TSP

Wygenerowano przez Doxygen 1.9.6

Rozdział 1

Indeks klas

1.1 Lista klas

Tutaj znajdują się klasy, struktury, unie i interfejsy wraz z ich krótkimi opisami:

Rozdział 2

Indeks plików

2.1 Lista plików

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

Rozdział 3

Dokumentacja klas

3.1 Dokumentacja struktury Graph

Atrybuty publiczne

- std::vector< char > **cities**
- std::map< std::pair< char, char >, int > **map**

Dokumentacja dla tej struktury została wygenerowana z pliku:

- Graph.h

Rozdział 4

Dokumentacja plików

4.1 Dokumentacja pliku File.cpp

Funkcje operacji na plikach.

```
#include "File.h"
```

Funkcje

- std::vector< std::string > split (std::string str)

  Dzielenie tekstu na fragmenty względem znaku.
- Graph loadGraph (std::string graphfile)

  Wczytywanie grafu z pliku.
- void saveGraph (std::vector< std::pair< std::vector< char >, int > > tsp, std::string filename)

  Zapis wyniku do pliku.

4.1.1 Opis szczegółowy

Funkcje operacji na plikach.

Autor

Piotr Nowak (nowakpiotr510@gmail.com)

Wersja

0.1

Data

2023-02-06

Copyright

Copyright (c) 2023

4.1.2 Dokumentacja funkcji

4.1.2.1 loadGraph()

```
Graph loadGraph (
            std::string graphfile )
```

Wczytywanie grafu z pliku.

Parametry

routefile	
graphfile	

Zwraca

Graph

4.1.2.2 saveGraph()

```
void saveGraph (
            std::vector< std::pair< std::vector< char >, int > > tsp,
            std::string filename )
```

Zapis wyniku do pliku.

Parametry

tsp	
filename	

4.1.2.3 split()

```
std::vector< std::string > split (
            std::string str )
```

Dzielenie tekstu na fragmenty względem znaku.

Parametry

str	

Zwraca

std::vector<std::string>

4.2 Dokumentacja pliku File.h

Plik nagłówkowy operacji na plikach.

```
#include "TSP.h"
#include <fstream>
#include <string>
#include <sstream>
#include <iostream>
```

Funkcje

- Graph loadGraph (std::string graphfile)
  Wczytywanie grafu z pliku.
- std::vector< std::string > split (std::string str)
  Dzielenie tekstu na fragmenty względem znaku.
- void saveGraph (std::vector< std::pair< std::vector< char >, int > > tsp, std::string filename)
  Zapis wyniku do pliku.

4.2.1 Opis szczegółowy

Plik nagłówkowy operacji na plikach.

Autor

Piotr Nowak (nowakpiotr510@gmail.com)

Wersja

0.1

Data

2023-02-06

Copyright

Copyright (c) 2023

4.2.2 Dokumentacja funkcji

4.2.2.1 loadGraph()

```
Graph loadGraph (
            std::string graphfile )
```

Wczytywanie grafu z pliku.

Parametry

routefile	
graphfile	

Zwraca

Graph

4.2.2.2 saveGraph()

```
void saveGraph (
            std::vector< std::pair< std::vector< char >, int > > tsp,
            std::string filename )
```

Zapis wyniku do pliku.

Parametry

tsp	
filename	

4.2.2.3 split()

```
std::vector< std::string > split (
            std::string str )
```

Dzielenie tekstu na fragmenty względem znaku.

Parametry

str	

Zwraca

std::vector<std::string>

4.3 File.h

Idź do dokumentacji tego pliku.

```
00001
00012 #include "TSP.h"
00013 #include <fstream>
```

```
00014 #include <string>
00015 #include <sstream>
00016 #include <iostream>
00017
00018 Graph loadGraph(std::string graphfile);
00019
00020 std::vector<std::string> split(std::string str);
00021
00022 void saveGraph(std::vector<std::pair<std::vector<char>, int» tsp, std::string filename);
```

4.4 Dokumentacja pliku Graph.cpp

Funkcje odpowiedzialne za graf.

```
#include "Graph.h"
```

Funkcje

- Graph initGraph (std::vector< char > cities)

  Inicjalizacja grafu.
- void addEdge (Graph &graph, char from, char to, int distance)

  Dodawanie krawędzi grafu (pojedyncze połączenia)

4.4.1 Opis szczegółowy

Funkcje odpowiedzialne za graf.

Autor

Piotr Nowak (nowakpiotr510@gmail.com)

Wersja

0.1

Data

2023-02-06

Copyright

Copyright (c) 2023

4.4.2 Dokumentacja funkcji

4.4.2.1 addEdge()

```
void addEdge (
            Graph & graph,
            char from,
            char to,
            int distance )
```

Dodawanie krawędzi grafu (pojedyncze połączenia)

Parametry

graph	
from	
to	
distance	

4.4.2.2 initGraph()

```
Graph initGraph (
            std::vector< char > cities )
```

Inicjalizacja grafu.

Parametry

index	
cities	

Zwraca

Graph

4.5 Dokumentacja pliku Graph.h

Plik nagłówkowy grafu.

```
#include <vector>
#include <map>
#include <iostream>
#include <algorithm>
```

Komponenty

- struct Graph

Funkcje

- Graph initGraph (std::vector< char > cities)

  Inicjalizacja grafu.
- void addEdge (Graph &graph, char from, char to, int distance)

  Dodawanie krawędzi grafu (pojedyncze połączenia)

4.5.1 Opis szczegółowy

Plik nagłówkowy grafu.

Autor

Piotr Nowak (nowakpiotr510@gmail.com)

Wersja

0.1

Data

2023-02-06

Copyright

Copyright (c) 2023

4.5.2 Dokumentacja funkcji

4.5.2.1 addEdge()

```
void addEdge (
            Graph & graph,
            char from,
            char to,
            int distance )
```

Dodawanie krawędzi grafu (pojedyncze połączenia)

Parametry

graph	
from	
to	
distance	

4.5.2.2 initGraph()

```
Graph initGraph (
            std::vector< char > cities )
```

Inicjalizacja grafu.

Parametry

index	
cities	

Zwraca

Graph

4.6 Graph.h

Idź do dokumentacji tego pliku.

```
00001
00011 #include <vector>
00012 #include <map>
00013 #include <iostream>
00014 #include <algorithm>
00015
00016 struct Graph{
00017     std::vector<char> cities;
00018     std::map<std::pair<char, char>, int> map;
00019 };
00020
00021 Graph initGraph(std::vector<char> cities);
00022
00023 void addEdge(Graph &graph, char from, char to, int distance);
```

4.7 Dokumentacja pliku main.cpp

Główny plik programu.

```
#include "File.h"
```

Funkcje

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

  Główna funkcja programu.

4.7.1 Opis szczegółowy

Główny plik programu.

Autor

Piotr Nowak (nowakpiotr510@gmail.com)

Wersja

0.1

Data

2023-02-06

Copyright

Copyright (c) 2023

4.7.2 Dokumentacja funkcji

4.7.2.1 main()

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

Główna funkcja programu.

Zwraca

int

4.8 Dokumentacja pliku TSP.cpp

Funkcje odpowiedzialne za rozwiązanie TSP.

```
#include "TSP.h"
```

Funkcje

- int calcFitness (std::vector< char > route, Graph graph)
  Funkcja obliczająca fitness danej trasy.
- std::vector< char > generateRoute (Graph graph)
  Funkcja generuje losowe ścieżki do populacji.
- std::vector< char > crossover (std::vector< char > &parent1, std::vector< char > &parent2)
  Funkcja odpowiadająca za krzyżowanie.
- std::pair< std::vector< char >, int > findBestRoute (std::vector< std::pair< std::vector< char >, int > > population)
  Funkcja znajdująca najlepszą trasę spośród tras w populacji.
- void mutate (std::vector< char > &route)
  Funkcja odpowiadająca za zachodzenie mutacji.
- std::vector< std::pair< std::vector< char >, int > > TSP (Graph graph, int population_size, int generations)
  Główna funkcja rozwiązująca TSP.

4.8.1 Opis szczegółowy

Funkcje odpowiedzialne za rozwiązanie TSP.

Autor

Piotr Nowak (nowakpiotr510@gmail.com)

Wersja

0.1

Data

2023-02-06

Copyright

Copyright (c) 2023

4.8.2 Dokumentacja funkcji

4.8.2.1 calcFitness()

```
int calcFitness (
            std::vector< char > route,
            Graph graph )
```

Funkcja obliczająca fitness danej trasy.

Parametry

route	
graph	

Zwraca

int

4.8.2.2 crossover()

```
std::vector< char > crossover (
            std::vector< char > & parent1,
            std::vector< char > & parent2 )
```

Funkcja odpowiadająca za krzyżowanie.

Parametry

parent1	
child1	
parent2	

Zwraca

std::vector<char>

4.8.2.3 findBestRoute()

```
std::pair< std::vector< char >, int > findBestRoute (
            std::vector< std::pair< std::vector< char >, int > > population )
```

Funkcja znajdująca najlepszą trasę spośród tras w populacji.

Parametry

population	

Zwraca

std::pair<std::vector<char>, int>

4.8.2.4 generateRoute()

```
std::vector< char > generateRoute (
            Graph graph )
```

Funkcja generuje losowe ścieżki do populacji.

Parametry

graph	

Zwraca

std::vector<char>

4.8.2.5 mutate()

```
void mutate (
            std::vector< char > & route )
```

Funkcja odpowiadająca za zachodzenie mutacji.

Parametry

route	

4.8.2.6 TSP()

```
std::vector< std::pair< std::vector< char >, int > > TSP (
            Graph graph,
            int population_size,
            int generations )
```

Główna funkcja rozwiązująca TSP.

Parametry

graph	
population_size	
generations	

Zwraca

std::vector<std::pair<std::vector<char>, int>>

4.9 Dokumentacja pliku TSP.h

Plik nagłówkowy TSP.

```
#include <vector>
#include <set>
#include "Graph.h"
```

Funkcje

- std::vector< std::pair< std::vector< char >, int > > TSP (Graph graph, int population_size, int generations)
  Główna funkcja rozwiązująca TSP.
- std::vector< char > crossover (std::vector< char > &parent1, std::vector< char > &parent2)
  Funkcja odpowiadająca za krzyżowanie.
- std::vector< char > generateRoute (Graph graph)
  Funkcja generuje losowe ścieżki do populacji.
- int calcFitness (std::vector< char > route, Graph graph)
  Funkcja obliczająca fitness danej trasy.
- void mutate (std::vector< char > &route)
  Funkcja odpowiadająca za zachodzenie mutacji.
- std::pair< std::vector< char >, int > findBestRoute (std::vector< std::pair< std::vector< char >, int > > population)
  Funkcja znajdująca najlepszą trasę spośród tras w populacji.

4.9.1 Opis szczegółowy

Plik nagłówkowy TSP.

Autor

Piotr Nowak (nowakpiotr510@gmail.com)

Wersja

0.1

Data

2023-02-06

Copyright

Copyright (c) 2023

4.9.2 Dokumentacja funkcji

4.9.2.1 calcFitness()

```
int calcFitness (
            std::vector< char > route,
            Graph graph )
```

Funkcja obliczająca fitness danej trasy.

Parametry

route	
graph	

Zwraca

int

4.9.2.2 crossover()

```
std::vector< char > crossover (
            std::vector< char > & parent1,
            std::vector< char > & parent2 )
```

Funkcja odpowiadająca za krzyżowanie.

Parametry

parent1	
child1	
parent2	

Zwraca

std::vector<char>

4.9.2.3 findBestRoute()

```
std::pair< std::vector< char >, int > findBestRoute (
            std::vector< std::pair< std::vector< char >, int > > population )
```

Funkcja znajdująca najlepszą trasę spośród tras w populacji.

Parametry

population	

Zwraca

std::pair<std::vector<char>, int>

4.9.2.4 generateRoute()

```
std::vector< char > generateRoute (
            Graph graph )
```

Funkcja generuje losowe ścieżki do populacji.

Parametry

graph	

Zwraca

std::vector<char>

4.9.2.5 mutate()

```
void mutate (
            std::vector< char > & route )
```

Funkcja odpowiadająca za zachodzenie mutacji.

Parametry

route	

4.9.2.6 TSP()

```
std::vector< std::pair< std::vector< char >, int > > TSP (
            Graph graph,
            int population_size,
            int generations )
```

Główna funkcja rozwiązująca TSP.

Parametry

graph	
population_size	
generations	

Zwraca

std::vector<std::pair<std::vector<char>, int>>

4.10 TSP.h

Idź do dokumentacji tego pliku.

```
00001
00012 #include <vector>
00013 #include <set>
00014 #include "Graph.h"
00015
00016 std::vector<std::pair<std::vector<char>, int» TSP(Graph graph, int population_size, int generations);
00017
00018 std::vector<char> crossover(std::vector<char> &parent1, std::vector<char> &parent2);
00019
00020 std::vector<char> generateRoute(Graph graph);
00021
00022 int calcFitness(std::vector<char> route, Graph graph);
00023
00024 void mutate(std::vector<char> &route);
00025
00026 std::pair<std::vector<char>, int> findBestRoute(std::vector<std::pair<std::vector<char>, int»
      population);
```

Skorowidz