Listonosz

Generated by Doxygen 1.10.0

1 File Index 1

1.1 File List . . . 1

2 File Documentation 3

2.1 Listonosz.cpp File Reference . . . 3

2.1.1 Typedef Documentation . . . 4

2.1.1.1 Graph . . . 4

2.1.2 Function Documentation . . . 4

2.1.2.1 addEdge() . . . 4

2.1.2.2 connectOddDegreeVertices() . . . 4

2.1.2.3 dfs() . . . 4

2.1.2.4 dijkstra() . . . 5

2.1.2.5 findEulerianCycle() . . . 5

2.1.2.6 isConnected() . . . 5

2.1.2.7 LoadFromFile() . . . 6

2.1.2.8 main() . . . 6

2.1.2.9 removeEdge() . . . 6

2.1.2.10 writeResultToFile() . . . 7

Index 9

Chapter 1

File Index

1.1 File List

Here is a list of all files with brief descriptions:

Listonosz.cpp 3

Chapter 2

File Documentation

2.1 Listonosz.cpp File Reference

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

Typedefs

- typedef map< int, multimap< int, pair< double, string > > > Graph

Functions

- Graph LoadFromFile (const string &fileName)

  Funkcja odczytująca układ ulic z pliku.
- void addEdge (Graph &graph, int cross1, int cross2, double weight, string streetName)

  Funkcja dodajaca ścieżke do grafu.
- void removeEdge (Graph &graph, int cross1, int cross2)

  Funkcja usuwająca ścieżke z grafu.
- void dfs (Graph &graph, int v, map< int, bool > &visited)

  Funkcja przeszukująca graf w głąb.
- bool isConnected (Graph &graph)

  Funkcja sprawdzająca czy graf jest spójny.
- map< int, double > dijkstra (const Graph &graph, int start)

  Funkcja znajduje najkrótsze ścieżeki w grafie od danego punktu.
- Graph connectOddDegreeVertices (const Graph &originalGraph)

  Funkcja łącząca nieprzyste wierzchołki w grafie.
- list< int > findEulerianCycle (Graph &graph, int startVertex)

  Funkcja znajduje cykl Eulera w grafie - cykl który przechodzi przez każdą krawędź dokładnie raz.
- void writeResultToFile (Graph graph, list< int > finalList, int startCross, string &fileName)

  Funkcja wypisująca wynikowy graf do pliku.
- int main (int n, char **carg)

  Funkcja główna programu.

2.1.1 Typedef Documentation

2.1.1.1 Graph

```
typedef map<int, multimap<int, pair<double, string> > > Graph
```

2.1.2 Function Documentation

2.1.2.1 addEdge()

```
void addEdge (
            Graph & graph,
            int cross1,
            int cross2,
            double weight,
            string streetName )
```

Funkcja dodajaca ścieżke do grafu.

Parameters

graph	Graf układ ulic
cross1	Pierwszy wierzchołek
cross2	Drugi wierzchołek
weight	Długość ulicy
streetName	Nazwa ulicy

2.1.2.2 connectOddDegreeVertices()

```
Graph connectOddDegreeVertices (
            const Graph & originalGraph )
```

Funkcja łącząca nieprzyste wierzchołki w grafie.

Parameters

originalGraph	Graf z układem ulic

Returns

Zwraca graf wejściowy ale z połączonymi wierzchołkami o nieparzystych stopniach

2.1.2.3 dfs()

```
void dfs (
            Graph & graph,
```

```
            int v,
            map< int, bool > & visited )
```

Funkcja przeszukująca graf w głąb.

Parameters

graf	Graf układ ulic
v	Wierzchołek startowy
visited	lista odwiedzanych wierzchołków

2.1.2.4 dijkstra()

```
map< int, double > dijkstra (
            const Graph & graph,
            int start )
```

Funkcja znajduje najkrótsze ścieżeki w grafie od danego punktu.

Parameters

graph	Graf z układem ulic
start	Startowy wierzchołek

Returns

Zwraca mape z najkrótszymi ścieżkami

2.1.2.5 findEulerianCycle()

```
list< int > findEulerianCycle (
            Graph & graph,
            int startVertex )
```

Funkcja znajduje cykl Eulera w grafie - cykl który przechodzi przez każdą krawędź dokładnie raz.

Parameters

graph	Graf z układem ulic
startVertex	Startowy wierzchołek

Returns

Zwraca listę z cyklem Eulera

2.1.2.6 isConnected()

```
bool isConnected (
            Graph & graph )
```

Funkcja sprawdzająca czy graf jest spójny.

Parameters

graph	Graf układ ulic

Returns

Zwraca wartość boolowską czy graf jest połączony czy nie

2.1.2.7 LoadFromFile()

```
Graph LoadFromFile (
            const string & fileName )
```

Funkcja odczytująca układ ulic z pliku.

Parameters

fileName	Nazwa pliku

Returns

Graf z układem ulic

2.1.2.8 main()

```
int main (
            int n,
            char ** carg )
```

Funkcja główna programu.

Parameters

n	Ilość argumentów wiersza poleceń
carg	Tablica argumentów wiersza poleceń

Returns

Kod zakończenia programu

2.1.2.9 removeEdge()

```
void removeEdge (
            Graph & graph,
```

```
            int cross1,
            int cross2 )
```

Funkcja usuwająca ścieżke z grafu.

Parameters

graph	Graf układ ulic
cross1	Pierwszy wierzchołek
cross2	Drugi wierzchołek

2.1.2.10 writeResultToFile()

```
void writeResultToFile (
            Graph graph,
            list< int > finalList,
            int startCross,
            string & fileName )
```

Funkcja wypisująca wynikowy graf do pliku.

Parameters

graph	Graf z układem ulic
finalList	Lista z najkrótszą drogą
startCross	Startowe skrzyżowanie
fileName	Nazwa pliku do którego wpisywany jest wynik

Index

addEdge
    Listonosz.cpp, 4

connectOddDegreeVertices
    Listonosz.cpp, 4

dfs
    Listonosz.cpp, 4
dijkstra
    Listonosz.cpp, 5

findEulerianCycle
    Listonosz.cpp, 5

Graph
    Listonosz.cpp, 4

isConnected
    Listonosz.cpp, 5

Listonosz.cpp, 3
    addEdge, 4
    connectOddDegreeVertices, 4
    dfs, 4
    dijkstra, 5
    findEulerianCycle, 5
    Graph, 4
    isConnected, 5
    LoadFromFile, 6
    main, 6
    removeEdge, 6
    writeResultToFile, 7
LoadFromFile
    Listonosz.cpp, 6

main
    Listonosz.cpp, 6

removeEdge
    Listonosz.cpp, 6

writeResultToFile
    Listonosz.cpp, 7