[PSZT] Problem komiwojażera dokumentacja

1. **Autorzy**:
   -Rafał Lewanczyk
   -Kacper Biegajski
   https://github.com/rafallewanczyk/Komiwojazer
2. **Temat**:
   Rozwiązać problem poszukiwania najkrótszej ścieżki Hamiltona w grafie nieskierowanym przy pomocy algorytmu A*. Porównać działanie algorytmu A* z brutalnym podejściem
3. **Założenia**:
   -Grafy wejściowe ze strony http://sndlib.zib.de/home.action
   -W celu implementacji oraz reprezentacji grafów została użyta zewnętrzna biblioteka *networkx*
4. **Opis funkcjonalności**:
   -klasa *Astar* zawiera implementacje algorytmu A* oraz dwóch różnych funkcji heurystycznych. Zwraca tuple (*ścieżka*, *koszt*, *liczba odwiedzonych stanów*), w przypadku braku ścieżki *ścieżka = [], koszt = inf*
   -klasa *BruteForce* zawiera implementacje algorytmu brutalnego. Zwraca wartości w tym samym formacie co *Astar*
   -klasa *GraphDrawer* odpowiada za rysowanie grafów oraz znalezionej ścieżki
   -klasa *GraphManager* odpowiada za wczytywanie grafów z pliku
5. **Uruchomienie programu**:
   Przy starcie programu musimy podać szereg informacji:
   -scieżka do pilku z wierzchołkami (obowiązkowo)
   -scieżka do pliku z połączeniami (obowiązkowo)
   -wierzchołek początkowy (obowiązkowo)
   Flagi:
   -a - rozwiązuje problem przy wykorzystaniu algorytmu a*
   -b - rozwiązuje problem przy użyciu algorytmu brutalnego
   -w nazwa_pliku - zapisuje rezultaty do pliku
   -d algorytm - rysuje graf ze znaleziona scieżką z danego algorytmu (a - a*, b - brutalny)
   -f funkcja - wybór heurystyki do algorytmu a* (z - *zero* [domyślna], g- *greedy*, s - *shortest_from_all*)
   -h - informacja o dostępnych flagach

6. **Rozwiązanie**:
   Rozwiązanie problemu komiwojażera przy pomocy algorytmu A* opiera się na mądrym generowaniu oraz przeszukiwaniu przestrzeni poszukiwań. Przestrzeń poszukiwań jest drzewem w którym każdy z wierzchołków przypisany ma pewną unikalną ścieżkę (permutację miast), koszt przejścia danej ścieżki oraz oszacowany przez funkcję heurystyczną koszt odwiedzenia pozostałych wierzchołków. Przestrzeń przeszukiwań generowana jest do momentu znalezienia pierwszej ścieżki

zawierającej wszystkie wierzchołki tańszej od nieodwiedzonych wygenerowanych stanów.

Zaimplementowaliśmy trzy funkcje heurystyczne:
-*shortest_from_all_heuristic* : funkcja przybliża koszt uzyskania n-elementowej ścieżki poprzez pomnożenie najtańszej nieużywanej krawędzi w grafie razy pozostałą potrzebną do uzyskania pełnej ścieżki liczbę wierzchołków

-*greedy_heuristic*: funkcja przybliża koszt uzyskania n-elementowej ścieżki poprzez pomnożenie najtańszej krawędzi wychodzącej z danego wierzchołka razy pozostałą potrzebną do uzyskania pełnej ścieżki liczbę wierzchołków. Funkcja ta nie spełnia warunku monotoniczności, przez co nie zawsze daje poprawny wynik, ale często znacznie przyśpiesza działanie algorytmu

-funkcja heurystyczna zwracająca zawsze zero. Uzyskujemy algorytm UCS (uniform cost search)

7. **Rezultaty**:

POLAND (12 wierzchołków, 18 połączeń, start w “Gdansk”)

Algorytm	**czas wykonania [s]**	**Odwiedzone stany**	**długość trasy**
Brute force	0.000997	408	22.477201
UCS	0.004985	395	22.477201
Greedy	0.003988	212	24.105720
Shortest from all	0.006981	343	22.477201

NORWAY (27 wierzchołków, 51 połączeń, start w “N20”)

Algorytm	**czas wykonania [s]**	**Odwiedzone stany**	**długość trasy**
Brute force	11.159687	2908075	3369.914569
UCS	73.289040	2476648	3369.914569
Greedy	0.132674	7609	3390.040272
Shortest from all	33.535063	1363428	3369.914569

FRANCE (25 wierzchołków, 45 połączeń, start w “N20”)

Algorytm	czas wykonania [s]	Odwiedzone stany	długość trasy
Brute force	0.168543	52595	inf
UCS	0.803188	52595	inf
Greedy	1.051533	52595	inf
Shortest from all	1.059732	52595	inf

Graf FRANCE nie zawierał ścieżki Hamiltona zaczynającej się na wierzchołku “N20”

8. Wnioski:

Algorytm brutalny daje lepsze wyniki pomimo konieczności odwiedzenia znacznie większej liczby stanów. Jest to spowodowane tym, że algorytm A* jest spowolniony przez konieczność utrzymywania kolejki priorytetowej oraz obliczenie funkcji heurystycznej dla każdego ze stanów.