Problem komiwojażera

Znalezienie minimalnego cyklu Hamiltona w pełnym grafie ważonym.

Objaśnienie definicji

Cykl Hamiltona - to taki cykl w grafie, w którym każdy wierzchołek grafu odwiedzany jest dokładnie raz (oprócz pierwszego wierzchołka, który odwiedzany jest 2 razy). *Nazwa cyklu i ścieżki pochodzi od irlandzkiego matematyka Hamiltona.*

Graf ważony - to taki graf, do którego **krawędzi**, lub **wierzchołków** można przypisać etykiety (**wagi**).

Pochodzenie nazwy

Nazwa pochodzi od typowej ilustracji problemu, przedstawiającej go z punktu widzenia wędrownego sprzedawcy (**komiwojażera**): dane jest **n miast**, które komiwojażer ma **odwiedzić**, oraz **odległość / cena podróży / czas podróży** pomiędzy każdą parą miast. Celem jest znalezienie **najkrótszej / najtańszej / najszybszej** drogi łączącej wszystkie miasta, zaczynającej się i kończącej się w określonym punkcie.

Komiwojażer symetryczny i asymetryczny

Symetryczny problem komiwojażera (STSP) polega na tym, że dla dowolnych miast A i B odległość z A do B jest taka sama jak z B do A.

Asymetrycznym problem komiwojażera (ATSP) polega na tym, że dla dowolnych miast A i B odległość z A do B różni się od odległości z B do A.

**W projekcie zastosowano ATSP*

Historia

W 1859 irlandzki matematyk William Rowan Hamilton sformułował problem istnienia cyklu o długości *n* w grafie *n*-wierzchołkowym.
Za pierwszego autora, który sformalizował matematycznie problem komiwojażera uznaje się austriackiego matematyka Karla Mengera, który zdefiniował go w 1930 zwracając szczególną uwagę na trudność w obliczeniu rozwiązania. Niezależnie od niego ten sam problem poruszył w 1934 Hassler Witney na wykładzie w Princeton University. Natomiast pierwsza próba rozwiązania problemu miała miejsce w 1937, gdy Merrill Flood pracował nad rozwiązaniem wyznaczania tras dla autobusów szkolnych.

Z uwagi na bardzo prosty opis problemu oraz opinię o bardzo trudnym obliczeniowo procesie optymalizacji, problem komiwojażera stał się bardzo popularny. Fascynacja ta trwa od lat pięćdziesiątych XX wieku do dziś, zarówno wśród amatorów jak i profesjonalistów.

Lista przykładowych algorytmów

- Brute-force (dokładny)
- Helda Karpa (dokładny)
- Forda Fulkersona (algorytm heurystyczny)
- Nearest Neighbor Algorithm - **Algorytm Najbliższego Sąsiada** (algorytm heurystyczny)
- Smallest Edge Algorithm - **Algorytm Najmniejszej Krawędzi** (algorytm heurystyczny)

**W projekcie zastosowano Algorytm Najbliższego Sąsiada i Algorytm Najmniejszej Krawędzi (oba algorytmy są algorytmami zachłannymi)*

Przykładowy opis algorytmu najbliższego sąsiada

Algorytm rozpoczyna działanie od wybranego wierzchołka i polega na kolejnym przechodzeniu do najbliższego nie odwiedzonego sąsiada ostatnio dodanego wierzchołka.

W bardziej formalnym zapisie algorytm działa w następujący sposób:

1. Wierzchołek początkowy oznaczamy jako odwiedzony i ustawiamy jako aktualny.
2. Znajdujemy najkrótszą spośród krawędzi łączących aktualny wierzchołek z jeszcze nie odwiedzonymi wierzchołkami.
3. Dołączamy do rozwiązania krawędź znalezioną w punkcie 2.
4. Wierzchołek będący *drugim końcem* krawędzi znalezionej w punkcie 2 oznaczamy jako odwiedzony i ustawiamy jako aktualny.
5. Jeśli są jeszcze nieodwiedzone wierzchołki, przechodzimy do punktu 2.
6. Dołączamy krawędź łączącą ostatnio dodany wierzchołek z wierzchołkiem początkowym. Zamykamy w ten sposób cykl.

http://algorytmy.ency.pl/artykul/algorytm_najblizszego_sasiada

Przykładowy opis algorytmu najmniejszej krawędzi

Algorytm polega na kolejnym dołączaniu do rozwiązania najkrótszych spośród dopuszczalnych krawędzi.

Działanie algorytmu można zapisać następująco:

1. Posortuj wszystkie krawędzie rosnąco według ich wag, umieść je w kolejce.
2. Pobierz z kolejki krawędź o najmniejszej wadze, usuń ją z kolejki.
3. Sprawdź, czy dołączenie tej krawędzi do rozwiązania nie spowoduje utworzenia cyklu (nie dotyczy ostatniej iteracji) lub powstania wierzchołka, z którego wychodzą trzy krawędzie. Jeśli nie, dołącz krawędź do rozwiązania.
4. Jeśli liczba krawędzi dołączonych do rozwiązania jest równa liczbie wierzchołków, zakończ działanie algorytmu. W przeciwnym razie przejdź do punktu 2.

http://algorytmy.ency.pl/artykul/algorytm_najmniejszej_krawedzi

Przebieg testów

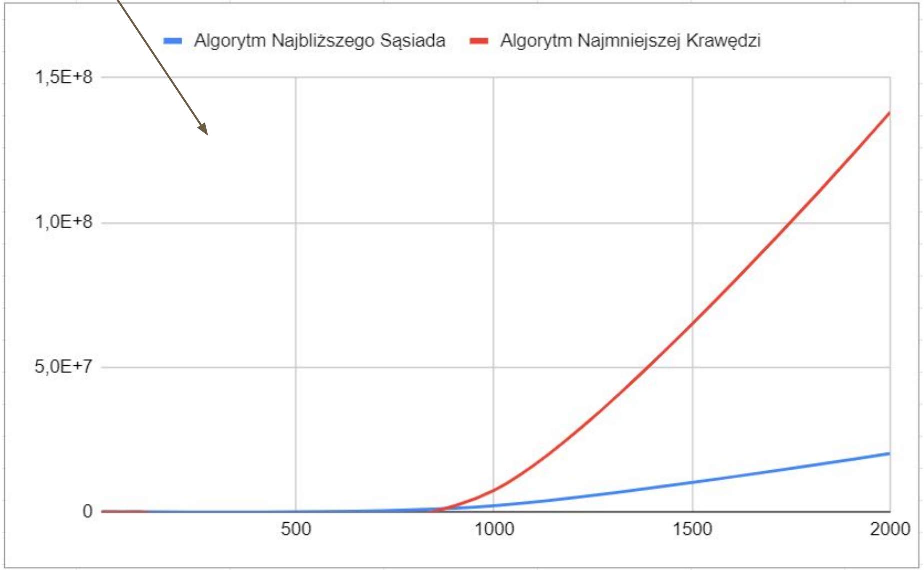
Nazwa algorytmu	Test dla 10 miast	Test dla 100 miast	Test dla 200 miast	Test dla 500 miast	Test dla 800 miast	Test dla 1000 miast	Test dla 2000 miast
Algorytm Najbliższego Sąsiada	6μs	2555μs	18045μs	343943μs	1026963μs (~1 sekunda)	2088392μs (~2 sekundy)	20166543μs (~20 sekund)
Algorytm Najmniejszej Krawędzi	13μs	5307μs	40876μs	745863μs	3023560μs (~3 sekundy)	7419437μs (~7,5 sekundy)	138106566μs (~2 min. i 18 sekund)

Przykład działania - wypisanie informacji w konsoli

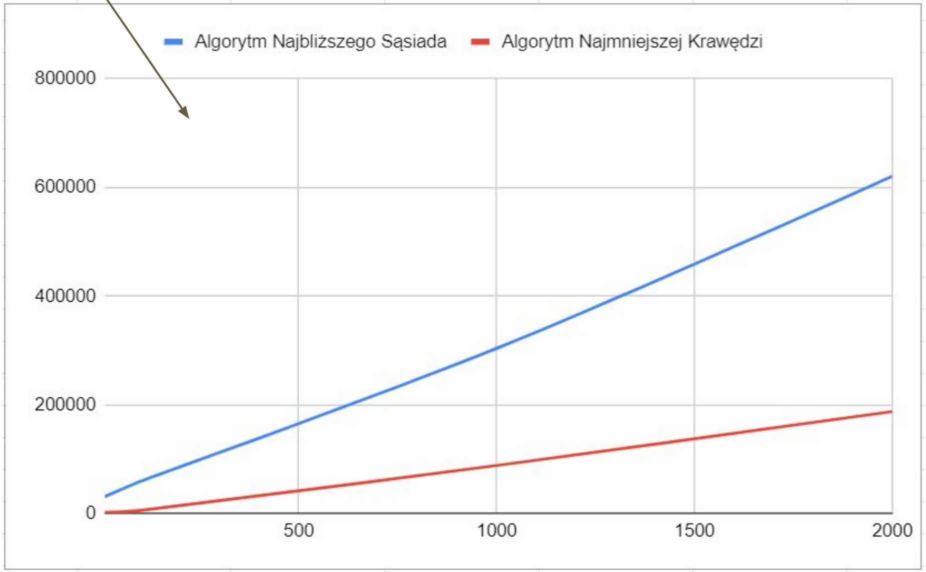
```
Podaj ilosc miast -  10

Oto odleglosci miedzy miastami:
10000 73 34 67 25 47 19 32 67 49
67 10000 76 42 59 72 94 71 57 94
45 90 10000 37 29 77 54 25 31 77
31 25 27 10000 83 91 7 9 28 22
1 5 74 85 10000 47 93 79 18 86
37 40 14 60 67 10000 22 27 53 27
67 35 77 8 21 12 10000 67 76 30
93 68 1 60 92 85 92 10000 55 86
39 34 30 81 56 24 99 29 10000 70
4 66 5 73 26 45 14 50 44 10000

Czas wykonania algorytmu najblizszego sasiada: 6microseconds
Czas wykonania algorytmu najmniejszej krawedzi: 13microseconds

Oto najszybsza droga od 1 do ostatniego miasta rozwiazana za pomoca algorytmu najblizszego sasiada
Kolejnosc miast: 0 -> 6 -> 3 -> 7 -> 2 -> 4 -> 1 -> 8 -> 5 -> 0

Oto najszybsza droga od 1 do ostatniego miasta rozwiazana za pomoca algorytmu najmniejszej krawedzi
Kolejnosc tworzenia drog miedzy miastami(pokazane sa wartosci krawedzi): 1 1 4 5 5 7 9 12 24 34
```

Wykresy

wykres - czas trwania algorytmu



wykres - długość przebytej drogi



Wnioski

W obu algorytmach dla testowanych wielkości(wielkość grafu): 10, 100, 200, 300, 500, 800, 1000, 2000 czas wykonania, oraz długość drogi znacząco się różnią. Algorytm najbliższego sąsiada dla małych wartości wykonuje się 2 razy szybciej od algorytmu najmniejszej krawędzi, jednak długość drogi jest znacząco większa w porównaniu z długością drogi obliczoną za pomocą algorytmu najmniejszej krawędzi. Różnica czasu wykonania algorytmów, oraz długości przebytej drogi zwiększa się wraz ze wzrostem wartości. Algorytm najbliższego sąsiada jest algorytmem szybszym jednak mniej wydajnym. Algorytm najmniejszej krawędzi jest algorytmem wolniejszym, lecz bardziej dokładnym.