

Tablice Routingu

Wygenerowano przez Doxygen 1.8.20

1 Indeks struktur danych	1
1.1 Struktury danych	1
2 Indeks plików	3
2.1 Lista plików	3
3 Dokumentacja struktur danych	5
3.1 Dokumentacja struktury połączenie	5
3.1.1 Opis szczegółowy	5
3.2 Dokumentacja struktury wierzcholek	5
3.2.1 Opis szczegółowy	6
4 Dokumentacja plików	7
4.1 Dokumentacja pliku Funkcje.h	7
4.1.1 Dokumentacja funkcji	7
4.1.1.1 dijkstra()	7
4.1.1.2 iloscPolaczen()	8
4.1.1.3 iloscWierzcholek()	8
4.1.1.4 inicjalizacja()	9
4.1.1.5 wczytajPolaczenia()	9
4.1.1.6 wczytajWierzcholki()	10
4.1.1.7 wierzcholkiPozostale()	10
4.1.1.8 wyczyscPlikWyjsciowy()	10
4.1.1.9 wywolajDlaWszystkich()	11
4.1.1.10 zapiszNaglowekDoPliku()	11
4.1.1.11 zapiszSciezkeDoPliku()	12
4.1.1.12 zwolnijPamiecListy()	12
4.2 Dokumentacja pliku Struktury.h	12
Indeks	13

Rozdział 1

Indeks struktur danych

1.1 Struktury danych

Tutaj znajdują się struktury danych wraz z ich krótkimi opisami:

połączenie	5
wierzcholek	5

Rozdział 2

Indeks plików

2.1 Lista plików

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

Funkcje.h	7
Struktury.h	12

Rozdział 3

Dokumentacja struktur danych

3.1 Dokumentacja struktury połączenie

```
#include <Struktury.h>
```

Pola danych

- int **odleglosc**
Odleglosc wierzchołka (do którego należy ta lista połączeń) od wierzchołka będącego obecnym elementem listy.
- struct **wierzcholek** * **wierzcholekPolaczony**
Wskaźnik na wierzchołek, do którego prowadzi to połączenie.
- struct **polaczenie** * **nastepnePolaczenie**
Następny element listy połączeń.

3.1.1 Opis szczegółowy

Lista jednokierunkowa połączeń wychodzących z danego wierzchołka. Każdy wierzchołek ma listę połączeń z niego wychodzących.

Dokumentacja dla tej struktury została wygenerowana z pliku:

- **Struktury.h**

3.2 Dokumentacja struktury wierzcholek

```
#include <Struktury.h>
```

Pola danych

- char * **nazwa**
Nazwa wierzchołka, np. A1.
- bool **odwiedzony**
Czy wierzchołek był już odwiedzony - ta zmienna potrzebna nam jest do algorytmu Dijkstry.
- int **koszt**
Koszt poruszenia się do wierzchołka - ta zmienna potrzebna nam jest do algorytmu Dijkstry.
- struct **wierzcholek** * **poprzedni**
Poprzedni wierzchołek.
- struct **wierzcholek** * **nastepny**
Następny wierzchołek.
- struct **polaczenie** * **polaczenie**
Lista jednokierunkowa połączeń z danego wierzchołka.

3.2.1 Opis szczegółowy

Struktura odpowiedzialna za wierzchołek grafu - zawiera jego nazwę, listę jednokierunkowych połączeń wychodzących z niego oraz zmienne potrzebne przy implementacji algorytmu Dijkstry.

Dokumentacja dla tej struktury została wygenerowana z pliku:

- **Struktury.h**

Rozdział 4

Dokumentacja plików

4.1 Dokumentacja pliku Funkcje.h

```
#include <stdbool.h>
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include "Struktury.h"
```

Funkcje

- char * **wczytajWierzcholki** (char *nazwaPliku)
- char * **wczytajPolaczenia** (char *nazwaPliku)
- int **iloscWierzcholkow** (char *nazwaPliku)
- int **iloscPolaczen** (char *nazwaPliku)
- char * **wierzcholkiPozostale** (char *wierzcholki, char *wierzcholekPoczatkowy)
- void **zapiszNaglowekDoPliku** (char *wyjscie, char *wierzcholekPoczatkowy)
- void **zapiszSciezkeDoPliku** (char *wyjscie, char *sciezka, float lacznyKoszt)
- void **wyczyscPlikWyjsciowy** (char *wyjscie)
- struct **wierzcholek** * **inicjalizacja** (char *nazwaPliku, char *wierzcholekPoczatkowy)
- void **wywołajDlaWszystkich** (char *wierzcholekPoczatkowy, char *pozostale, char *wejscie, char *wyjscie)
- int **dijkstra** (struct **wierzcholek** *head, int ileWierzcholkow, int ilePolaczen, char *wierzcholekPoczatkowy, char *wierzcholekDocelowy)
- void **zwolnijPamiecListy** (struct **wierzcholek** *head)

4.1.1 Dokumentacja funkcji

4.1.1.1 dijkstra()

```
int dijkstra (
    struct wierzcholek * head,
    int ileWierzcholkow,
    int ilePolaczen,
    char * wierzcholekPoczatkowy,
    char * wierzcholekDocelowy )
```

Wykorzystując algorytm Dijkstry wyznacza najbardziej korzystne drogi z jednego wierzchołka do drugiego

Parametry

<i>head</i>	Wskaźnik na pierwszy element listy wierzchołków
<i>ileWierzchołkow</i>	Ilość wierzchołków, znaleziona za pomocą funkcji <code>iloscWierzchołkow</code>
<i>ilePolaczen</i>	Ilość połączeń, znaleziona za pomocą funkcji <code>iloscPolaczen</code>
<i>wierzcholekPoczątkowy</i>	Nazwa wierzchołka, od którego zaczynamy się poruszać (np. A1)
<i>wierzcholekDocelowy</i>	Nazwa wierzchołka, do którego chcemy się poruszyć (np. D1)

Zwraca

Koszt poruszenia się z wierzchołka początkowego do docelowego (np. 5)

Nota

Funkcja jest wywoływana dla par wierzchołków przez funkcję "wywołajDlaWszystkich"

4.1.1.2 iloscPolaczen()

```
int iloscPolaczen (
    char * nazwaPliku )
```

Zlicza ilość połączeń w pliku (ilość elementów poza pierwszym w każdym wierszu)

Parametry

<i>nazwaPliku</i>	Napis zawierający nazwę pliku wejściowego
-------------------	---

Zwraca

Ilość połączeń w pliku wejściowym

Nota

Uznajemy, że każde połączenie zakończone jest średnikiem

4.1.1.3 iloscWierzchołkow()

```
int iloscWierzchołkow (
    char * nazwaPliku )
```

Zlicza ilość wierzchołków w pliku (tylko pierwszy element w każdym wierszu)

Parametry

<i>nazwaPliku</i>	Napis zawierający nazwę pliku wejściowego
-------------------	---

Zwraca

Ilość wierzchołków w pliku wejściowym

Nota

Uznajemy, że każda linia zawiera opis jednego wierzchołka (puste linie są pomijane)

4.1.1.4 inicjalizacja()

```
struct wierzcholek* inicjalizacja (
    char * nazwaPliku,
    char * wierzcholekPoczatkowy )
```

Inicjalizuje listę wierzchołków nazwaną i zwraca wskaźnik na pierwszy jej element

Parametry

<i>nazwaPliku</i>	Napis zawierający nazwę pliku wejściowego
<i>wierzcholekPoczatkowy</i>	Napis zawierający nazwę wierzchołka początkowego (od którego zaczynamy szukać ścieżki do kolejnych)

Zwraca

Wskaźnik na pierwszy element listy wierzchołków

4.1.1.5 wczytajPolaczenia()

```
char* wczytajPolaczenia (
    char * nazwaPliku )
```

Wczytuje połączenia z pliku wejściowego o podanej nazwie jako jeden napis

Parametry

<i>nazwaPliku</i>	Napis zawierający nazwę pliku wejściowego
-------------------	---

Zwraca

Napis zawierający wszystkie połączenia oddzielone przecinkami, np. A1_B1_1,A1_C1_2

4.1.1.6 wczytajWierzcholki()

```
char* wczytajWierzcholki (
    char * nazwaPliku )
```

Wczytuje wierzcholki z pliku wejściowego o podanej nazwie jako jeden napis

Parametry

<i>nazwaPliku</i>	Napis zawierający nazwę pliku wejściowego
-------------------	---

Zwraca

Napis zawierający wszystkie wierzcholki oddzielone przecinkami, np. A1,B1,C1,D1

4.1.1.7 wierzcholkiPozostale()

```
char* wierzcholkiPozostale (
    char * wierzcholki,
    char * wierzcholekPoczątkowy )
```

Zwraca napis zawierający wszystkie nazwy wierzchołków oprócz wierzchołka początkowego, podanego jako parametr

Parametry

<i>wierzcholki</i>	Napis zawierający wierzcholki wczytane z pliku (zwrócony przez funkcję "wczytajWierzcholki")
<i>wierzcholekPoczątkowy</i>	Napis zawierający nazwę wierzchołka początkowego (od którego zaczynamy szukać ścieżki do kolejnych) @return Napis zawierający wszystkie wierzcholki oprócz początkowego, oddzielone przecinkami, np. B1,C1,D1,E1,F1

4.1.1.8 wyczyscPlikWyjściowy()

```
void wyczyscPlikWyjściowy (
    char * wyjście )
```

Czysta plik wyjściowy, tak by przy każdym uruchomieniu programu znajdowały się w nim tylko obecne dane wyjściowe

Parametry

<i>wyjście</i>	Napis zawierający nazwę pliku wyjściowego
----------------	---

Nota

Funkcja otwiera plik wyjściowy z przełącznikiem "w", co usuwa całą jego zawartość. Jeśli plik nie istnieje, zostanie stworzony (i również będzie pusty)

4.1.1.9 wywołajDlaWszystkich()

```
void wywołajDlaWszystkich (
    char * wierzcholekPoczątkowy,
    char * pozostałe,
    char * wejście,
    char * wyjście )
```

Wywołuje algorytm Dijkstry (funkcja "dijkstra") dla wszystkich wierzchołków oprócz wierzchołka początkowego, podanego jako parametr

Parametry

<i>wierzcholekPoczątkowy</i>	Wierzchołek, dla którego będziemy wywoływać algorytm
<i>pozostałe</i>	Napis zawierający wierzchołki pozostałe (wszystkie poza początkowym - zwrócony przez funkcję wierzchołkiPozostałe)
<i>wyjście</i>	Napis zawierający nazwę pliku wyjściowego

Nota

Ta funkcja wykona algorytm Dijkstry dla wierzchołka początkowego oraz każdego wierzchołka zawartego w napisie "pozostałe"

Funkcja zapisuje do pliku wyjściowego wyniki działania algorytmu (ścieżki wraz ze średnimi kosztami)

4.1.1.10 zapiszNaglowekDoPliku()

```
void zapiszNaglowekDoPliku (
    char * wyjście,
    char * wierzcholekPoczątkowy )
```

Zapisuje nagłówek do pliku wyjściowego (aby format był zgodny z poleceniem)

Parametry

<i>wyjście</i>	Napis zawierający nazwę pliku wyjściowego
<i>wierzcholekPoczątkowy</i>	Napis zawierający nazwę wierzchołka początkowego

Nota

Funkcja zapisze do pliku napis np. "dla wierzcholka A1"

4.1.1.11 zapiszSciezkeDoPliku()

```
void zapiszSciezkeDoPliku (
    char * wyjście,
    char * sciezka,
    float lacznyKoszt )
```

Wypisuje do pliku wyjściowego o podanej nazwie sciezke oraz laczny koszt

Parametry

<i>wyjście</i>	Napis zawierający nazwę pliku wyjściowego
<i>sciezka</i>	Napis zawierający sciezke do wierzcholka docelowego, w następującym formacie: do W1 przez (W2) : koszt
<i>lacznyKoszt</i>	Lacny koszt dla tej sciezki (moze nie byc calkowity)

4.1.1.12 zwolnijPamiecListy()

```
void zwolnijPamiecListy (
    struct wierzcholek * head )
```

Zwalnia pamiec uzyta na liste wierzchołkow, listy polaczen oraz napisy

Parametry

<i>head</i>	Wskaznik na pierwszy element listy, ktorej pamiec zwalniamy
-------------	---

Nota

Funkcja zwolni rowniez pamiec zaalokowana na listy polaczen oraz napisy

4.2 Dokumentacja pliku Struktury.h

Struktury danych

- struct **polaczenie**
- struct **wierzcholek**

Zmienne

- struct **polaczenie** * **Polaczenie**
- struct **wierzcholek** * **Wierzcholek**

Indeks

dijkstra

Funkcje.h, 7

Funkcje.h, 7

dijkstra, 7

iloscPolaczen, 8

iloscWierzcholkow, 8

inicjalizacja, 9

wczytajPolaczenia, 9

wczytajWierzcholki, 10

wierzcholkiPozostale, 10

wyczyscPlikWyjscowy, 10

wywolajDlaWszystkich, 11

zapiszNaglowekDoPliku, 11

zapiszSciezkeDoPliku, 12

zwolnijPamiecListy, 12

iloscPolaczen

Funkcje.h, 8

iloscWierzcholkow

Funkcje.h, 8

inicjalizacja

Funkcje.h, 9

polaczenie, 5

Struktury.h, 12

wczytajPolaczenia

Funkcje.h, 9

wczytajWierzcholki

Funkcje.h, 10

wierzcholek, 5

wierzcholkiPozostale

Funkcje.h, 10

wyczyscPlikWyjscowy

Funkcje.h, 10

wywolajDlaWszystkich

Funkcje.h, 11

zapiszNaglowekDoPliku

Funkcje.h, 11

zapiszSciezkeDoPliku

Funkcje.h, 12

zwolnijPamiecListy

Funkcje.h, 12