



---

# STRUKTURY DANYCH I ZŁOŻONOŚĆ OBLICZENIOWA

---

Zadanie projektowe nr 3: Implementacja i analiza efektywności algorytmów optymalnych o pseudowielomianowej złożoności obliczeniowej dla wybranych problemów kombinatorycznych



7 CZERWCA 2017

ŁUKASZ BROLL

225972

Wstęp teoretyczny.....	2
Złożoność obliczeniowa .....	2
Dyskretny problem plecakowy.....	2
Przegląd zupełny .....	2
Algorytm zachłanny .....	2
Asymetryczny problem komiwojażera .....	3
Przegląd zupełny .....	3
Algorytm zachłanny .....	3
Plan eksperymentu .....	3
Wyniki eksperymentu .....	4
Dyskretny problem plecakowy.....	4
Przegląd zupełny .....	4
Algorytm zachłanny .....	6
Asymetryczny problem komiwojażera .....	8
Przegląd zupełny .....	8
Algorytm zachłanny .....	8
Wnioski .....	9
Dyskretny problem plecakowy.....	9
Asymetryczny problem komiwojażera .....	9
Bibliografia.....	9

Celem projektu implementacja oraz dokonanie pomiaru czasu działania algorytmów dla następujących problemów kombinatorycznych:

- Dyskretny problem plecakowy
- Asymetryczny problem komiwojażera

Dla obu problemów opracowane zostały algorytmy metodą przeglądu zupełnego (brute force) oraz algorytm zachłanny.

Czasy działania algorytmów:

- Problem plecakowy – przegląd zupełny:  $O(2^n)$
- Problem plecakowy – algorytm zachłanny:  $O(n \cdot \log n)$
- Problem komiwojażera – brute force:  $O(n!)$
- Problem komiwojażera – algorytm zachłanny:  $O(n \cdot \log n)$

### DYSKRETNY PROBLEM PLECAKOWY

Problem jest jednym z najczęściej poruszanych problemów optymalizacyjnych. Nazwa zagadnienia pochodzi od maksymalizacyjnego problemu wyboru przedmiotów, tak by ich sumaryczna wartość była jak największa i jednocześnie mieściły się w plecaku. Przy podanym zbiorze elementów o podanej wadze i wartości, należy wybrać taki podzbiór by suma wartości była możliwie jak największa, a suma wag była nie większa od danej pojemności plecaka.

---

#### PRZEGLĄD ZUPEŁNY

Algorytm sprawdza na początku wszystkie istniejące podzbiory zbioru  $n$  przedmiotów, a następnie odrzuca te, których waga sumaryczna przekracza pojemność plecaka. W tym momencie złożoność obliczeniowa jest duża. Na koniec algorytm porównuje ze sobą wszystkie podzbiory. Złożoność takiego algorytmu wynosi  $O(2^n)$ . Jest to bardzo nieefektywna metoda. Złożoność ta wynika z ciągu zero jedynkowego na  $n$  polach. Jest to najprostsza metoda wyszukiwania.

---

#### ALGORYTM ZACHŁANNY

W algorytmie zachłannym sortuje się tablicę przedmiotów i wrzuca po kolei do plecaka. Skraca to ogromnie czas obliczeń w porównaniu do metody brute force. Dla rozmiaru plecaka  $<1000$  otrzymywano wyniki oscylujące w okolicach 2 ms. Wykres można było utworzyć dla większej pojemności.

## ASYMETRYCZNY PROBLEM KOMIWOJAŻERA

Zagadnienie optymalizacyjne, polegające na znalezieniu minimalnego cyklu Hamiltona w pełnym grafie ważonym. 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. Symetryczny problem komiwojażera polega na tym, że dla dowolnych miast A i B odległość z A do B jest taka sama jak z B do A. W asymetrycznym problemie komiwojażera odległości te mogą być różne.

### PRZEGLĄD ZUPEŁNY

Metoda brute-force polega na znalezieniu permutacji wszystkich miast, wyliczenia sumy kosztów ścieżek między nimi i wybraniu najmniejszej. Ma dużą złożoność obliczeniową, czas wykonania może szybko zmienić się od kilkudziesięciu sekund do stuleci.

W założeniach projektu trzeba było zbadać średnie czasy dla  $>10$  miast, lecz już dla 12 eksperyment trwał bardzo długo. Zmniejszono zakres i uśredniono czasy dla 5, 8 i 10 miast.

### ALGORYTM ZACHŁANNY

W algorytmie zachłannym nie dokonuje się permutacji miast. Program na bieżąco sprawdza, którą trasę obrać, kierując się wytyczną – jak najtaniej. Skraca to ogromnie czas obliczeń w porównaniu do metody brute-force. Dla liczby miast  $<1000$  otrzymywano wyniki oscylujące w okolicach 2 ms. Wykres można było utworzyć dla ponad 1000 miast.

## PLAN EKSPERYMENTU

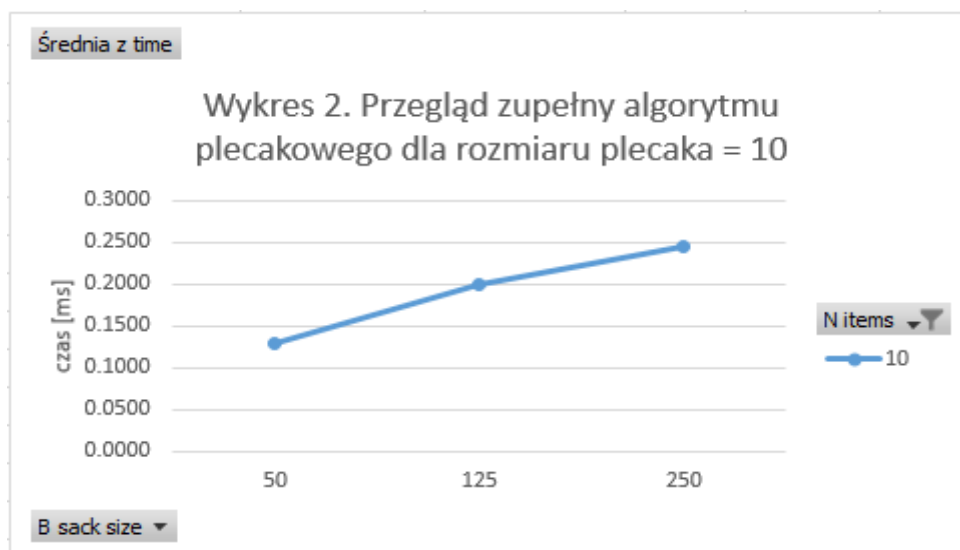
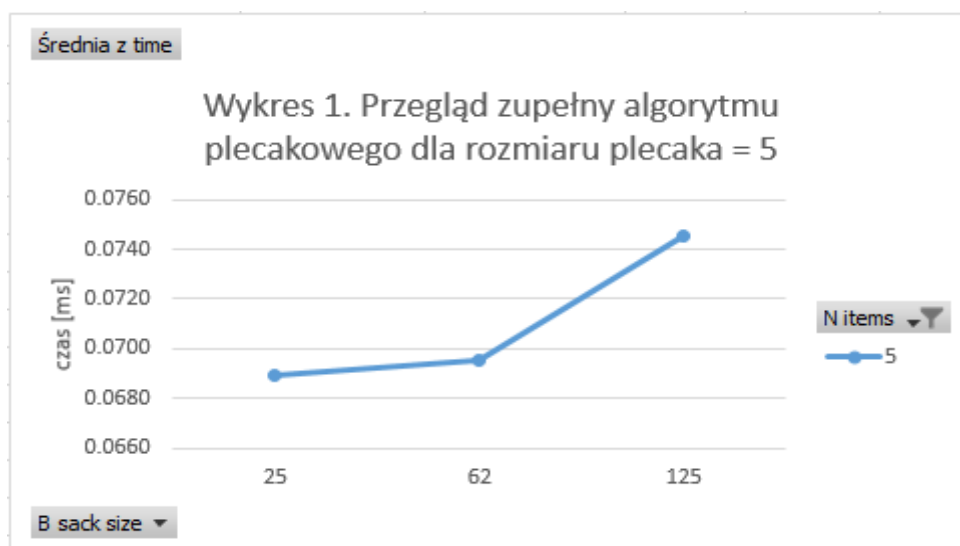
- Wielkość struktur ustalana jest dynamicznie,
- Do reprezentacji odległości między miastami wykorzystano macierz sąsiedztwa,
- Wszystkie badania przeprowadzać w wersji RELEASE,
- Program potrzebny do badania podstawowych struktur danych został napisany w języku C++ w programie Visual Studio 2015,
- Specyfikacja laptopa, na którym wykonywane były testy:

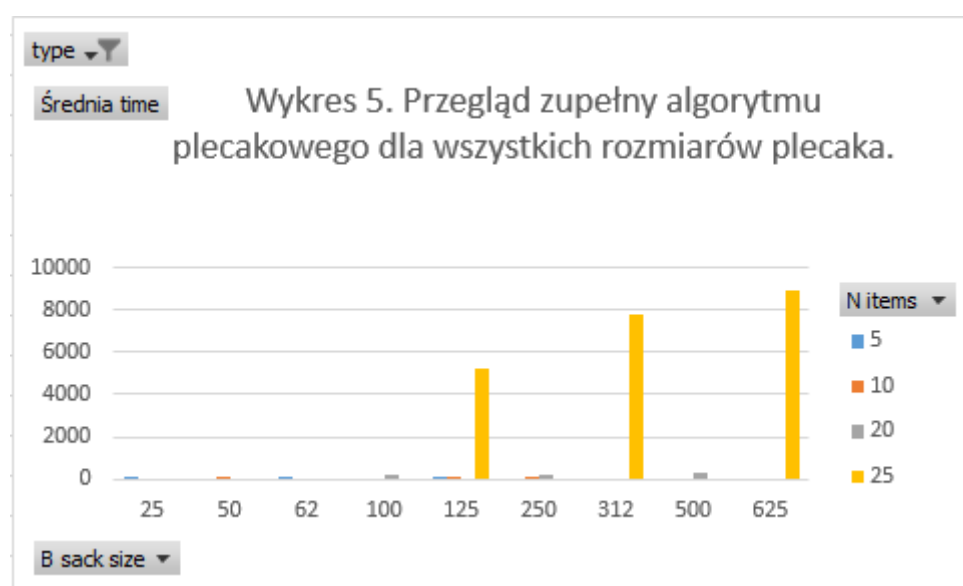
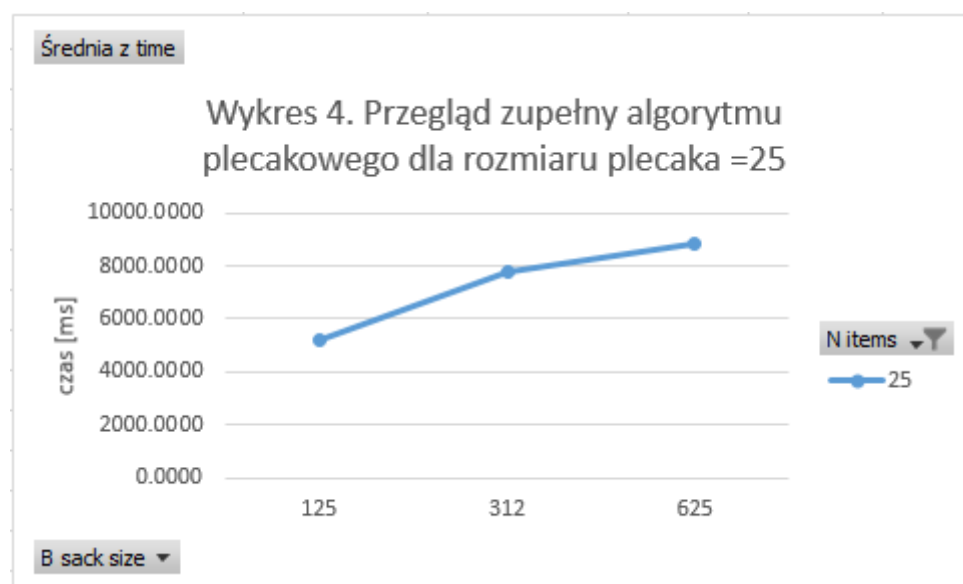
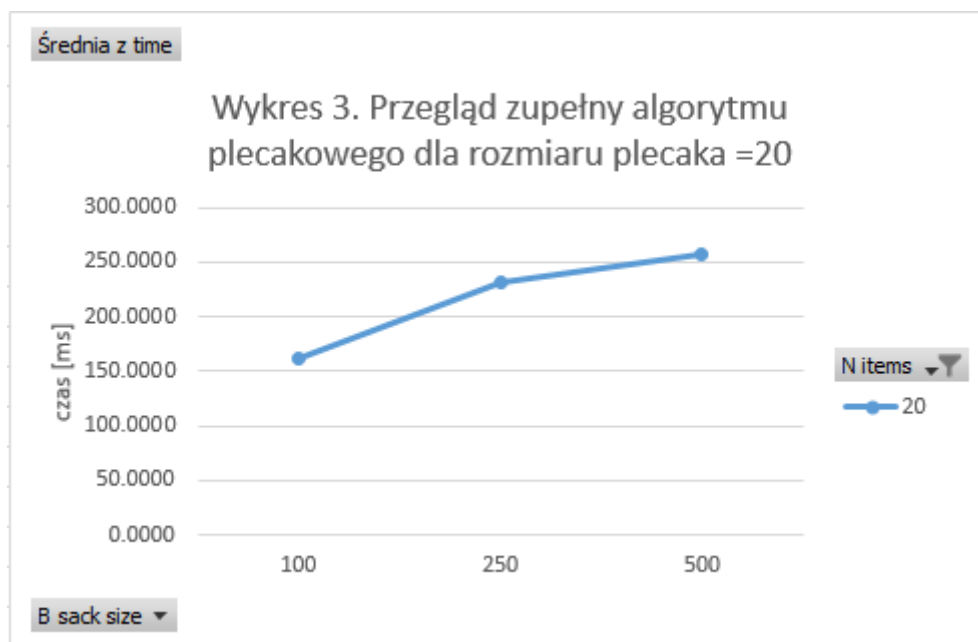
#### System

Processor:	Intel(R) Core(TM) i5-2410M CPU @ 2.30GHz 2.30 GHz
Installed memory (RAM):	8.00 GB (7.90 GB usable)
System type:	64-bit Operating System, x64-based processor
Pen and Touch:	No Pen or Touch Input is available for this Display

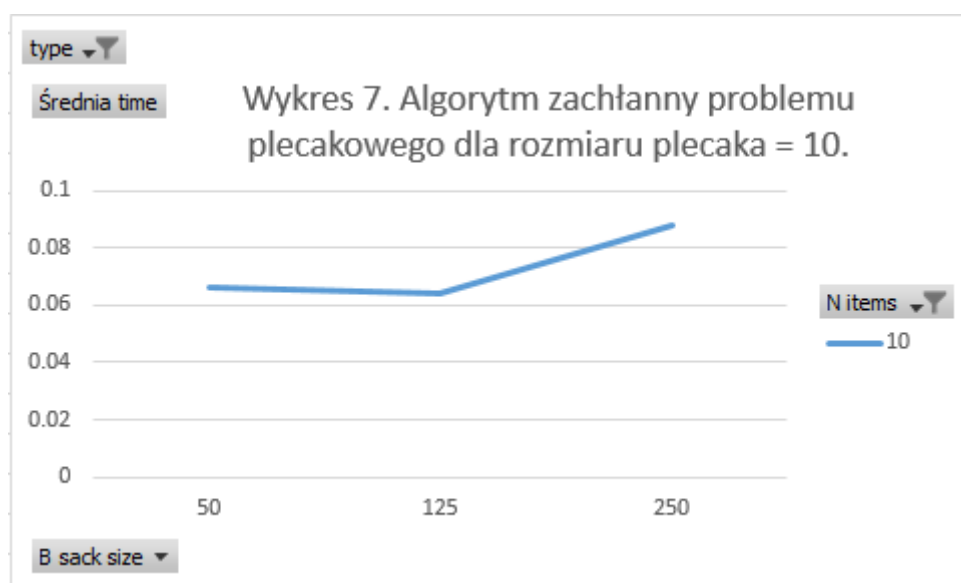
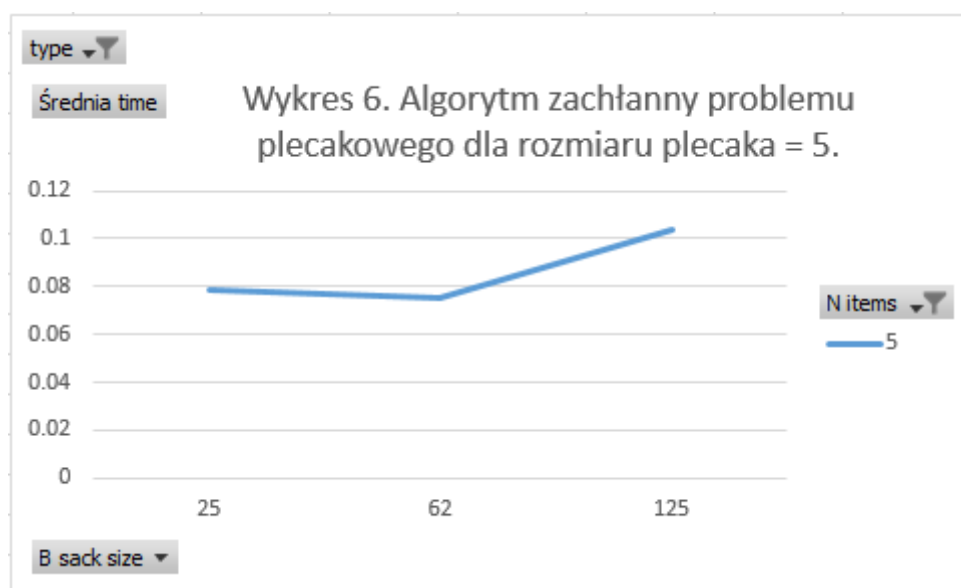
- Pomiarów czasu dokonano za pomocą funkcji:  
`BOOL QueryPerformanceCounter ( __out LARGE_INTEGER *lpPerformanceCount );`  
Źródło licznika: <http://cpp0x.pl/forum/temat/?id=21331>

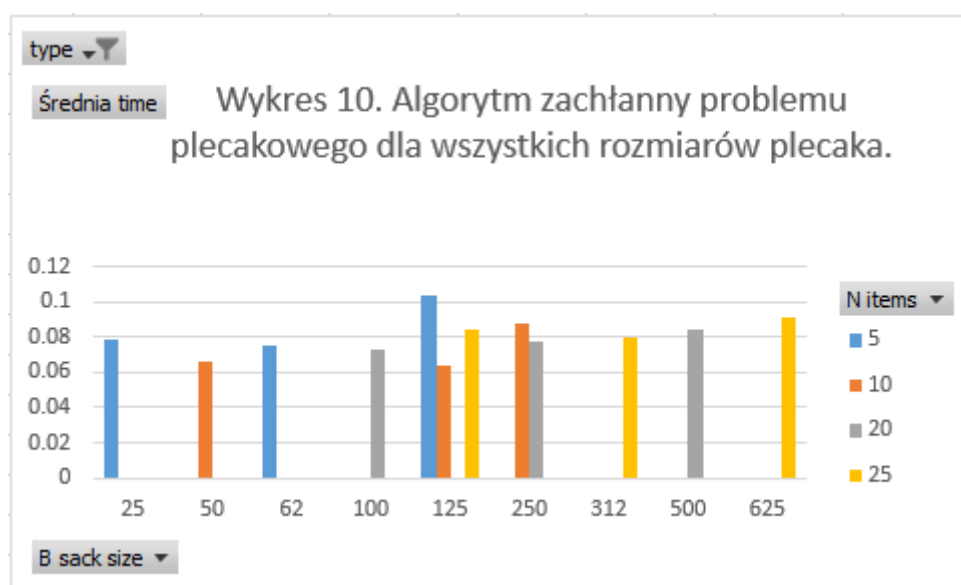
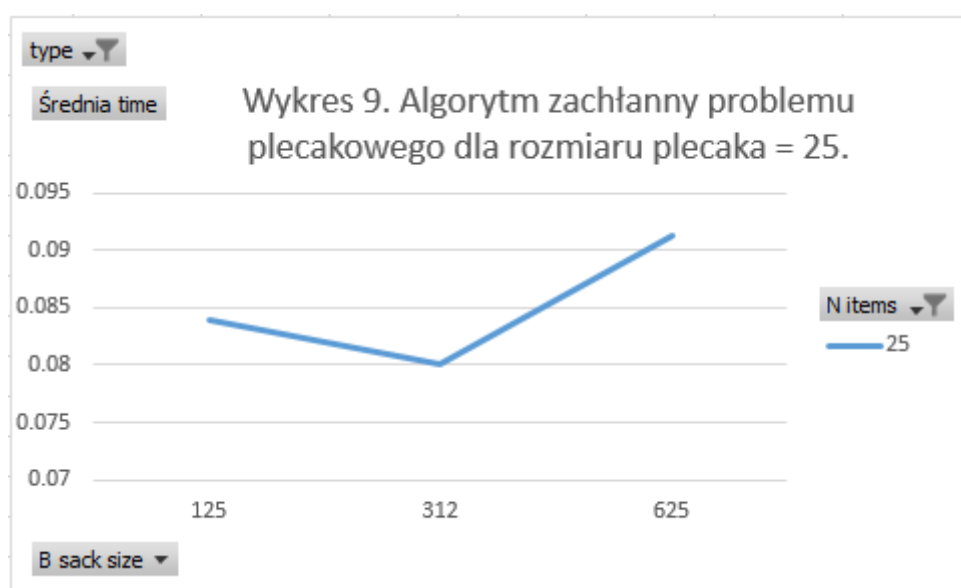
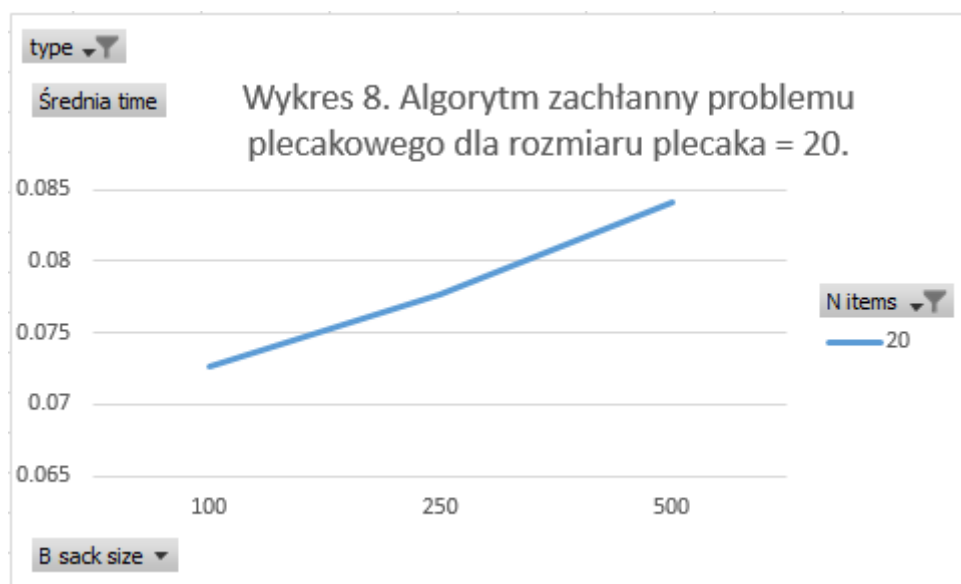
Średnia czasy	Ilość przedmiotów			
Rozmiar plecaka	5	10	20	25
25	0.0689			
50		0.1280		
62	0.0695			
100			161.9588	
125	0.0745	0.1990		5212.8999
250		0.2438	231.0820	
312				7806.7250
500			257.1949	
625				8865.1039





Średnia czasy	Ilość przedmiotów			
Rozmiar plecaka	5	10	20	25
25	0.0786			
50		0.0662		
62	0.0753			
100			0.0727	
125	0.1040	0.0639		0.0839
250		0.0876	0.0777	
312				0.0801
500			0.0841	
625				0.0912

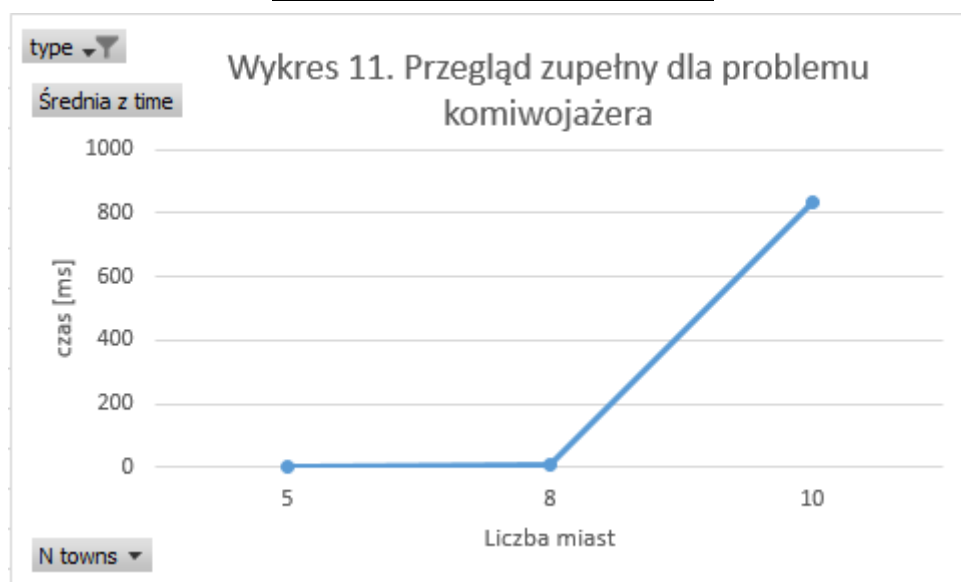






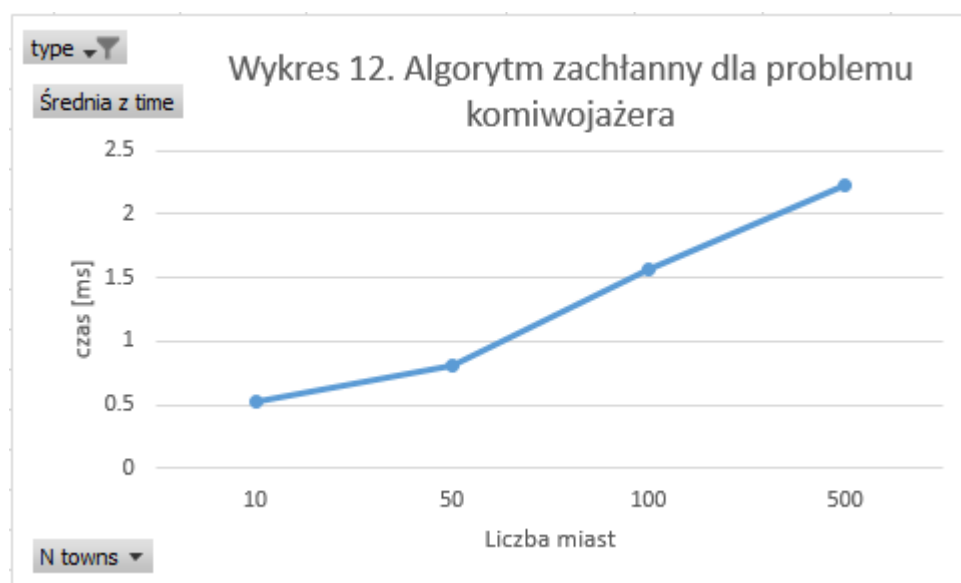
## PRZEGLĄD ZUPEŁNY

Ilość miast	Średnie czasy
5	0.0925
8	8.8576
10	835.3037



## ALGORYTM ZACHŁANNY

Liczba miast	Średnie czasy
10	0.5284
50	0.8112
100	1.5667
500	2.2338



---

DYSKRETNY PROBLEM PLECAKOWY

- Algorytm zachłanny

Z przedstawionych wykresów widać zależność problemu od czasu jego wykonania. Im problem stawał się większy, bardziej skomplikowany, czas algorytmu również wydłużał się. Na wykresie widać, że problem rośnie liniowo, co jest zgodne z logiką. Algorytm wykonuje się odpowiednio długo, tzn. czas jest w sam raz dla danego problemu.

- Przegląd zupełny

Jak widać na wykresie, algorytm ten naprawdę jest nieefektywny. Uzyskane czasy zgadzają się z teorią i złożonością tego algorytmu. Algorytm dla coraz to większej liczby  $n$ , wykonuje się  $2^n$  razy dłużej. Różna pojemność plecaka nie miała większego wpływu na czas działania algorytmu. Największy problem w tym algorytmie jest to, że przegląda wszystkie możliwości. Pętla wykonywała się  $2^n$  razy, co sprawiało, że algorytm trwa bardzo długo.

---

ASYMETRYCZNY PROBLEM KOMIWOJAŻERA

- Przegląd zupełny

Algorytm ten jest nieefektywny, zajmuje bardzo dużo czasu przy większej ilości miast, lecz przy kilku – kilkunastu jest w stanie prawie idealnie podać najbardziej optymalną ścieżkę dla komiwojażera.

- Algorytm zachłanny

O ile czasy drastycznie zmalały i można tym algorytmem zbadać o wiele większą ilość miast, tak wyniki nie były już zadowalające. Wydawać by się mogło, że kryterium najtańszej drogi od ostatnio odwiedzonego miasta brzmi rozsądnie, jednak nie zawsze oznacza to optymalną trasę. Czasem metodą brute force można było otrzymać bardziej optymalną ścieżkę.

---

BIBLIOGRAFIA

1. Cormen Thomas H., Leiserson Charles E., Rivest Ronald L., *Wprowadzenie do Algorytmów*, wyd. IV, Warszawa: Wydawnictwa Naukowo-Techniczne
2. Problem plecakowy – Wikipedia [https://pl.wikipedia.org/wiki/Problem\\_plecakowy](https://pl.wikipedia.org/wiki/Problem_plecakowy)
3. Problem komiwojażera – Wikipedia [https://pl.wikipedia.org/wiki/Problem\\_komiwoja%C5%BCera](https://pl.wikipedia.org/wiki/Problem_komiwoja%C5%BCera)