

numer indeksu	imię i nazwisko
259190	Jędrzej Czykier

(1) Brute force

Spis treści	strona
1. Sformułowanie zadania	2
2. Opis metody	3
3. Opis algorytmu	4
4. Dane testowe	5
5. Procedura badawcza	6
6. Wyniki	7
7. Analiza wyników i wnioski	8

1. Sformułowanie zadania

Zadanie polega na opracowaniu, implementacji i zbadaniu efektywności algorytmu przeglądu zupełnego rozwiązującego problem komiwożera w wersji optymalizacyjnej. Nazywany również metodą „Brute force”, algorytm ten sprawdza po kolei wszystkie możliwości co czyni go relatywnie prostym do implementacji lecz powolnym w działaniu.

Problem komiwożera, nazywany też TSP (*eng. Traveling Salesman Problem*), polega na znalezieniu minimalnego cyklu Hamiltona czyli takiego w którym każdy wierzchołek grafu odwiedzany jest tylko raz, z wyjątkiem pierwszego do którego algorytm powinien powrócić na końcu działania.

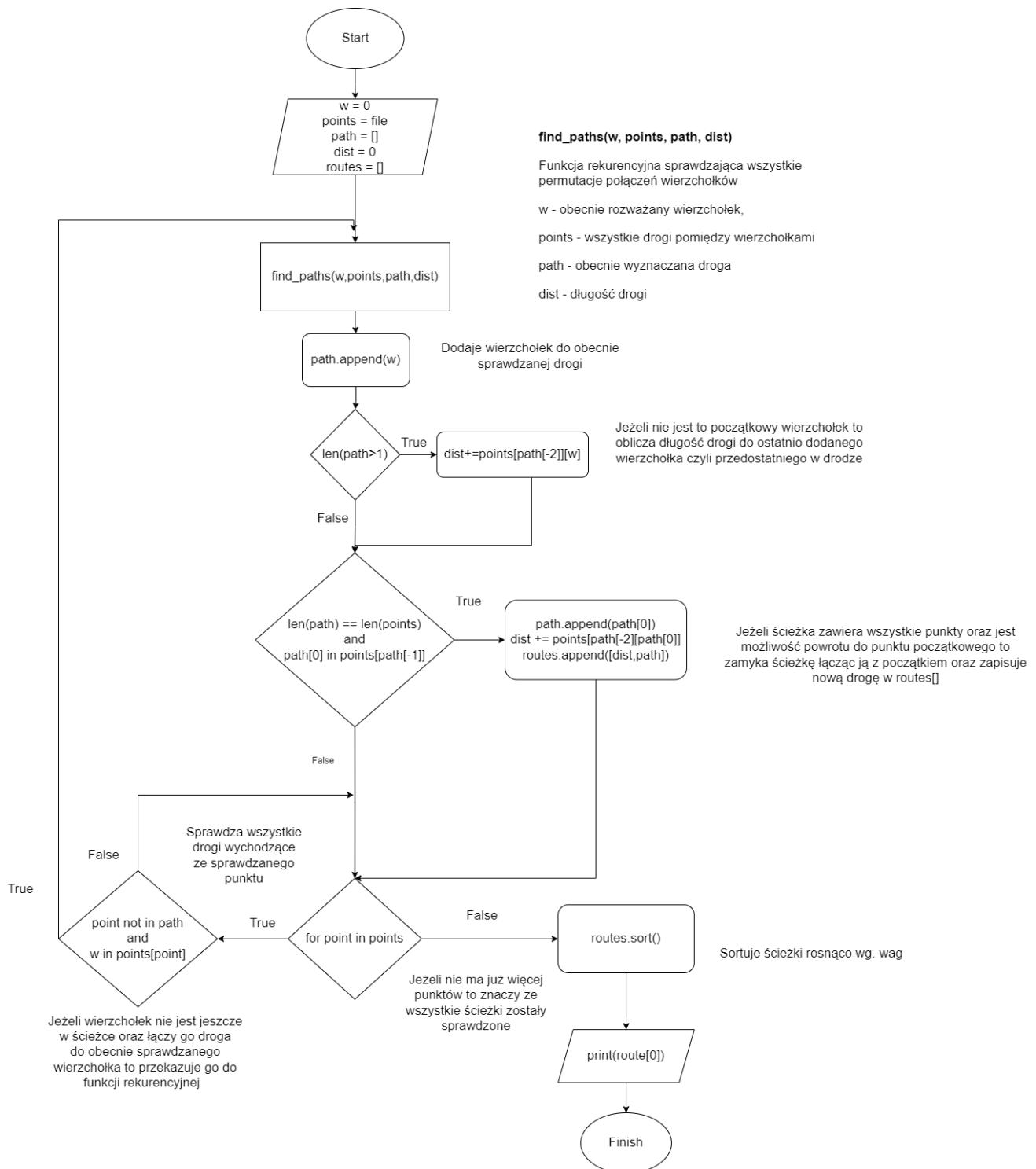
2. Metoda

Metoda przeglądu zupełnego, tzw. Przeszukiwanie wyczerpujące (*eng. Exhaustive search*), bądź metoda siłowa (*eng. Brute force*), polega na znalezieniu i sprawdzeniu wszystkich rozwiązań dopuszczalnych problemu, wyliczeniu dla nich wartości funkcji celu i wyborze rozwiązania o ekstremalnej wartości funkcji celu – najniżej (problem minimalizujący) bądź najwyżej (problem maksymalizujący). Takie algorytmy są z reguły nie optymalne ale jednocześnie najprostsze w implementacji. Dla niektórych problemów jest to jedyne niezawodne podejście. W teoretycznym przypadku, dysponując dowolnie długim czasem oraz bez limitacji sprzętowych, algorytm brute force zawsze zakończyłby się powodzeniem.

Rozważając problem komiwojażera, zastosowanie przeszukiwania wyczerpującego polega na sprawdzeniu wszystkich możliwych ścieżek spełniających kryterium problemu (muszą być cyklem Hamiltona) oraz wybraniu ścieżki z najmniejszą wagą krawędzi.

3. Algorytm

Przedstawienie algorytmu za pomocą schematu blokowego:



4. Dane testowe

Dane wykorzystane do sprawdzenia poprawności algorytmu:

1. tsp_6_1.txt,
2. tsp_6_2.txt,
3. tsp_10.txt

Dane wykorzystane do badań:

1. tsp_6_1.txt,
2. tsp_6_2.txt,
3. tsp_10.txt,
4. tsp_12.txt,
5. tsp_13.txt,
6. tsp_14.txt

<http://jaroslaw.mierzwa.staff.iiar.pwr.wroc.pl/pea-stud/tsp/>

5. Procedura badawcza

Należało zbadać zależność czasu rozwiązania problemu od wielkości instancji. W przypadku algorytmu realizującego przegląd zupełny przestrzeni rozwiązań dopuszczalnych nie występowały parametry programu, które mogły mieć wpływ na czas i jakość uzyskanego wyniku. W związku z tym procedura badawcza polegała na uruchomieniu programu sterowanego plikiem inicjującym .INI

Treść pliku .ini:

```
[section_a]
```

```
file0 = tsp_6_1.txt 20
```

```
file1 = tsp_6_2.txt 20
```

```
file2 = tsp_10.txt 10
```

```
file3 = tsp_12.txt 5
```

```
file4 = tsp_13.txt 2
```

```
file5 = tsp_14.txt 1
```

```
outputFile = outputBruteForce.csv
```

Każda z instancji rozwiązywana była zgodnie z liczbą jej wykonan, np. tsp_6_1.txt wykonana została 20 razy. Do pliku wyjściowego outputBruteForce.csv zapisywany był czas wykonania, otrzymane rozwiązanie (koszt ścieżki) oraz ścieżka (numery kolejnych węzłów). Plik wyjściowy zapisywany był w formacie csv. Poniżej przedstawiono fragment zawartości pliku wyjściowego.

```
tsp_6_1.txt,0.0,132,"['0', '1', '2', '3', '4', '5', '0']"
```

```
tsp_6_1.txt,0.0,132,"['0', '1', '2', '3', '4', '5', '0']"
```

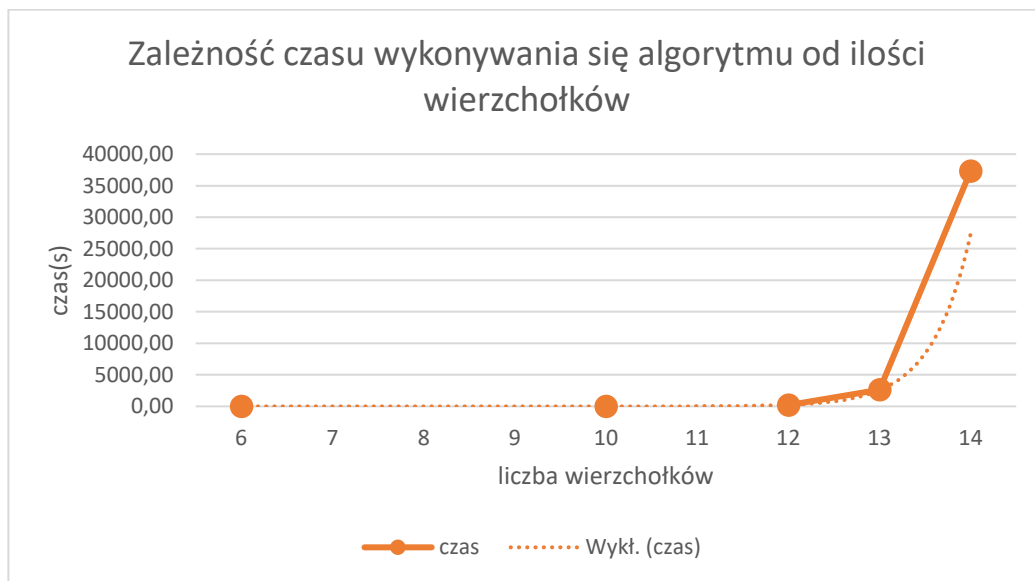
```
tsp_6_1.txt,0.0,132,"['0', '1', '2', '3', '4', '5', '0']"
```

```
...
```

```
tsp_14.txt,37321.246992123453,282,"['0', '10', '3', '5', '7', '9', '13', '11', '2', '6', '4', '8', '1', '12', '0']"
```

6. Wyniki

Wyniki zgromadzone zostały w plikach: outputBruteForce.csv. Wszystkie ww. pliku zostały dołączone do raportu i znajdują się na dysku Google pod adresem <https://drive.google.com/drive/folders/1JwNuyB3QtiVINktmGQQhSeyfqd2DFPET?usp=sharing>. Wyniki przedstawione zostały w postaci wykresu zależności czasu uzyskania rozwiązania problemu od wielkości instancji (rysunek 1)



Rys. 1

7. Analiza wyników i wnioski

Krzywa wzrostu czasu względem wielkości instancji ma charakter wykładniczy (rysunek 1). Nałożenie krzywej $O(n!)$ potwierdza, że badany algorytm wyznacza rozwiązania problemu komiwojażera dla badanych instancji w czasie $n!$ zależnym względem wielkości instancji (obie krzywe są zgodne co do kształtu). Złożoność czasowa opracowanego algorytmu wynosi $O(n!)$.