

# Projektowanie efektywnych algorytmów

## Projekt

**263934 Michał Pawlus**

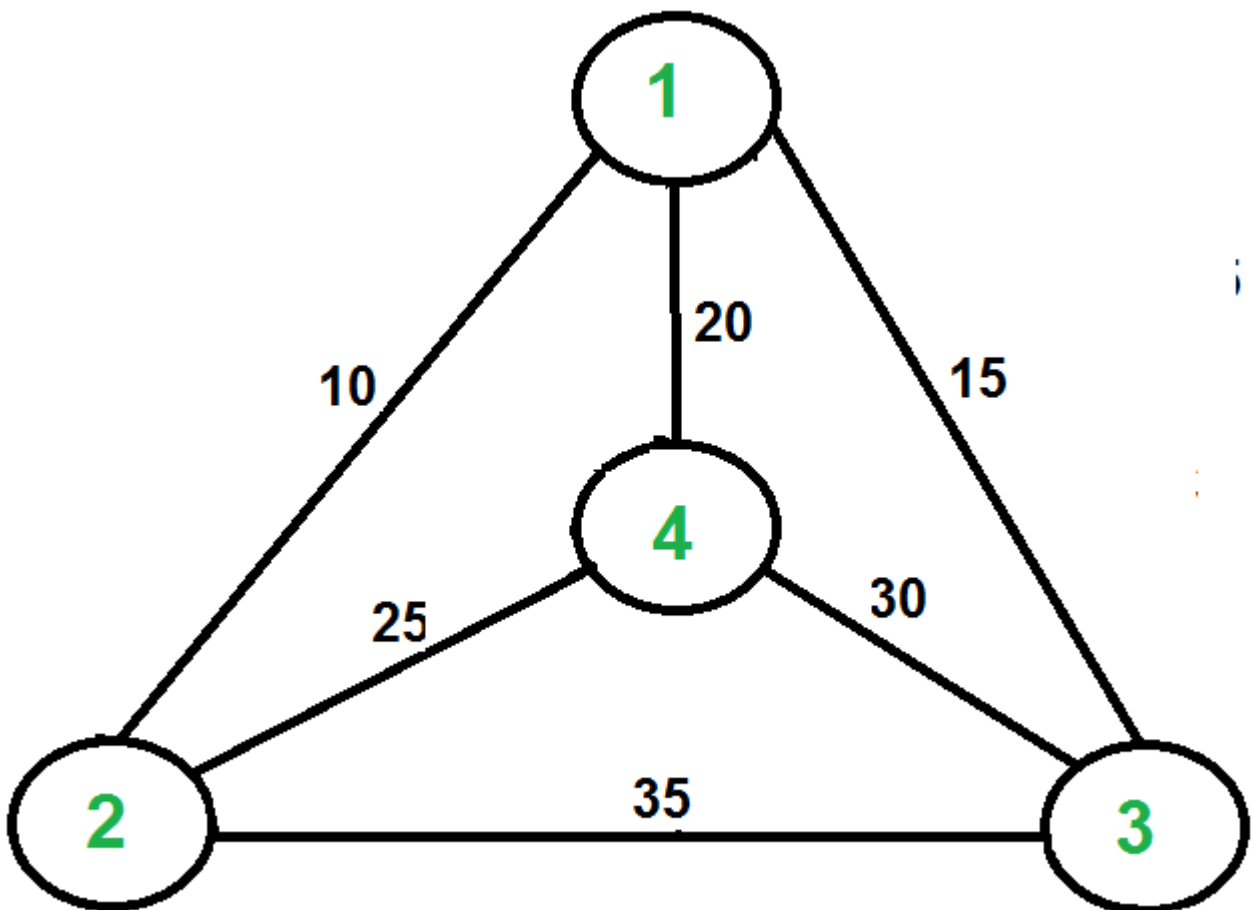
## Brute force

[illegible]

## 1. Sformułowanie zadania

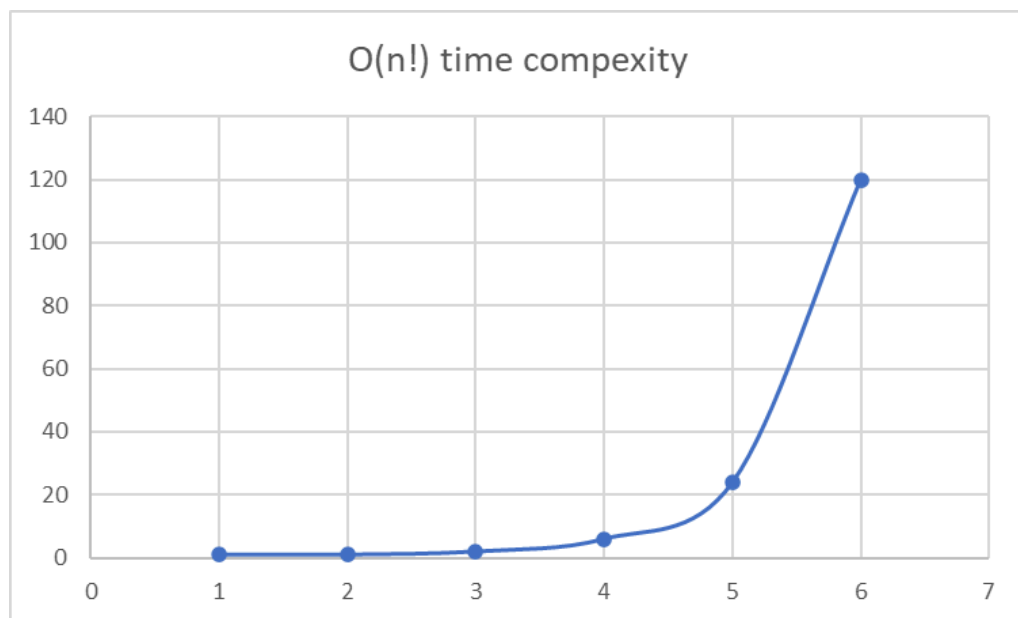
Zadanie polega na opracowaniu, implementacji i zbadaniu efektywności algorytmu przeglądu zupełnego rozwiązującego problem komiwojażera w wersji optymalizacyjnej. Problem komiwojażera Polega na znalezieniu najkrótszej zamkniętej trasy, która odwiedza wszystkie zadane punkty (miasta) dokładnie raz i wraca do punktu początkowego. Taka trasa nazywana jest również ścieżką Hamiltona

**Przykładowa instancja problemu(najkrótsza ścieżka 1-2-4-3-1 ma koszt 80)**

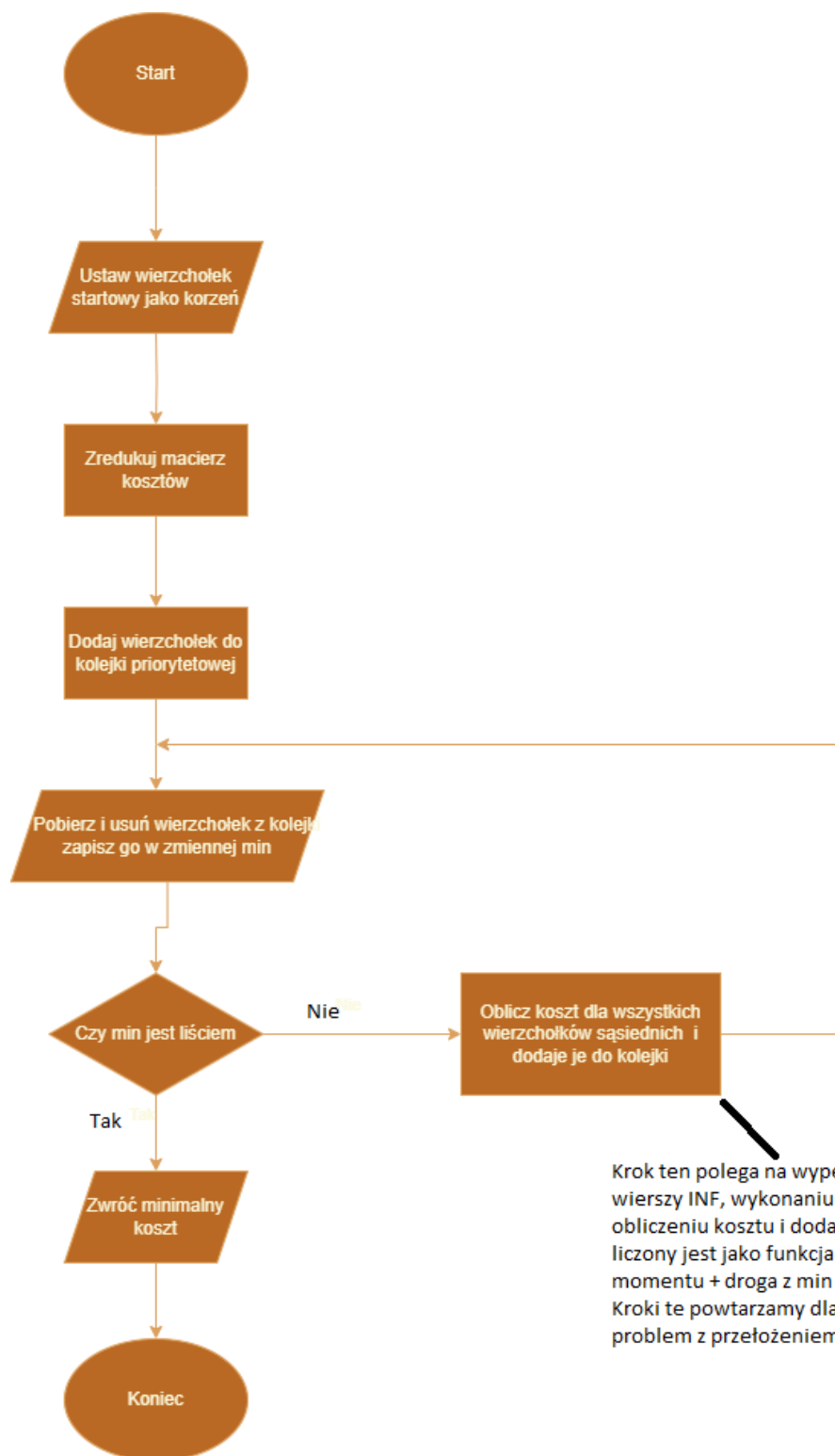


## 2. Metoda

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 – najniższej (problem minimalizacyjny) bądź najwyższej (problem maksymalizacyjny). Złożoność obliczeniowa algorytmu brute force wynosi  $O(n!)$ , gdzie "n" to liczba wierzchołków grafu. Ze względu na swoją wykładniczą złożoność, algorytm brute force jest praktycznie stosowany tylko do rozwiązywania bardzo małych instancji problemu TSP (zwykle do około 10-15 wierzchołków).



## 3. Algorytm



Text

Krok ten polega na wypełnieniu odpowiednich kolumn i wierszy INF, wykonaniu redukcji macierzy kosztów, obliczeniu kosztu i dodaniu wierzchołka do kolejki. Koszt liczony jest jako funkcja  $y = \text{przebyta droga do tego momentu} + \text{droga z min do wierzchołka} + \text{koszt z redukcji}$ . Kroki te powtarzamy dla każdego z sąsiadów. Miałem problem z przełożeniem tego na schemat

## 4. Dane testowe

Do sprawdzenia poprawności działania algorytmu wybrano następujące dane testowe:

1. test\_6\_1.csv; [http://jaroslaw.mierzwa.staff.iiar.pwr.wroc.pl/pea-stud/tsp/tsp\\_6\\_1.txt](http://jaroslaw.mierzwa.staff.iiar.pwr.wroc.pl/pea-stud/tsp/tsp_6_1.txt) (dane te same tylko rozszerzenie i separator zmieniony)
2. test\_10.csv; [http://jaroslaw.mierzwa.staff.iiar.pwr.wroc.pl/pea-stud/tsp/tsp\\_10.txt](http://jaroslaw.mierzwa.staff.iiar.pwr.wroc.pl/pea-stud/tsp/tsp_10.txt) (dane te same tylko rozszerzenie i separator zmieniony)

Do wykonania badań wybrano następujący zestaw instancji (dane o nazwie graph.csv są wygenerowane przeze mnie i znajdują się na dysku google:

<https://drive.google.com/drive/folders/16Lex52AayZoPmcr0SsMjuWLAC8jpeke2?usp=sharing>):

1. graph1.csv
2. graph2.csv
3. graph3.csv]
4. graph4.csv
5. graph5.csv
6. test\_6\_1.csv
7. graph7.csv
8. graph8.csv
9. graph9.csv
10. test\_10.csv
11. graph11.csv
12. graph12.csv
13. graph13.csv

## 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 i pomiaru czasu dla każdej instancji danych badawczych.

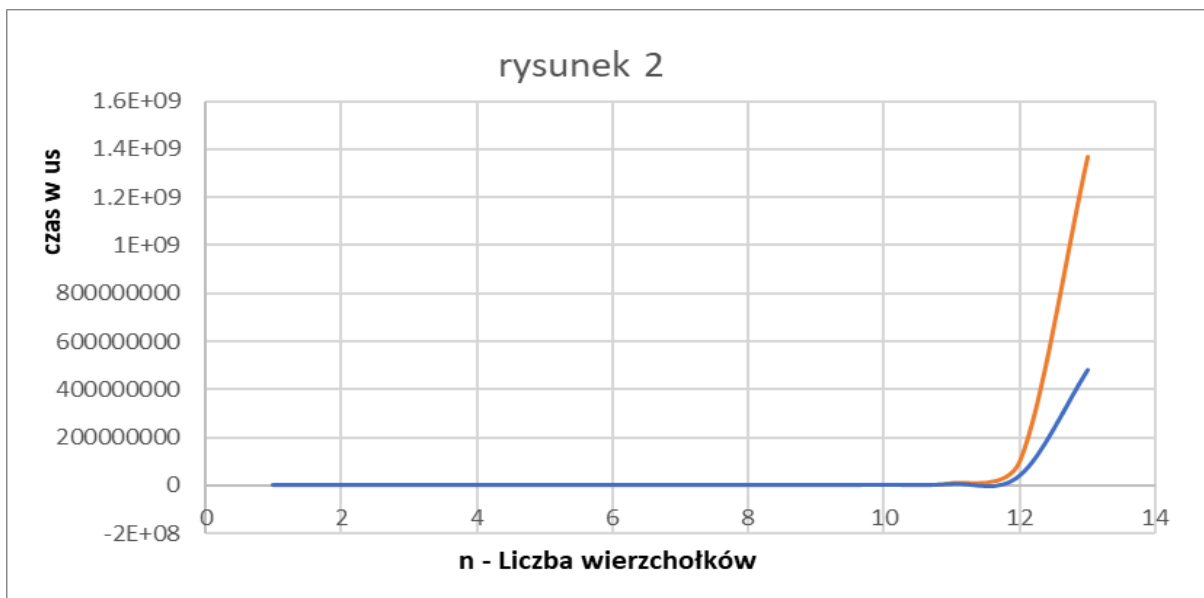
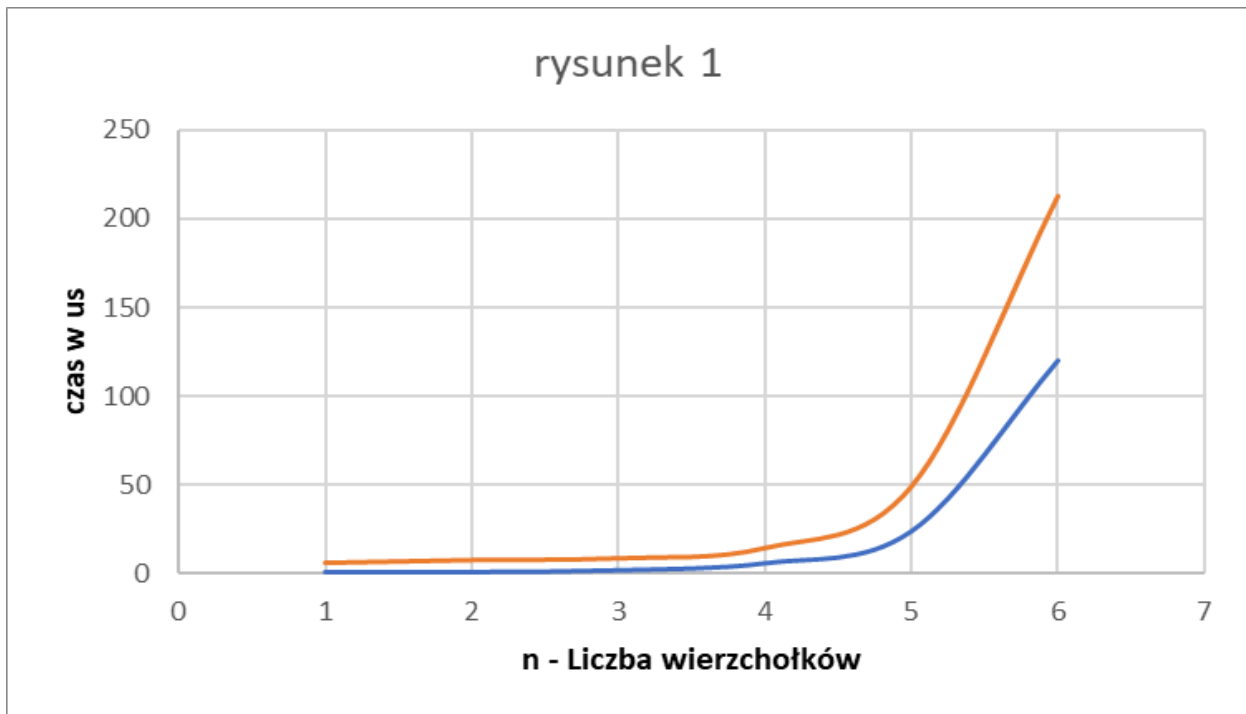
```
graph1.csv 50 0 [0]
graph2.csv 50 187 [0,1,0]
graph3.csv 50 148 [0, 1, 2, 0]
graph4.csv 50 139 [0, 3, 2, 1, 0]
graph5.csv 50 194 [0, 1, 4, 2, 3, 0]
test_6_1.csv 50 132 [0, 1, 2, 3, 4, 5, 0]
graph7.csv 50 239[0, 1, 2, 6, 4, 3, 5, 0]
graph8.csv 50 266 [0, 5, 6, 7, 4, 3, 2, 1, 0]
graph9.csv 50 276 [0, 5, 3, 4, 1, 2, 7, 6, 8, 0]
test_10.csv 50 212 [0, 3, 4, 2, 8, 7, 6, 9, 1, 5, 0]
graph11.csv 50 313 [0, 4, 3, 5, 7, 9, 10, 6, 2, 1, 8, 0]
graph12.csv 20 366 [0, 1, 10, 6, 9, 2, 4, 5, 11, 3, 7, 8, 0]
graph13.csv 20 398 [0, 4, 6, 9, 5, 1, 12, 11, 8, 2, 3, 10, 7, 0]
```

Każda instancji rozwiązywana była zgodnie z liczbą jej wykonań, np. graph1.csv wykonana została 50 razy. Do pliku wyjściowego data.csv zapisywany był czas wykonania algorytmu. Plik wyjściowy zapisywany był w formacie csv. Dane takie jak minimalna ścieżka oraz jej poszczególne węzły zostały ręcznie zapisane w pliku excel.

## 6. Wyniki

Wyniki zgromadzone zostały w plikach: data.csv, bfTSP.xlsx

Wyniki przedstawione zostały w postaci wykresu zależności czasu uzyskania rozwiązania problemu od wielkości instancji (rysunek 1 przedstawia dane dla  $n < 7$  w celu lepszego zaprezentowania krzywej wykresu, rysunek 2 zawiera wszystkie zbadane instancje).



## 7. Analiza wyników i wniosków

Krzywa wzrostu czasu (pomarańczowa) względem wielkości instancji ma charakter wykładniczy (rysunek 1, 2). Nałożenie krzywej  $O(n!)$  (niebieska) 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  $O(n!)$ .