

259113 Hubert Bełkot

Brute Force(1)

<i>Spis treści</i>	<i>strona</i>
<i>Sformułowanie zadania</i>	<i>2</i>
<i>Metoda</i>	<i>3</i>
<i>Algorytm</i>	<i>4</i>
<i>Dane testowe</i>	<i>6</i>
<i>Procedura badawcza</i>	<i>7</i>
<i>Wyniki</i>	<i>8</i>
<i>Analiza wyników</i>	<i>9</i>

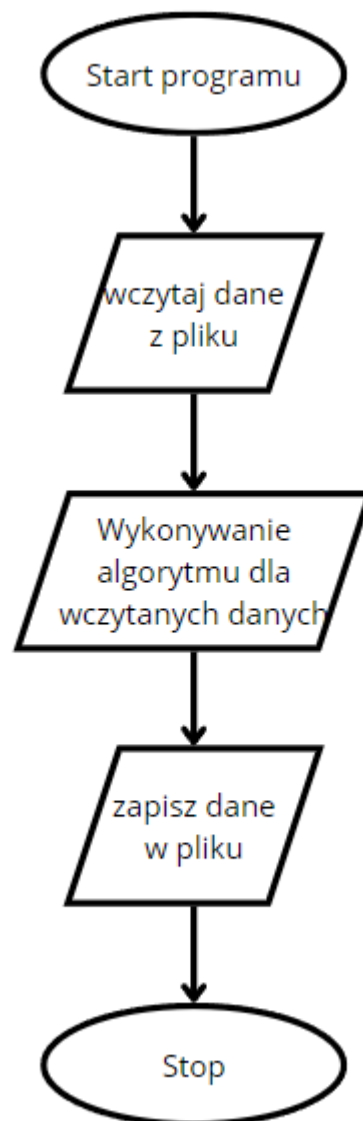
1. Sformułowanie Zadania

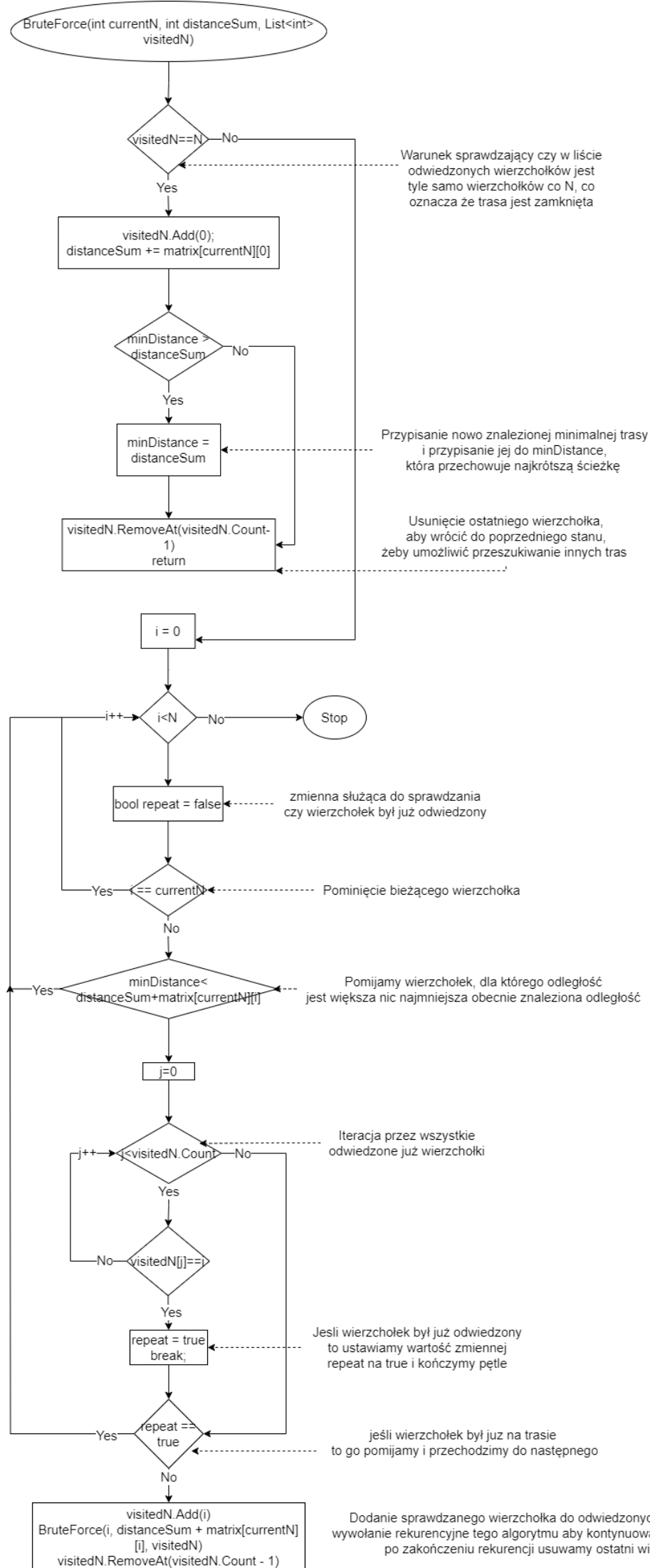
Zadanie polegało na opracowaniu algorytmu i zbadaniu efektywności algorytmu przeglądu zupełnego rozwiązującego problem komiwojażera. Problem komiwojażera (ang TSP – Travelling Salesman Problem), czyli problem wędrującego handlarza, polega na odwiedzeniu wszystkich wierzchołków grafu, które są reprezentacją miast. Rozwiązanie polega na znalezieniu najbardziej optymalnej ścieżki odwiedzającej wszystkie wierzchołki. Trasa jest cyklem przechodzącym przez wierzchołek grafu dokładnie jeden raz, jest to cykl Hamiltona.

2. Metoda

Metoda polega na przeanalizowaniu wszystkich możliwych rozwiązań, aby wybrać to najlepsze, najkorzystniejsze, które spełnia warunki zadania. Metoda przeglądu zupełnego to tak zwana metoda siłowa (ang. brute force) lub wyczerpująca(ang. Exhaustive search). Minusem takiej metody jest duża złożoność obliczeniowa algorytmów, które ją realizują. Dzięki niej jesteśmy w stanie znaleźć optymalne rozwiązanie, kosztem zasobów.

3. Algorytm





4. Dane testowe

Do sprawdzenia poprawności działania algorytmu posłużyły następujące zestawy danych:

- tsp_6_1.txt
- tsp_10.txt

Do wykonania pomiarów wykorzystano:

- tsp_6_1.txt
- tsp_6_2.txt
- tsp_10.txt
- tsp_12.txt
- tsp_13.txt
- tsp_14.txt
- tsp_15.txt

Ze strony: <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 tsp.INI (nazwa_instancji, liczba_wykonan, rozwiązanie_optymalne, [ścieżka optymalna]; nazawa_pliku_wyjściowego.csv)

Treść pliku:

tsp_6_1.txt 100 132 [0 1 2 3 4 5 0]

tsp_6_2.txt 100 80 [0 5 1 2 3 4 0]

tsp_10.txt 100 212 [0 3 4 2 8 7 6 9 1 5 0]

tsp_12.txt 100 264 [0 1 8 4 6 2 11 9 7 5 3 10 0]

tsp_13.txt 50 269 [0 10 3 5 7 9 11 2 6 4 8 1 12 0]

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

tsp_15.txt 5 291 [0 10 3 5 7 9 13 11 2 6 4 8 14 1 12 0]

outputBF.csv

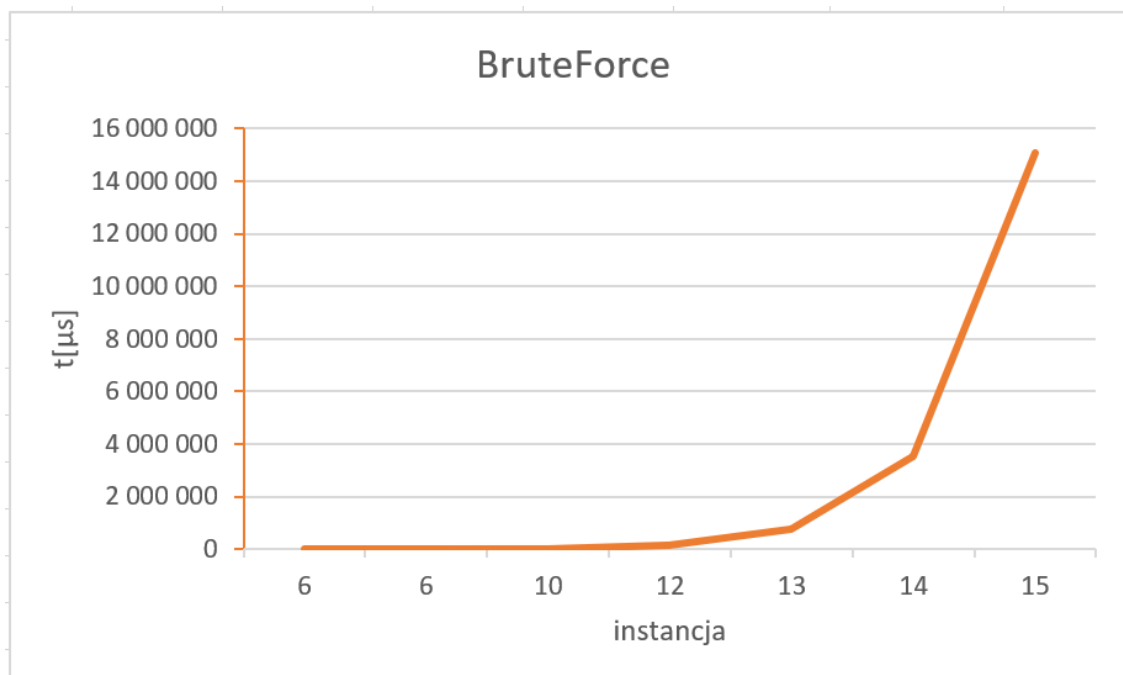
W pliku wyjściowym zapisywany był: jako nagłówek dane pobierane z pliku.INI, a następnie liczba wykonań, czas wykonania(ms), otrzymane rozwiązanie oraz ścieżka. Plik wyjściowy zapisywany jest w formacie csv.

Fragment pliku wynikowego outputBF.csv (Wyniki opracowane w programie MS Excel):

tsp_10.txt	100	212 [0 3 4 2 8 7 6 9 1 5 0]
3772,9	212	[0 3 4 2 8 7 6 9 1 5 0]
3752,5	212	[0 3 4 2 8 7 6 9 1 5 0]
3734	212	[0 3 4 2 8 7 6 9 1 5 0]
3744	212	[0 3 4 2 8 7 6 9 1 5 0]
3754,6	212	[0 3 4 2 8 7 6 9 1 5 0]
3722,6	212	[0 3 4 2 8 7 6 9 1 5 0]
3745	212	[0 3 4 2 8 7 6 9 1 5 0]
3738,6	212	[0 3 4 2 8 7 6 9 1 5 0]
3754,3	212	[0 3 4 2 8 7 6 9 1 5 0]
3731,2	212	[0 3 4 2 8 7 6 9 1 5 0]
3743,9	212	[0 3 4 2 8 7 6 9 1 5 0]

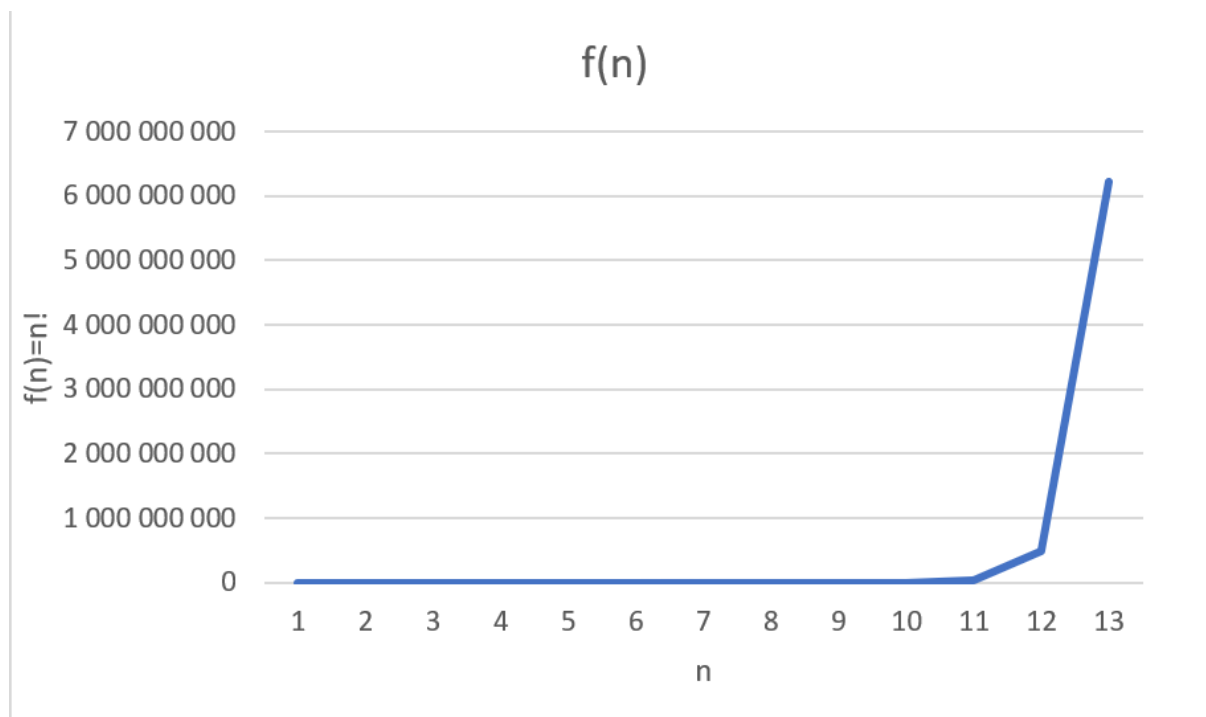
6. Wyniki

Wykres zależności czasu od działania algorytmu



Wykres1: czas działania algorytmu dla danych testowych.

Uśredniony czas jest wyliczony ze wzoru:
(suma czasu z każdego wykonania konkretnego graf)/(ilość wykonań).



Wykres2: wykres pokazujący funkcję $f(n)=n!$.

7. Analiza wyników

Porównując wyniki krzywej(Wykres1) średniego wykonywania się algorytmu Brute Force'a dla kolejnych danych testowych, można zauważyć, że owa krzywa ma tendencje wykładniczą. Dla porównania poniżej przedstawiamy krzywą funkcji $f(n)=n!$ (Wykres2), który potwierdza, że Wykres1 jest wykładniczy. Tym samym możemy potwierdzić, że nasz algorytm ma złożoność obliczeniową $O(n!)$.