go na mapę
- `bool czy_nowy cykl (const Cycles &cycles, const Cycle &cycle)`
funkcja sprawdza czy podany cykl występuje w wektorze cykli
- `void wypisz_graf (const Graph &graph)`
funkcja wypisuje zawartość mapy w postaci grafu skierowanego (x -> y)
- `Cycles szukaj_cyklow (const Graph &graph)`
funkcja zwraca znalezione cykle w podanym grafie skierowanym
- `void podszukiwanie (const Graph &graph, const int starting, int node, Cycles &cycles, Cycle cycle, std::set< int > visited, int depth)`
rekurencyjna funkcja znajdująca cykle w grafie o podanym początku
- `void wyswietl_cykle (const Cycles &cycles)`
funkcja wyświetla podany wektor cykli
- `void zapisz_cykle (const Cycles &cycles, const std::string filename)`
funkcja zapisuje podany wektor cykli do pliku

2.2.1 Opis szczegółowy

plik zawiera deklaracje funkcji

Autor

Adam Goczol

Data

13.01.2024

2.2.2 Dokumentacja funkcji

2.2.2.1 czy_nowy_cykl()

```
bool czy_nowy_cykl (
    const Cycles & cycles,
    const Cycle & cycle )
```

funkcja sprawdza czy podany cykl występuje w wektorze cyklow

Parametry

<i>cycles</i>	- wektor cyklow
<i>cycle</i>	- pojedynczy wektor zawierający cykl

Zwraca

true jeśli nie występuje, false jeśli występuje

2.2.2.2 czytaj_graf()

```
Graph czytaj_graf (
    const std::string & nazwa_pliku )
```

funkcja odczytuje zawartość podanego pliku i konwertuje go na mapę

Parametry

<i>nazwa_pliku</i>	- nazwa pliku z którego chcemy czytać
--------------------	---------------------------------------

Zwraca

mapa zawierająca graf skierowany

2.2.2.3 isnumber()

```
bool isnumber (
    std::string & text )
```

funkcja sprawdza czy podany string mozna przekonwertowac na int

Parametry

<i>text</i>	- string do sprawdzenia
-------------	-------------------------

Autor

Adam Goczol

Zwraca

true jesli sie da, false jesli nie

2.2.2.4 podszukanie()

```
void podszukanie (
    const Graph & graph,
    const int starting,
    int node,
    Cycles & cycles,
    Cycle cycle,
    std::set< int > visited,
    int depth )
```

rekurencyjna funkcja znajdujaca cykle w grafie o podanym poczatku

Parametry

<i>graph</i>	- graf skierowany
<i>starting</i>	- poczatek ktory musi byc koncem cyklu
<i>node</i>	- aktualnie sprawdzana krawedz grafu skierowanego
<i>cycles</i>	- wektor zawierajacy znalezione cykle
<i>cycle</i>	- aktualnie sprawdzany cykl
<i>visited</i>	- odwiedzone krawedzie
<i>depth</i>	- glebokosc rekurencji

2.2.2.5 porownaj()

```
bool porownaj (
    std::vector< int > v1,
    std::vector< int > v2 )
```

funkcja porownuje dwa wektory (uzywa sortowania)

Parametry

<i>v1</i>	- pierwszy wektor
<i>v2</i>	- drugi wektor

Zwraca

true jesli sa identyczne, false jesli nie

2.2.2.6 szukaj_cykw()

```
Cycles szukaj_cykw (
    const Graph & graph )
```

funkcja zwraca znalezione cykle w podanym grafie skierowanym

Parametry

<i>graph</i>	- graf skierowany
--------------	-------------------

Zwraca

wektor zawierajacy znalezione cykle

2.2.2.7 wypisz_graf()

```
void wypisz_graf (
    const Graph & graph )
```

funkcja wypisuje zawartosc mapy w postaci grafu skierowanego (x -> y)

Parametry

<i>graph</i>	- mapa zawierajaca graf skierowany
--------------	------------------------------------

2.2.2.8 wyswietl_cykle()

```
void wyswietl_cykle (
    const Cycles & cycles )
```

funkcja wyswietla podany wektor cykw

Parametry

<i>cycles</i>	- wektor zawierajacy cykle do wyswietlenia
---------------	--

2.2.2.9 zapisz_cykle()

```
void zapisz_cykle (
    const Cycles & cycles,
    const std::string filename )
```

funkcja zapisuje podany wektor cykli do pliku

Parametry

<i><code>cycles</code></i>	- wektor zawierający cykle do zapisania
<i><code>filename</code></i>	- nazwa pliku

2.3 funkcje.h

[Idź do dokumentacji tego pliku.](#)

```
00001
00007 #ifndef FUNKCJE_H
00008 #define FUNKCJE_H
00009
00010 #include <iostream>
00011 #include <fstream>
00012 #include <set>
00013 #include <utility>
00014 #include <vector>
00015 #include <string>
00016 #include <sstream>
00017 #include <map>
00018 #include <algorithm>
00019 #include <fstream>
00020
00025 typedef std::map<int, std::vector<int> > Graph;
00026
00031 typedef std::vector<int> Cycle;
00032
00037 typedef std::vector<Cycle> Cycles;
00038
00046 bool isnumber(std::string& text);
00047
00048
00056 bool porownaj(std::vector<int> v1, std::vector<int> v2);
00057
00064 Graph czytaj_graf(const std::string& nazwa_pliku);
00065
00073 bool czy_nowy_cykl(const Cycles& cycles, const Cycle& cycle);
00074
00075
00081 void wypisz_graf(const Graph& graph);
00082
00089 Cycles szukaj_cyklow(const Graph& graph);
00090
00102 void podszukanie(const Graph& graph, const int starting, int node,
00103     Cycles& cycles, Cycle cycle, std::set<int> visited, int depth);
00104
00110 void wyswietl_cykle(const Cycles& cycles);
00111
00118 void zapisz_cykle(const Cycles& cycles, const std::string filename);
00119
00120
00121
00122 #endif // !FUNKCJE_H
00123
```

2.4 Dokumentacja pliku main.cpp

plik zawiera funkcje main

```
#include "funkcje.h"
```

Funkcje

- int **main** (int n, char *params[])

2.4.1 Opis szczegółowy

plik zawiera funkcje main

Autor

Adam Goczol

Data

13.01.2024

Skorowidz

czy_nowy_cykl
funkcje.cpp, 4
funkcje.h, 8

czytaj_graf
funkcje.cpp, 4
funkcje.h, 8

funkcje.cpp, 3
czy_nowy_cykl, 4
czytaj_graf, 4
isnumber, 4
podszukanie, 5
porownaj, 5
szukaj_cyklow, 5
wypisz_graf, 6
wyswietl_cykle, 6
zapisz_cykle, 6

funkcje.h, 7
czy_nowy_cykl, 8
czytaj_graf, 8
isnumber, 8
podszukanie, 9
porownaj, 9
szukaj_cyklow, 10
wypisz_graf, 10
wyswietl_cykle, 10
zapisz_cykle, 11

isnumber
funkcje.cpp, 4
funkcje.h, 8

main.cpp, 11

podszukanie
funkcje.cpp, 5
funkcje.h, 9

porownaj
funkcje.cpp, 5
funkcje.h, 9

szukaj_cyklow
funkcje.cpp, 5
funkcje.h, 10

wypisz_graf
funkcje.cpp, 6
funkcje.h, 10

wyswietl_cykle
funkcje.cpp, 6
funkcje.h, 10

zapisz_cykle
funkcje.cpp, 6
funkcje.h, 11