

Dijkstra

Generated by Doxygen 1.8.20

1 Class Index	1
1.1 Class List	1
2 File Index	3
2.1 File List	3
3 Class Documentation	5
3.1 NextPt Struct Reference	5
3.1.1 Detailed Description	5
3.2 PreviousPt Struct Reference	5
3.2.1 Detailed Description	6
4 File Documentation	7
4.1 Dijkstra/CommandLineManagment.h File Reference	7
4.1.1 Function Documentation	7
4.1.1.1 showHelp()	7
4.2 Dijkstra/Dijkstra.h File Reference	7
4.2.1 Function Documentation	8
4.2.1.1 dijkstra()	8
4.3 Dijkstra/FileManagment.h File Reference	8
4.3.1 Function Documentation	9
4.3.1.1 findPrevious()	9
4.3.1.2 inputGraf()	9
4.3.1.3 inputStartPkt()	9
4.3.1.4 outputDijkstra()	10
4.4 Dijkstra/OperationOfDijkstra.h File Reference	10
4.4.1 Function Documentation	11
4.4.1.1 findMin()	11
4.4.1.2 mapToMatrix()	11
4.4.1.3 operationOfDijkstra()	11
Index	13

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

NextPt	5
PreviousPt	5

Chapter 2

File Index

2.1 File List

Here is a list of all documented files with brief descriptions:

Dijkstra/ CommandLineManagament.h	7
Dijkstra/ Dijkstra.h	7
Dijkstra/ FileManagment.h	8
Dijkstra/ NextPt.h	??
Dijkstra/ OperationOfDijkstra.h	10
Dijkstra/ PreviousPt.h	??

Chapter 3

Class Documentation

3.1 NextPt Struct Reference

```
#include <NextPt.h>
```

Public Attributes

- int **nextPt**
- double **distance**

3.1.1 Detailed Description

Struktura odpowiedzialna za przechowanie nastepnikow danego punktu oraz ich krawedzi

Parameters

<i>nextPt</i>	Punkt nastepny dla danego punktu poczatkowego
<i>distance</i>	waga krawedzi w grafie od danego punktu do jego nastepnika

The documentation for this struct was generated from the following file:

- Dijkstra/NextPt.h

3.2 PreviousPt Struct Reference

```
#include <PreviousPt.h>
```

Public Attributes

- int **previous**
- double **distance**
- bool **wasUsed**

3.2.1 Detailed Description

Struktura odpowiedzialna za przechowanie danych poprzedników wraz z informacją czy zosta³ już wykorzystany w algorytmie.

Parameters

<i>previous</i>	Poprzedni punkt dla danego punktu
<i>distance</i>	Odległość drogi jaką przebywa się do danego punktu
<i>wasUsed</i>	Bool którego wartość true powiadamia, iż punkt został już użyty w algorytmie

The documentation for this struct was generated from the following file:

- Dijkstra/PreviousPt.h

Chapter 4

File Documentation

4.1 Dijkstra/CommandLineManagment.h File Reference

```
#include <string>
#include <iostream>
```

Functions

- void [showHelp](#) ()

4.1.1 Function Documentation

4.1.1.1 showHelp()

```
void showHelp ( )
```

Funkcja wywietajaca pomoc uzyta aby uzytkownik po wpisaniu -h otrzymal liste argumentow programu

4.2 Dijkstra/Dijkstra.h File Reference

```
#include <iostream>
#include <vector>
#include <map>
#include "NextPt.h"
#include "PreviousPt.h"
#include "OperationOfDijkstra.h"
```

Functions

- `std::map< int, PreviousPt > dijkstra (std::map< int, std::vector< NextPt >> neighbors, int start, std::map< int, std::map< int, double >> matrix)`

4.2.1 Function Documentation

4.2.1.1 dijkstra()

```
std::map<int, PreviousPt> dijkstra (
    std::map< int, std::vector< NextPt >> neighbors,
    int start,
    std::map< int, std::map< int, double >> matrix )
```

Funkcja odpowiedzialna za algorytm dijkstry

Parameters

<i>neighbors</i>	Mapa przechowująca wektor struktur, czyli następników danego punktu, który jest id danej mapy.
<i>start</i>	Punkt od którego ma zostać liczona najkrótsza droga do innych punktów grafu.
<i>matrix</i>	Mapa map imitująca macierz, po której wyszukiwana zostaje najkrótsza droga do punktów

Returns

`map<int, PreciousPt>` Mapa w której znajduje się struktura punktów z najkrótszą drogą oraz poprzednikiem danego punktu

4.3 Dijkstra/FileManagment.h File Reference

```
#include <iostream>
#include <fstream>
#include <string>
#include <vector>
#include <map>
#include "NextPt.h"
#include "PreviousPt.h"
#include "OperationOfDijkstra.h"
```

Functions

- `std::map< int, std::vector< NextPt >> inputGraf (std::string fileWithGraph)`
- `std::vector< int > inputStartPkt (std::string fileWithStarts, std::map< int, std::vector< NextPt >> neighbors)`
- `std::vector< int > findPrevious (int current, std::map< int, PreviousPt > previousOne, int start)`
- `void outputDijkstra (std::map< int, std::vector< NextPt >> neighbors, int startingPt, std::map< int, PreviousPt > previousOne, std::string outputFile)`

4.3.1 Function Documentation

4.3.1.1 findPrevious()

```
std::vector<int> findPrevious (
    int current,
    std::map< int, PreviousPt > previousOne,
    int start )
```

Funkcja szukająca poprzedników dla danego punktu i tworząca z nich wektor aby można było odczytać poprzedników (funkcja używana w wypisywaniu danych do pliku).

Parameters

<i>current</i>	aktualnie rozpatrywany punkt podawany jako wskaźnik na id mapy previousOne
<i>previousOne</i>	Mapa przechowująca poprzedników, odległość oraz bool czy dany punkt został użyty dla danego punktu.
<i>start</i>	Punkt startowy, który jest aktualnie rozpatrywany, podawany jako wartość danego wektora.

Returns

std::vector<int> wektor posiadający poprzedników danego punktu

4.3.1.2 inputGraf()

```
std::map<int, std::vector<NextPt> > inputGraf (
    std::string fileWithGraph )
```

Funkcja wczytująca z pliku graf Dijkstry.

Parameters

<i>fileWithGraph</i>	Nazwa pliku, z którego zostaje pobrany graf
----------------------	---

Returns

std::map<int, std::vector<NextPt>> Mapa punktów wraz z ich wektorem następników i odległościami dla danego następnika, przechowywane w strukturze [NextPt](#)

4.3.1.3 inputStartPkt()

```
std::vector<int> inputStartPkt (
    std::string fileWithStarts,
    std::map< int, std::vector< NextPt >> neighbors )
```

Funkcja wczytująca z pliku liste punktów startowych oraz tworząca dla ich w mapie neighbors punkt z brakiem następnika i zerową odległością.

Parameters

<i>fileWithStarts</i>	Nazwa pliku, z którego zostaje pobrana lista (wektor) punktów
<i>neighbors</i>	Mapa przechowująca wektor struktów, czyli następników danego punktu, zaznaczonego jako id danej mapy (wpisywany do niej zostaje punkt startowy jako punkt z zerową odległością).

Returns

`std::vector<int>` wektor punktów startowych

4.3.1.4 outputDijkstra()

```
void outputDijkstra (
    std::map< int, std::vector< NextPt >> neighbors,
    int startingPt,
    std::map< int, PreviousPt > previousOne,
    std::string outputFile )
```

Funkcja wypisująca dane do pliku wyjściowego.

Parameters

<i>neighbors</i>	Mapa przechowująca wektor struktów, czyli następników danego punktu, zaznaczonego jako id danej mapy
<i>startingPt</i>	Punkt startowy, który jest aktualnie rozpatrywany
<i>previousOne</i>	Mapa przechowująca poprzedników, odległość oraz bool czy dany punkt został użyty dla danego punktu
<i>outputFile</i>	Nazwa pliku wyjściowego

4.4 Dijkstra/OperationOfDijkstra.h File Reference

```
#include <map>
#include <vector>
#include <stdio.h>
#include "NextPt.h"
#include "PreviousPt.h"
#include "Dijkstra.h"
#include "FileManagment.h"
```

Functions

- `std::map< int, std::map< int, double > >` [mapToMatrix](#) (`std::map< int, std::vector< NextPt >> neighbors`)
- `int` [findMin](#) (`std::map< int, PreviousPt > previousOne`)
- `void` [operationOfDijkstra](#) (`std::string fileWithGraph, std::string fileWithStarts, std::string outputFile`)

4.4.1 Function Documentation

4.4.1.1 findMin()

```
int findMin (
    std::map< int, PreviousPt > previousOne )
```

Funkcja szukająca minimum w mapie previousOne oraz zwracająca indeks tej mapy.

Parameters

<i>previousOne</i>	Mapa przechowująca poprzedników, odległość oraz bool czy dany punkt został użyty dla danego punktu.
--------------------	---

Returns

int Zwracany jest indeks na mapie z najkrótszą trasą

4.4.1.2 mapToMatrix()

```
std::map<int, std::map<int, double> > mapToMatrix (
    std::map< int, std::vector< NextPt >> neighbors )
```

Funkcja konwertująca mapę neighbors na macierz, potrzebna w algorytmie dijkstra.

Parameters

<i>neighbors</i>	Mapa przechowująca wektor struktur, czyli następników danego punktu, który jest id danej mapy.
------------------	--

Returns

map<int, std::map<int,double>> Zwracana zostaje mapa map imitująca macierz

4.4.1.3 operationOfDijkstra()

```
void operationOfDijkstra (
    std::string fileWithGraph,
    std::string fileWithStarts,
    std::string outputFile )
```

Funkcja w której inicjalizowana zostają zbiory, funkcje i zmienne, oraz wywołana zostaje funkcja algorytmu dijkstry i wypisywania do pliku dla każdego punktu startowego.

Parameters

<i>fileWithGraph</i>	Nazwa pliku, w którym znajduje się graf
<i>fileWithStarts</i>	Nazwa pliku, w którym znajduje się lista punktów startowych
<i>outputFile</i>	Nazwa pliku wyjściowego, do którego będą wpisywane dane

Index

CommandLineManagment.h
 showHelp, [7](#)

dijkstra
 Dijkstra.h, [8](#)
Dijkstra.h
 dijkstra, [8](#)
Dijkstra/CommandLineManagment.h, [7](#)
Dijkstra/Dijkstra.h, [7](#)
Dijkstra/FileManagment.h, [8](#)
Dijkstra/OperationOfDijkstra.h, [10](#)

FileManagment.h
 findPrevious, [9](#)
 inputGraf, [9](#)
 inputStartPkt, [9](#)
 outputDijkstra, [10](#)

findMin
 OperationOfDijkstra.h, [11](#)
findPrevious
 FileManagment.h, [9](#)

inputGraf
 FileManagment.h, [9](#)
inputStartPkt
 FileManagment.h, [9](#)

mapToMatrix
 OperationOfDijkstra.h, [11](#)

NextPt, [5](#)

operationOfDijkstra
 OperationOfDijkstra.h, [11](#)
OperationOfDijkstra.h
 findMin, [11](#)
 mapToMatrix, [11](#)
 operationOfDijkstra, [11](#)
outputDijkstra
 FileManagment.h, [10](#)

PreviousPt, [5](#)

showHelp
 CommandLineManagment.h, [7](#)