

GRAF - dokumentacja

Generated by Doxygen 1.8.6

Wed May 28 2014 15:07:16

Contents

1	Class Index	1
1.1	Class List	1
2	File Index	3
2.1	File List	3
3	Class Documentation	5
3.1	Graf Class Reference	5
3.1.1	Detailed Description	6
3.1.2	Constructor & Destructor Documentation	6
3.1.2.1	~Graf	6
3.1.3	Member Function Documentation	6
3.1.3.1	a_star	6
3.1.3.2	BFS	6
3.1.3.3	czy_spojny	7
3.1.3.4	DFS	7
3.1.3.5	generuj_macierz	7
3.1.3.6	wczytaj_graf	7
3.1.3.7	wyswietl	7
3.1.4	Member Data Documentation	7
3.1.4.1	E	7
3.1.4.2	Tablica	7
3.1.4.3	V	7
3.1.4.4	wierzcholek	8
3.2	porownanie Class Reference	8
3.2.1	Detailed Description	8
3.2.2	Member Function Documentation	8
3.2.2.1	operator()	8
3.3	wezel Struct Reference	8
3.3.1	Detailed Description	9
3.3.2	Constructor & Destructor Documentation	9
3.3.2.1	wezel	9

3.3.3	Member Function Documentation	9
3.3.3.1	heurystyka	9
3.3.4	Member Data Documentation	10
3.3.4.1	akt_cel	10
3.3.4.2	koszt	10
3.3.4.3	ktory	10
3.3.4.4	start_akt	10
3.3.4.5	x	10
3.3.4.6	y	10
4	File Documentation	11
4.1	C:/Users/Ania/Documents/GitHub/sort/inc/graf.hh File Reference	11
4.1.1	Detailed Description	11
4.1.2	Typedef Documentation	11
4.1.2.1	wezel	12
4.2	C:/Users/Ania/Documents/GitHub/sort/src/graf.cpp File Reference	12
4.3	C:/Users/Ania/Documents/GitHub/sort/src/main.cpp File Reference	12
4.3.1	Function Documentation	12
4.3.1.1	main	12
	Index	13

Chapter 1

Class Index

1.1 Class List

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

Graf	Klasa graf w ktorej zdefiniowane sa wszystkie parametry i funkcje zwiazane z operacjami przeprowadzanymi na nim	5
porownanie	Klasa porownanie wykorzystywana do poszukiwania A*. Sluzaca do definicji kolejki piorytowej - abstrakcyjnego typu danych sluzacego do przechowywania elemntow zbioru w ustalonym porzadku	8
wezel	Struktura w ktorej definiujemy wezel- wiercholek	8

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

C:/Users/Ania/Documents/GitHub/sort/inc/ graf.hh	
Plik w którym wczytujemy plik tworzymy macierz sasiedztwa oraz zdefiniowane sa funkcje zwiazane z grafem do przesukiwan drogi miedzy punktami	11
C:/Users/Ania/Documents/GitHub/sort/src/ graf.cpp	12
C:/Users/Ania/Documents/GitHub/sort/src/ main.cpp	12

Chapter 3

Class Documentation

3.1 Graf Class Reference

Klasa graf w ktorej zdefiniowane sa wszystkie parametry i funkcje zwiazane z operacjami przeprowadzanymi na nim.

```
#include <graf.hh>
```

Public Member Functions

- void **generuj_macierz** (int ilosc, int gestosc)
- bool **czy_spojny** ()
- void **wyswietl** ()
Funkcja wyswietla graf wczytany z pliku, a dokładnie wyswietla jego macierz sasiedztwa.
- void **wczytaj_graf** (string nazwa_pliku)
Funkcja wczytuje graf z pliku (jego macierz sasiedztwa) i zapisuje do pamieci abysmy mogli nastepnie uzywac.
- int **BFS** (int pocz, int koniec)
BST - Breadth-first search Funkcja przeszukiwania grafu wszere. Przechodzenie grafu rozpoczyna sie od zadanego wierzcholka i polega na odwiedzeniu wszystkich osiagalnych z niego wierzcholkow w celu odnalezienia drogi do drugiego wierzcholka. Funkcja dziala na macierzy sasiedztwa.
- int **DFS** (int pocz, int koniec)
DST - Depth-first search Funkcja przeszukiwania grafu w glab. Przechodzenie grafu rozpoczyna sie od zadanego wierzcholka i polega na badaniu wszystkich krawedzi wychodzacych z podanego wierzcholka. Po zbadaniu wszystkich krawedzi wychodzacych z danego wierzcholka algorytm poawraca do wierzcholka, z ktorego dany wierzcholek zostal odiwedzony. Funkcja uzyta w celu odnalezienia drogi do drugiego wierzcholka. Funkcja dziala na macierzy sasiedztwa.
- int **a_star** (int pocz, int koniec)
A - A star Algorytm heurystyczny sluzacy do odnalezienia najkrotszej sciezki w grafie. Algorytm jest zupelny i optymalny, znajduje sciezke jezeli tylko taka istnieje i przy tym najkrtosza sciezke. Algorytm A* od poczatku tworzy sciezke, za kazdym razem wybierajac wierzcholek x z dostepnych w danym kroku niezbadanych wierzcholkow tak aby minimalizowac funkcje $f(x)$ - zmienna koszt. Funkcja ta nigdy nie przeoczy sciezki o nizszym koszcie, dlatego wyszukuje najkrotsza droge sposrod trzech przeszukiwan w moim programie (BFS, DFS, A*).*
- **~Graf** ()
*Dekonstruktor klasy **Graf** (p. 5). Gdy zakonczymy dzialanie tej klasie pamiec zostaje zwolniona.*

Public Attributes

- int ** **Tablica**
Tablica, ktora jest macierza sasiedztwa.
- int **V**

Zmienna wykorzystywana do przechowywania liczby wierzchołkow w grafie.

- **int E**

Zmienna wykorzystywana do przechowywania liczby krawedzi uzytych w grafie.

- **wezel * wierzcholek**

Lista wierzchołkow.

3.1.1 Detailed Description

Klasa graf w ktorej zdefiniowane sa wszystkie parametry i funkcje zwiazane z operacjami przeprowadzanymi na nim.

3.1.2 Constructor & Destructor Documentation

3.1.2.1 Graf::~Graf ()

Dekonstruktor klasy **Graf** (p. 5). Gdy zakonczymy dzialanie tej klasie pamiec zostaje zwolniona.

3.1.3 Member Function Documentation

3.1.3.1 int Graf::a_star (int pocz, int koniec)

A* - A star Algorytm heurystyczny sluzacy do odnalezienia najkrotszej sciezki w grafie. Algorytm jest zupelny i optymalny, znajduje sciezke jezeli tylko taka istnieje i przy tym najkrotsza sciezke. Algorytm A* od poczatku tworzy sciezke, za kazdym razem wybierajac wierzcholek x z dostepnych w danym kroku niezbadanych wierzchołkow tak aby minimalizowac funkcje $f(x)$ - zmienna koszt. Funkcja ta nigdy nie przeoczy sciezki o nizszym koszcie, dlatego wyszukuje najkrotsza droge sposrod trzech przeszukiwan w moim programie (BFS, DFS, A*).

Parameters

in	pocz	- punkt poczatkowy do znalezienia drogi w grafie
in	koniec	- punkt koncowy do znalezienia drogi w grafie

Returns

Dzieki funkcji otrzymac mozemy dlugosc (koszt) drogi z punkty A do B.

3.1.3.2 int Graf::BFS (int pocz, int koniec)

BST - Breadth-first search Funkcja przeszukiwania grafu wszere. Przechodzenie grafu rozpoczyna sie od zadanego wierzchołka i polega na odwiedzeniu wszystkich osiagalnych z niego wierzchołkow w celu odnalezienia drogi do drugiego wierzchołka. Funkcja dziala na macierzy sasiedztwa.

Parameters

in	pocz	- punkt poczatkowy do znalezienia drogi w grafie
in	koniec	- punkt koncowy do znalezienia drogi w grafie

Returns

Dzieki funkcji otrzymac mozemy dlugosc (koszt) drogi z punkty A do B.

3.1.3.3 bool Graf::czy_spojny ()

3.1.3.4 int Graf::DFS (int *pocz*, int *koniec*)

DST - Depth-first search Funkcja przeszukiwania grafu w głąb. Przechodzenie grafu rozpoczyna się od zadanego wierzchołka i polega na badaniu wszystkich krawędzi wychodzących z podanego wierzchołka. Po zbadaniu wszystkich krawędzi wychodzących z danego wierzchołka algorytm poawraca do wierzchołka, z którego dany wierzchołek został odiwiedzony. Funkcja użyta w celu odnalezienia drogi do drugiego wierzchołka. Funkcja działa na macierzy sąsiedztwa.

Parameters

in	<i>pocz</i>	- punkt początkowy do znalezienia drogi w grafie
in	<i>koniec</i>	- punkt końcowy do znalezienia drogi w grafie

Returns

Dzięki funkcji otrzymać możemy długość (koszt) drogi z punktu A do B.

3.1.3.5 void Graf::generuj_macierz (int *ilosc*, int *gestosc*)

3.1.3.6 void Graf::wczytaj_graf (string *nazwa_pliku*)

Funkcja wczytuje graf z pliku (jego macierz sąsiedztwa) i zapisuje do pamięci abyśmy mogli następnie używać.

Returns

(brak)

3.1.3.7 void Graf::wyswietl ()

Funkcja wyświetla graf wczytany z pliku, a dokładnie wyświetla jego macierz sąsiedztwa.

Returns

(brak)

3.1.4 Member Data Documentation

3.1.4.1 int Graf::E

Zmienna wykorzystywana do przechowywania liczby krawędzi użytych w grafie.

3.1.4.2 int** Graf::Tablica

Tablica, która jest macierzą sąsiedztwa.

3.1.4.3 int Graf::V

Zmienna wykorzystywana do przechowywania liczby wierzchołków w grafie.

3.1.4.4 wezel* Graf::wierzcholek

Lista wierzchołkow.

The documentation for this class was generated from the following files:

- C:/Users/Ania/Documents/GitHub/sort/inc/**graf.hh**
- C:/Users/Ania/Documents/GitHub/sort/src/**graf.cpp**

3.2 porownanie Class Reference

Klasa porownanie wykorzystywana do poszukiwania A*. Sluzaca do definicji kolejki priorytetowej - abstrakcyjnego typu danych sluzacego do przechowywania elemntow zbioru w ustalonym porzadku.

```
#include <graf.hh>
```

Public Member Functions

- bool **operator()** (const **wezel** &pier, const **wezel** &drugi) const

Funkcja pomocna do ustalenia porzadku w jakim maja byc przechowywane wierzcholki nieodwiedzone. W naszym przypadku ma byc uporządkowane według kosztu wierzcholka, czyli wagi rosnaco.

3.2.1 Detailed Description

Klasa porownanie wykorzystywana do poszukiwania A*. Sluzaca do definicji kolejki priorytetowej - abstrakcyjnego typu danych sluzacego do przechowywania elemntow zbioru w ustalonym porzadku.

3.2.2 Member Function Documentation

3.2.2.1 bool porownanie::operator() (const wezel & pier, const wezel & drugi) const [inline]

Funkcja pomocna do ustalenia porzadku w jakim maja byc przechowywane wierzcholki nieodwiedzone. W naszym przypadku ma byc uporządkowane według kosztu wierzcholka, czyli wagi rosnaco.

Parameters

in	<i>pier</i>	- wezel uzyty do porownywania wag z innym wezelem
in	<i>drugi</i>	- wezel uzyty do porownywania wag z innym wezelem

Returns

Funkcja zwraca true jezeli waga pier jest wieksza niz drugiego w przeciwnym wypadku false.

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

- C:/Users/Ania/Documents/GitHub/sort/inc/**graf.hh**

3.3 wezel Struct Reference

Struktura w ktorej definiujemy wezel- wierzcholek.

```
#include <graf.hh>
```

Public Member Functions

- double **heurystyka** (**wezel** cel)

Funkcja znajdujaca rozwiazanie, dla ktorej nie ma gwarancji znalezienia metody optymalnej. Sluzy do znajdowania rozwiazan przyblizonych, na podstawie ktorej wyliczany jest ostateczny rezultat. W funkcji znajdujemy droge pomiedzy dwoma wierzchołkami w linii prostej na podstawie twierdzenia Pitagorasa. Kazdy wierzcholek ma wspolrzedne x i y.

- **wezel** ()

Konstruktor w ktorym definiujemy poczatowe wartosci wezla aby informacje o wierzchołku nie przyjmowaly nam losowych wartosci.

Public Attributes

- int **ktory**

Numer wierzchoła, jego ID.

- int **start_akt**

Droga jak pokona algorytm poszukiwania binarnego od pocztka do miejsca w ktorym situlanie znajdujemy.

- double **akt_cel**

Droga jaka zostaa do pokonoania. Zmienna wykorzystywana w poszukiwaniu A. Do zmiennej przypisujemy heurystyke, ktora wyszukuje nam optymalna droge.*

- double **koszt**

Suma zmiennych start_akt i akt_cel, czyli jest to droga jaka prawdopodobnie trzeba pokonac aby przejść punktu A do punktu B.

- int **x**

Wspolrzedna x wierzchoła wykorzystywana w A.*

- int **y**

Wspolrzedna y wierzchoła wykorzystywana w A.*

3.3.1 Detailed Description

Struktura w ktorej definiujemy wezel- wierzcholek.

3.3.2 Constructor & Destructor Documentation

3.3.2.1 wezel::wezel () [inline]

Konstruktor w ktorym definiujemy poczatowe wartosci wezla aby informacje o wierzchołku nie przyjmowaly nam losowych wartosci.

3.3.3 Member Function Documentation

3.3.3.1 double wezel::heurystyka (wezel cel)

Funkcja znajdujaca rozwiazanie, dla ktorej nie ma gwarancji znalezienia metody optymalnej. Sluzy do znajdowania rozwiazan przyblizonych, na podstawie ktorej wyliczany jest ostateczny rezultat. W funkcji znajdujemy droge pomiedzy dwoma wierzchołkami w linii prostej na podstawie twierdzenia Pitagorasa. Kazdy wierzcholek ma wspolrzedne x i y.

Parameters

<code>in</code>	<code>cel</code>	- informacje ktore sa zawarte w wierzcholku do ktorego mamy znalezc droge
-----------------	------------------	---

Returns

Funkcja zwraca liczbe zmiennoprzecinkowa, ktora jest najkrotsza droga z punktu A do B.

3.3.4 Member Data Documentation

3.3.4.1 `double wezel::akt_cel`

Droga jaka zostaa do pokonoania. Zmienna wykorzystywana w poszukiwaniu A*. Do zmiennej przypisujemy heurystyke, ktora wyszukuje nam optymalna droge.

3.3.4.2 `double wezel::koszt`

Suma zmiennych `start_akt` i `akt_cel`, czyli jest to droga jaka prawdopodobnie trzeba pokonac aby przejść punktu A do punktu B.

3.3.4.3 `int wezel::ktory`

Numer wierzchoka, jego ID.

3.3.4.4 `int wezel::start_akt`

Droga jak pokona algorytm poszukiwania binarnego od pocztku do miejsca w ktorym situlanie znajdujemy.

3.3.4.5 `int wezel::x`

Wspolrzedna x wierzcholka wykorzystywana w A*.

3.3.4.6 `int wezel::y`

Wspolrzedna y wierzcholka wykorzystywana w A*.

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

- `C:/Users/Ania/Documents/GitHub/sort/inc/graf.hh`
- `C:/Users/Ania/Documents/GitHub/sort/src/graf.cpp`

Chapter 4

File Documentation

4.1 C:/Users/Ania/Documents/GitHub/sort/inc/graf.hh File Reference

Plik w którym wczytujemy plik tworzymy macierz sasiedztwa oraz zdefiniowane sa funkcje zwiazane z grafem do przesukiwan drogi miedzy punktami.

```
#include <iostream>
#include <cstdlib>
#include <fstream>
#include <vector>
#include <algorithm>
#include <iomanip>
#include <stack>
```

Classes

- struct **wezel**

Struktura w ktorej definiujemy wezel- wiercholek.

- class **porownanie**

Klasa porownanie wykorzystywana do poszukiwania A. Sluzaca do definicji kolejki priorytetowej - abstrakcyjnego typu danych sluzacego do przechowywania elemntow zbioru w ustalonym porzadku.*

- class **Graf**

Klasa graf w ktorej zdefiniowane sa wszystkie parametry i funkcje zwiazane z operacjami przeprowadzanymi na nim.

Typedefs

- typedef struct **wezel wezel**

Struktura w ktorej definiujemy wezel- wiercholek.

4.1.1 Detailed Description

Plik w którym wczytujemy plik tworzymy macierz sasiedztwa oraz zdefiniowane sa funkcje zwiazane z grafem do przesukiwan drogi miedzy punktami.

4.1.2 Typedef Documentation

4.1.2.1 typedef struct wezel wezel

Struktura w ktorej definiujemy wezel- wiercholek.

4.2 C:/Users/Ania/Documents/GitHub/sort/src/graf.cpp File Reference

```
#include "graf.hh"  
#include <queue>  
#include <stack>  
#include <cmath>
```

4.3 C:/Users/Ania/Documents/GitHub/sort/src/main.cpp File Reference

```
#include "graf.hh"  
#include <queue>
```

Functions

- int **main** ()

4.3.1 Function Documentation

4.3.1.1 int main ()

Index

- ~Graf
 - Graf, 6
- a_star
 - Graf, 6
- akt_cel
 - wezel, 10
- BFS
 - Graf, 6
- C:/Users/Ania/Documents/GitHub/sort/inc/graf.hh, 11
- C:/Users/Ania/Documents/GitHub/sort/src/graf.cpp, 12
- C:/Users/Ania/Documents/GitHub/sort/src/main.cpp, 12
- czy_spojny
 - Graf, 6
- DFS
 - Graf, 7
- E
 - Graf, 7
- generuj_macierz
 - Graf, 7
- Graf, 5
 - ~Graf, 6
 - a_star, 6
 - BFS, 6
 - czy_spojny, 6
 - DFS, 7
 - E, 7
 - generuj_macierz, 7
 - Tablica, 7
 - V, 7
 - wczytaj_graf, 7
 - wierzcholek, 7
 - wyswietl, 7
- graf.hh
 - wezel, 11
- heurystyka
 - wezel, 9
- koszt
 - wezel, 10
- ktory
 - wezel, 10
- main
 - main.cpp, 12
- main.cpp
 - main, 12
- operator()
 - porownanie, 8
- porownanie, 8
 - operator(), 8
- start_akt
 - wezel, 10
- Tablica
 - Graf, 7
- V
 - Graf, 7
- wczytaj_graf
 - Graf, 7
- wezel, 8
 - akt_cel, 10
 - graf.hh, 11
 - heurystyka, 9
 - koszt, 10
 - ktory, 10
 - start_akt, 10
 - wezel, 9
 - x, 10
 - y, 10
- wierzcholek
 - Graf, 7
- wyswietl
 - Graf, 7
- x
 - wezel, 10
- y
 - wezel, 10