

Hamilton

Generated by Doxygen 1.10.0

1 Class Index	1
1.1 Class List	1
2 File Index	3
2.1 File List	3
3 Class Documentation	5
3.1 cmd_args Struct Reference	5
3.2 config_struct Struct Reference	5
3.3 edge Struct Reference	5
3.4 graph_struct Struct Reference	6
4 File Documentation	7
4.1 C:/Users/Jarek/source/repos/f5abc478-gr22-repo/projekt/hamilton_projekt/hamilton↵ _algorithm.h File Reference	7
4.1.1 Function Documentation	7
4.1.1.1 DFS()	7
4.1.1.2 find_the_path()	8
4.2 hamilton_algorithm.h	8
4.3 C:/Users/Jarek/source/repos/f5abc478-gr22-repo/projekt/hamilton_projekt/hamilton↵ _algotihm.cpp File Reference	8
4.3.1 Function Documentation	9
4.3.1.1 DFS()	9
4.3.1.2 find_the_path()	9
4.4 C:/Users/Jarek/source/repos/f5abc478-gr22-repo/projekt/hamilton_projekt/hamilton↵ _main.cpp File Reference	9
4.5 C:/Users/Jarek/source/repos/f5abc478-gr22-repo/projekt/hamilton_projekt/hamilton↵ output.cpp File Reference	10
4.5.1 Function Documentation	10
4.5.1.1 is_number()	10
4.5.1.2 output_the_path()	11
4.5.1.3 read_command_line_arguments()	11
4.5.1.4 read_graph()	11
4.6 C:/Users/Jarek/source/repos/f5abc478-gr22-repo/projekt/hamilton_projekt/hamilton↵ input_output.h File Reference	12
4.6.1 Function Documentation	12
4.6.1.1 is_number()	12
4.6.1.2 output_the_path()	12
4.6.1.3 read_command_line_arguments()	13
4.6.1.4 read_graph()	13
4.7 input_output.h	13
Index	15

Chapter 1

Class Index

1.1 Class List

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

cmd_args	5
config_struct	5
edge	5
graph_struct	6

Chapter 2

File Index

2.1 File List

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

C:/Users/Jarek/source/repos/f5abc478-gr22-repo/projekt/hamilton_projekt/hamilton_projekt/[hamilton_algorithm.h](#)
7

C:/Users/Jarek/source/repos/f5abc478-gr22-repo/projekt/hamilton_projekt/hamilton_projekt/[hamilton_algotihm.cpp](#)
8

C:/Users/Jarek/source/repos/f5abc478-gr22-repo/projekt/hamilton_projekt/hamilton_projekt/[hamilton_main.cpp](#)
9

C:/Users/Jarek/source/repos/f5abc478-gr22-repo/projekt/hamilton_projekt/hamilton_projekt/[input_output.cpp](#)
10

C:/Users/Jarek/source/repos/f5abc478-gr22-repo/projekt/hamilton_projekt/hamilton_projekt/[input_output.h](#)
12

Chapter 3

Class Documentation

3.1 cmd_args Struct Reference

Public Attributes

- int **argc**
- char ** **argv**

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

- C:/Users/Jarek/source/repos/f5abc478-gr22-repo/projekt/hamilton_projekt/hamilton_projekt/[input_output.h](#)

3.2 config_struct Struct Reference

Public Attributes

- int **vertex**
- std::string **input**
- std::string **output**

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

- C:/Users/Jarek/source/repos/f5abc478-gr22-repo/projekt/hamilton_projekt/hamilton_projekt/[input_output.h](#)

3.3 edge Struct Reference

Public Attributes

- int **vertex**
- int **weight**

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

- C:/Users/Jarek/source/repos/f5abc478-gr22-repo/projekt/hamilton_projekt/hamilton_projekt/[input_output.h](#)

3.4 graph_struct Struct Reference

Public Attributes

- `std::vector< std::list< edge > > graph`

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

- `C:/Users/Jarek/source/repos/f5abc478-gr22-repo/projekt/hamilton_projekt/hamilton_projekt/input_output.h`

Chapter 4

File Documentation

4.1 C:/Users/Jarek/source/repos/f5abc478-gr22-repo/projekt/hamilton_projekt/hamilton_projekt/hamilton_algorithm.h File Reference

```
#include "input_output.h"
```

Functions

- void [DFS](#) (int v, int ¤t_weight, int &best_weight, std::vector< bool > &visited, std::vector< int > ¤t_path, std::vector< int > &best_path, const [graph_struct](#) adjacency_list)
- bool [find_the_path](#) (const [graph_struct](#) adjacency_list, const [config_struct](#) config, std::vector< int > &best_path)

4.1.1 Function Documentation

4.1.1.1 DFS()

```
void DFS (
    int v,
    int & current_weight,
    int & best_weight,
    std::vector< bool > & visited,
    std::vector< int > & current_path,
    std::vector< int > & best_path,
    const graph\_struct adjacency_list )
```

Funkcja DFS przetwarza wszystkie dostępne wierzchołki grafu. Parametry są przekazane przez wartość lub referencję.

Parameters

	<i>v</i>	początkowy wierzchołek
in, out	<i>current_weight</i>	waga obecnej ścieżki
in, out	<i>best_weight</i>	waga ścieżki o najmniejszej wadze
in, out	<i>visited</i>	wektor odwiedzonych wierzchołków
in, out	<i>current_path</i>	wektor zawierający aktualną ścieżkę
	<i>best_path[in,out]</i>	wektor zawierający najkrótszą ścieżka Hamiltona
	<i>adjacencies_list</i>	lista sąsiedztwa zawierająca graf

4.1.1.2 find_the_path()

```
bool find_the_path (
    const graph_struct adjacency_list,
    const config_struct config,
    std::vector< int > & best_path )
```

Funkcja znajduje najkrótszą ścieżkę Hamiltona. Parametry są przekazane przez wartość lub referencję.

Parameters

<i>adjacency_list</i>	lista sąsiedztwa zawierająca graf
<i>config</i>	konfiguracje, zawierają początkowy wierzchołek ścieżki
<i>best_path[in,out]</i>	wektor zawierający najkrótszą ścieżka Hamiltona

Returns

zwraca true kiedy udało się znaleźć ścieżkę

4.2 hamilton_algorithm.h

[Go to the documentation of this file.](#)

```
00001
00002 #pragma once
00003
00004 #include "input_output.h"
00005
00016 void DFS(int v, int& current_weight, int& best_weight, std::vector<bool>& visited, std::vector<int>&
    current_path, std::vector<int>& best_path, const graph_struct adjacency_list);
00017
00025 bool find_the_path(const graph_struct adjacency_list, const config_struct config, std::vector<int>&
    best_path);
```

4.3 C:/Users/Jarek/source/repos/f5abc478-gr22-repo/projekt/hamilton_↵ projekt/hamilton_projekt/hamilton_algotihm.cpp File Reference

```
#include <iostream>
#include <vector>
#include <list>
#include "input_output.h"
```

Functions

- void **DFS** (int v, int ¤t_weight, int &best_weight, std::vector< bool > &visited, std::vector< int > ¤t_path, std::vector< int > &best_path, const **graph_struct** adjacency_list)
- bool **find_the_path** (const **graph_struct** adjacency_list, const **config_struct** config, std::vector< int > &best_↵_path)

4.3.1 Function Documentation

4.3.1.1 DFS()

```
void DFS (
    int v,
    int & current_weight,
    int & best_weight,
    std::vector< bool > & visited,
    std::vector< int > & current_path,
    std::vector< int > & best_path,
    const graph_struct adjacency_list )
```

Funkcja DFS przetwarza wszystkie dostępne wierzchołki grafu. Parametry są przekazane przez wartość lub referencję.

Parameters

	<i>v</i>	początkowy wierzchołek
<i>in, out</i>	<i>current_weight</i>	waga obecnej ścieżki
<i>in, out</i>	<i>best_weight</i>	waga ścieżki o najmniejszej wadze
<i>in, out</i>	<i>visited</i>	wektor odwiedzonych wierzchołków
<i>in, out</i>	<i>current_path</i>	wektor zawierający aktualnią ścieżkę
	<i>best_path[in,out]</i>	wektor zawierający najkrótszą ścieżka Hamiltona
	<i>adjacency_list</i>	lista sąsiedztwa zawierająca graf

4.3.1.2 find_the_path()

```
bool find_the_path (
    const graph_struct adjacency_list,
    const config_struct config,
    std::vector< int > & best_path )
```

Funkcja znajduje najkrótszą ścieżkę Hamiltona. Parametry są przekazane przez wartość lub referencję.

Parameters

<i>adjacency_list</i>	lista sąsiedztwa zawierająca graf
<i>config</i>	konfigurację, zawierają początkowy wierzchołek ścieżki
<i>best_path[in,out]</i>	wektor zawierający najkrótszą ścieżka Hamiltona

Returns

zwraca true kiedy udało się znaleźć ścieżkę

4.4 C:/Users/Jarek/source/repos/f5abc478-gr22-repo/projekt/hamilton_↵ projekt/hamilton_projekt/hamilton_main.cpp File Reference

```
#include <iostream>
#include <fstream>
```

```
#include <string>
#include <vector>
#include <list>
#include "input_output.h"
#include "hamilton_algorithm.h"
```

Functions

- int **main** (int argc, char **argv)

4.5 C:/Users/Jarek/source/repos/f5abc478-gr22-repo/projekt/hamilton_↵ projekt/hamilton_projekt/input_output.cpp File Reference

```
#include <iostream>
#include <fstream>
#include <string>
#include <vector>
#include <list>
#include <limits>
#include "input_output.h"
```

Functions

- bool [read_command_line_arguments](#) (cmd_args parameter, [config_struct](#) &config)
- bool [is_number](#) (const std::string &s)
- bool [read_graph](#) (const [config_struct](#) config, [graph_struct](#) &adjacency_list)
- bool [output_the_path](#) (const [config_struct](#) config, std::vector< int > best_path)

4.5.1 Function Documentation

4.5.1.1 is_number()

```
bool is_number (
    const std::string & s )
```

Funkcja sprawdza czy dany string jest liczbą. Parametr jest przekazany przez wartość.

Parameters

s	string do zbadanie
---	--------------------

Returns

zwraca true kiedy string jest liczbą

4.5.1.2 output_the_path()

```
bool output_the_path (
    const config_struct config,
    std::vector< int > best_path )
```

Funkcja generuje plik ze znalezioną najkrótszą ścieżką. Parametry są przekazane przez wartość.

Parameters

<i>config</i>	konfiguracje, zawierają nazwę pliku, do którego zapisujemy ścieżkę
<i>best_path</i>	najkrótsza ścieżka Hamiltona

Returns

zwraca true kiedy udało się wygenerować plik

4.5.1.3 read_command_line_arguments()

```
bool read_command_line_arguments (
    cmd_args parameter,
    config_struct & config )
```

Funkcja wczytuje parametry z wiersza poleceń. Parametry są przekazane przez wartość lub referencję.

Parameters

<i>parameter</i>	parametry wprowadzone przez użytkownika
<i>config[in,out]</i>	konfiguracje odczytane z parametrów

Returns

zwraca true kiedy udało się odczytać parametry

4.5.1.4 read_graph()

```
bool read_graph (
    const config_struct config,
    graph_struct & adjacency_list )
```

Funkcja wczytuje graf z pliku do struktury danych. Parametry są przekazane przez wartość lub referencję.

Parameters

	<i>config</i>	konfiguracje, zawierają nazwę pliku, z którego odczytujemy graf
<i>in, out</i>	<i>adjacency_list</i>	lista sąsiedztwa, do której wczytujemy graf

Returns

zwraca true kiedy udało się wczytać graf

4.6 C:/Users/Jarek/source/repos/f5abc478-gr22-repo/projekt/hamilton_↵ projekt/hamilton_projekt/input_output.h File Reference

Classes

- struct [cmd_args](#)
- struct [config_struct](#)
- struct [edge](#)
- struct [graph_struct](#)

Functions

- bool [read_command_line_arguments](#) ([cmd_args](#) parameter, [config_struct](#) &config)
- bool [is_number](#) (const std::string &s)
- bool [read_graph](#) (const [config_struct](#) config, [graph_struct](#) &adjacency_list)
- bool [output_the_path](#) (const [config_struct](#) config, std::vector< int > best_path)

4.6.1 Function Documentation

4.6.1.1 is_number()

```
bool is_number (
    const std::string & s )
```

Funkcja sprawdza czy dany string jest liczbą. Parametr jest przekazany przez wartość.

Parameters

s	string do zbadanie
---	--------------------

Returns

zwraca true kiedy string jest liczbą

4.6.1.2 output_the_path()

```
bool output_the_path (
    const config\_struct config,
    std::vector< int > best_path )
```

Funkcja generuje plik ze znalezioną najkrótszą ścieżką. Parametry są przekazane przez wartość.

Parameters

<i>config</i>	konfiguracje, zawierają nazwę pliku, do którego zapisujemy ścieżkę
<i>best_path</i>	najkrótsza ścieżka Hamiltona

Returns

zwraca true kiedy udało się wygenerować plik

4.6.1.3 read_command_line_arguments()

```
bool read_command_line_arguments (
    cmd_args parameter,
    config_struct & config )
```

Funkcja wczytuje parametry z wiersza poleceń. Parametry są przekazane przez wartość lub referencję.

Parameters

<i>parameter</i>	parametry wprowadzone przez użytkownika
<i>config[in,out]</i>	konfiguracje odczytane z parametrów

Returns

zwraca true kiedy udało się odczytać parametry

4.6.1.4 read_graph()

```
bool read_graph (
    const config_struct config,
    graph_struct & adjacency_list )
```

Funkcja wczytuje graf z pliku do struktury danych. Parametry są przekazane przez wartość lub referencję.

Parameters

	<i>config</i>	konfiguracje, zawierają nazwę pliku, z którego odczytujemy graf
<i>in, out</i>	<i>adjacency_list</i>	lista sąsiedztwa, do której wczytujemy graf

Returns

zwraca true kiedy udało się wczytać graf

4.7 input_output.h

[Go to the documentation of this file.](#)

```
00001
00002 #pragma once
00003
00004 struct cmd_args
00005 {
00006     int argc;
00007     char** argv;
00008 };
00009
00010 struct config_struct
00011 {
00012     int vertex;
00013     std::string input;
00014     std::string output;
00015 };
00016
00017 struct edge
00018 {
00019     int vertex; //vertex to which the edge points to
00020     int weight; //weight of the edge
00021 };
00022
00023 struct graph_struct
00024 {
00025     std::vector<std::list<edge>> graph;
00026 };
00027
00034 bool read_command_line_arguments(cmd_args parameter, config_struct& config);
00035
00041 bool is_number(const std::string& s);
00042
00049 bool read_graph(const config_struct config, graph_struct& adjacency_list);
00050
00057 bool output_the_path(const config_struct config, std::vector<int> best_path);
```

Index

C:/Users/Jarek/source/repos/f5abc478-gr22-repo/projekt/hamilton_projekt/hamilton_projekt/hamilton_algorithm.h,
7, 8
input_output.h, 13
C:/Users/Jarek/source/repos/f5abc478-gr22-repo/projekt/hamilton_projekt/hamilton_algotihm.cpp,
8
input_output.cpp, 11
C:/Users/Jarek/source/repos/f5abc478-gr22-repo/projekt/hamilton_projekt/hamilton_main.cpp,
9
C:/Users/Jarek/source/repos/f5abc478-gr22-repo/projekt/hamilton_projekt/hamilton_projekt/input_output.cpp,
10
C:/Users/Jarek/source/repos/f5abc478-gr22-repo/projekt/hamilton_projekt/hamilton_projekt/input_output.h,
12, 13
cmd_args, 5
config_struct, 5

DFS
 hamilton_algorithm.h, 7
 hamilton_algotihm.cpp, 9

edge, 5

find_the_path
 hamilton_algorithm.h, 8
 hamilton_algotihm.cpp, 9

graph_struct, 6

hamilton_algorithm.h
 DFS, 7
 find_the_path, 8
hamilton_algotihm.cpp
 DFS, 9
 find_the_path, 9

input_output.cpp
 is_number, 10
 output_the_path, 10
 read_command_line_arguments, 11
 read_graph, 11
input_output.h
 is_number, 12
 output_the_path, 12
 read_command_line_arguments, 13
 read_graph, 13
is_number
 input_output.cpp, 10
 input_output.h, 12

output_the_path
 input_output.cpp, 10
 input_output.h, 12

read_command_line_arguments