

Problem niezawodności sieci

1.0

Wygenerowano przez Doxygen 1.9.8

1 Indeks plików	1
1.1 Lista plików	1
2 Dokumentacja plików	3
2.1 Dokumentacja pliku files.h	3
2.1.1 Opis szczegółowy	4
2.1.2 Dokumentacja funkcji	4
2.1.2.1 Bfs()	4
2.1.2.2 create_graph()	4
2.1.2.3 readfile()	5
2.1.2.4 reverse_kruskal()	5
2.1.2.5 writefile()	5
2.2 files.h	6
2.3 Dokumentacja pliku messages.h	6
2.3.1 Opis szczegółowy	7
2.3.2 Dokumentacja funkcji	7
2.3.2.1 show_data()	7
2.3.2.2 show_graph()	8
2.4 messages.h	8
Skorowidz	9

Rozdział 1

Indeks plików

1.1 Lista plików

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

files.h	3
messages.h	6

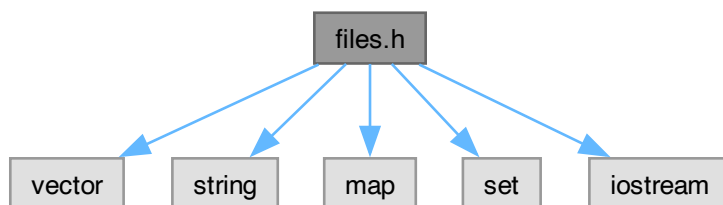
Rozdział 2

Dokumentacja plików

2.1 Dokumentacja pliku files.h

```
#include <vector>
#include <string>
#include <map>
#include <set>
#include <iostream>
```

Wykres zależności załączania dla files.h:



Definicje typów

- `typedef std::map< std::string, std::set< std::string > > Graph`
mapa, w której klucze są napisami reprezentującymi dany wierzchołek, wartości to sety napisów, reprezentujące wierzchołki z którymi połączony jest klucz
- `typedef std::vector< std::pair< std::pair< std::string, std::string >, double > > Data`
wektor par, para składa się z kolejnej pary, reprezentującej krawędź, druga wartość zewnętrznej pary to waga krawędzi

Funkcje

- **Data readfile** (const std::string &path)
wczytuje z pliku dane i zapisuje do struktury Data
- **Graph create_graph** (const Data &data)
tworzy reprezentacje grafu, jako mapy
- std::set< std::string > **Bfs** (const Graph &graph, const std::string &node)
algorytm przeszukiwania wszerek
- **Graph reverse_kruskal** (const Data &data, const int min_connections, const Graph &graph)
zmodyfikowany reverse-delete algorytm
- void **writefile** (const std::string &path, Graph &graph)
Zapis wyniku do pliku wyjściowego.

2.1.1 Opis szczegółowy

Autor

Kamil Kasperek

Data

18.01.2023r.

2.1.2 Dokumentacja funkcji

2.1.2.1 Bfs()

```
std::set< std::string > Bfs (  
    const Graph & graph,  
    const std::string & node )
```

algorytm przeszukiwania wszerek

Parametry

<i>graph</i>	graf w strukturze Graph
<i>node</i>	wierzchołek startowy

Zwraca

set wierzchołkow odwiedzonych

2.1.2.2 create_graph()

```
Graph create_graph (  
    const Data & data )
```

tworzy reprezentacje grafu, jako mapy

Parametry

<i>data</i>	graf w strukturze Data
-------------	------------------------

Zwraca

strukture Graph

2.1.2.3 readfile()

```
Data readfile (
    const std::string & path )
```

wczytuje z pliku dane i zapisuje do struktury Data

Parametry

<i>path</i>	ścieżka do pliku
-------------	------------------

Zwraca

strukture Data

2.1.2.4 reverse_kruskal()

```
Graph reverse_kruskal (
    const Data & data,
    const int min_connections,
    const Graph & graph )
```

zmodyfikowany reverse-delete algorytm

Parametry

<i>data</i>	Struktura Data, reprezentująca krawędzie grafu
<i>min_connections</i>	Żądany poziom niezawodności (unsigned int)
<i>graph</i>	Struktura Graph, reprezentująca wierzchołki grafu

Zwraca

Struktura Graph, reprezentującą wynik

2.1.2.5 writefile()

```
void writefile (
    const std::string & path,
    Graph & graph )
```

Zapis wyniku do pliku wyjściowego.

Parametry

<i>path</i>	ścieżka do pliku wyjściowego
<i>graph</i>	Struktura Graph, reprezentująca wynik

2.2 files.h

[Idź do dokumentacji tego pliku.](#)

```

00001
00006 #include <vector>
00007 #include <string>
00008 #include <map>
00009 #include <set>
00010 #include <iostream>
00011
00012 #ifndef FILES_H
00013 #define FILES_H
00014
00019 typedef std::map<std::string, std::set<std::string> > Graph ;
00020
00025 typedef std::vector<std::pair<std::pair<std::string, std::string>, double > > Data ;
00026
00031 Data readfile(const std::string & path) ;
00032
00037 Graph create_graph(const Data & data) ;
00038
00044 std::set<std::string> Bfs(const Graph& graph, const std::string& node) ;
00045
00052 Graph reverse_kruskal(const Data & data, const int min_connections, const Graph & graph) ;
00053
00059 void writefile(const std::string & path, Graph & graph) ;
00060
00061 #endif

```

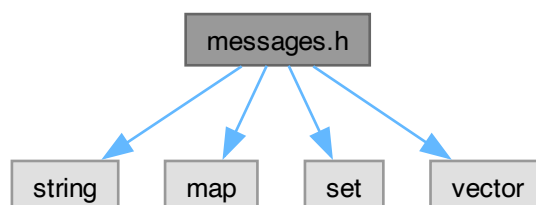
2.3 Dokumentacja pliku messages.h

```

#include <string>
#include <map>
#include <set>
#include <vector>

```

Wykres zależności załączania dla messages.h:



Definicje typów

- typedef std::map< std::string, std::set< std::string > > **Graph**
- typedef std::vector< std::pair< std::pair< std::string, std::string >, double > > **Data**

Funkcje

- void **help** ()
Wypisuje na standardowe wyjście użycie przełączników.
- void **no_arguments** ()
Wypisuje na standardowe wyjście błąd o nie użyciu przełączników.
- void **fileopen_error** ()
Wypisuje na standardowe wyjście informacje o błędzie przy otwarciu pliku.
- void **no_input** ()
Wypisuje na standardowe wyjście informacje o braku przełącznika -i.
- void **no_output** ()
Wypisuje na standardowe wyjście informacje o braku przełącznika -o.
- void **no_reliability** ()
Wypisuje na standardowe wyjście informacje o braku przełącznika -n.
- void **negative_reliability** ()
Wypisuje na standardowe wyjście informacje o minusowej niezawodności.
- void **show_graph** (const **Graph** &graph)
Wypisuje na standardowe wyjście zawartość mapy reprezentującą graf.
- void **show_data** (const **Data** &data)
Wypisuje na standardowe wyjście dane w postaci wektora par.

2.3.1 Opis szczegółowy

Autor

Kamil Kasperek

Data

18.01.2023r.

2.3.2 Dokumentacja funkcji

2.3.2.1 show_data()

```
void show_data (  
    const Data & data )
```

Wypisuje na standardowe wyjście dane w postaci wektora par.

Parametry

<i>data</i>	dane zapisane za pomocą struktury Data
-------------	---

2.3.2.2 show_graph()

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

Wypisuje na standardowe wyjście zawartość mapy reprezentującą graf.

Parametry

<i>graph</i>	graf do wyświetlenia w strukturze Graph
--------------	---

2.4 messages.h

[Idź do dokumentacji tego pliku.](#)

```
00001
00007 #include <string>
00008 #include <map>
00009 #include <set>
00010 #include <vector>
00011
00012 #ifndef MESSAGES_H
00013 #define MESSAGES_H
00014
00015 typedef std::map<std::string, std::set<std::string> > Graph ;
00016 typedef std::vector<std::pair<std::pair<std::string, std::string>, double > > Data ;
00017
00019 void help() ;
00020
00022 void no_arguments() ;
00023
00025 void fileopen_error() ;
00026
00028 void no_input() ;
00029
00031 void no_output() ;
00032
00034 void no_reliability() ;
00035
00037 void negative_reliability() ;
00038
00041 void show_graph(const Graph & graph) ;
00042
00046 void show_data(const Data & data) ;
00047
00048 #endif
```

Skorowidz

Bfs

files.h, [4](#)

create_graph

files.h, [4](#)

files.h, [3](#)

Bfs, [4](#)

create_graph, [4](#)

readfile, [5](#)

reverse_kruskal, [5](#)

writefile, [5](#)

messages.h, [6](#)

show_data, [7](#)

show_graph, [8](#)

readfile

files.h, [5](#)

reverse_kruskal

files.h, [5](#)

show_data

messages.h, [7](#)

show_graph

messages.h, [8](#)

writefile

files.h, [5](#)