

Projektowanie Efektywnych Algorytmów

Projekt

18/10/2023

263916 Kamil Gondek

(1) Brute force

Spis treści	strona
Sformułowanie zadania	2
Metoda	3
Algorytm	4
Dane testowe	5
Procedura badawcza	6
Wyniki	7
Analiza wyników i wnioski	8

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 trasy, która przebiega przez wszystkie miasta dokładnie raz a następnie wraca do 1 miasta. Jest to problem optymalizacji kombinatorycznej, który ma zastosowanie między innymi w logistyce i planowaniu tras w transporcie publicznym.

Algorytm przeglądu zupełnego (Brute force), polegający na sprawdzeniu wszystkich możliwych kombinacji rozwiązań, jest skuteczny dla małych instancji problemu komiwojażera.

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).

Zaletą tej metody jest łatwość w jej implementacji. Jednakże metoda ta jest bardzo mało skuteczna dla większych instancji problemu czego przyczyną jest to, że ta metoda sprawdza wszystkie możliwe kombinacje a następnie szuka w śród nich najlepszej co skutkuje złożonością czasową na poziomie $O(n!)$.

3. Algorytm

Mój algorytm w pierwszym kroku w sposób rekurencyjny generuje wszystkie możliwe kombinacje odwiedzania wierzchołków dzięki czemu tworzymy wektor zawierający wektory kolejnych wierzchołków tj. dla liczby wierzchołków równej 3 algorytm wygeneruje następujące wektory:

1 2 3

1 3 2

2 1 3

2 3 1

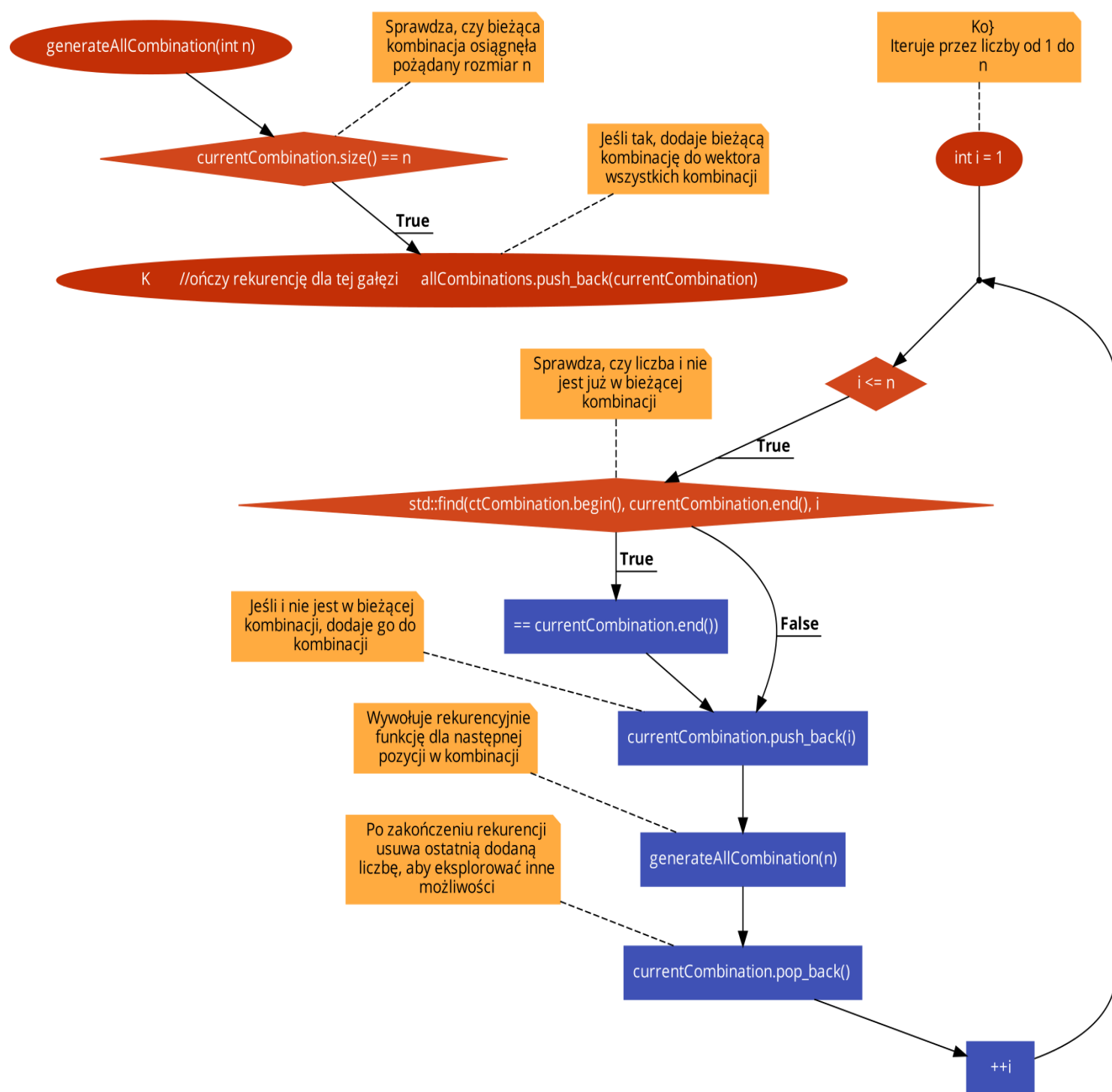
3 1 2

3 2 1

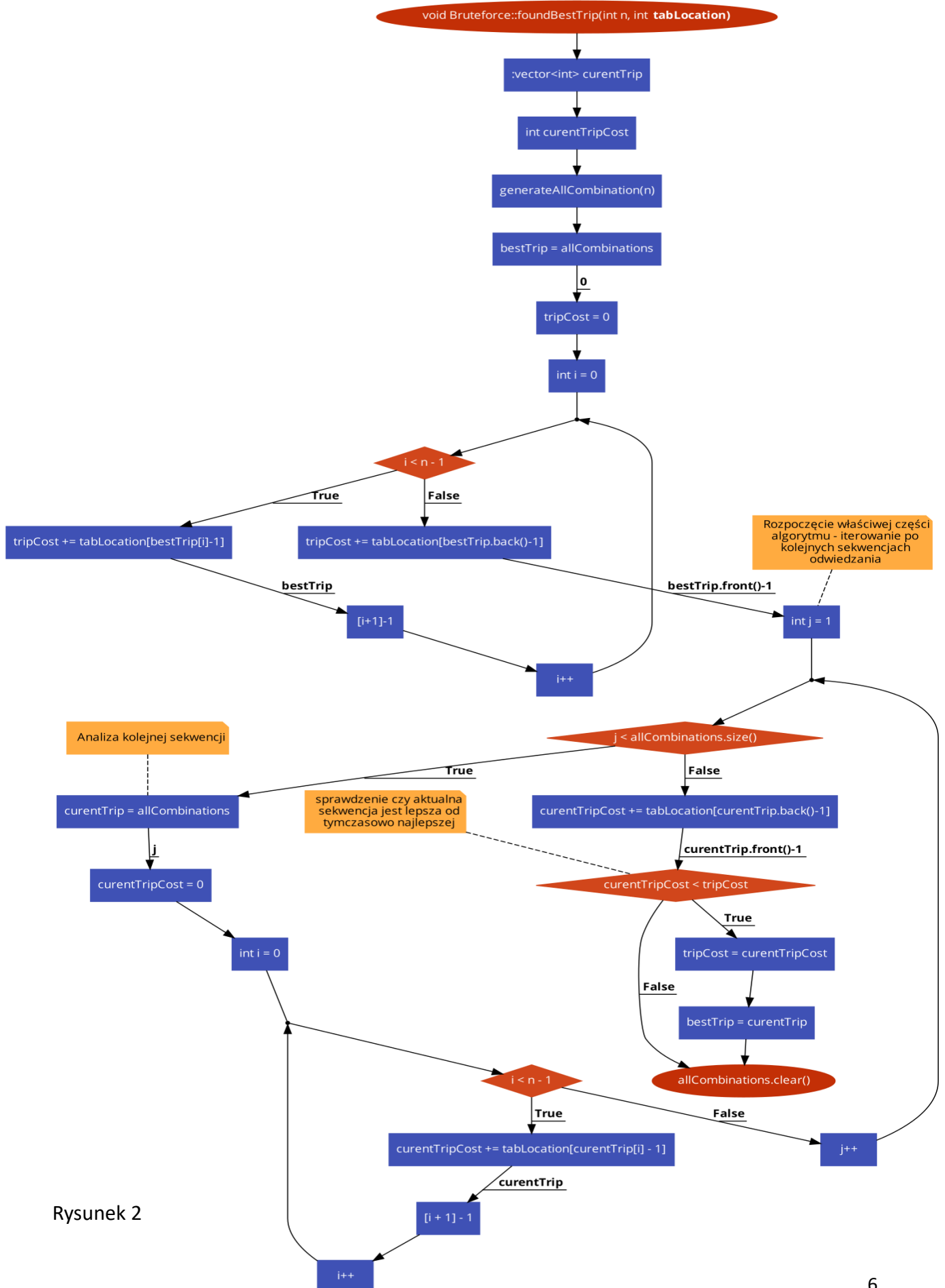
Skutkiem takiego podejścia jest złożoność pamięciowa $O(n!)$, ponieważ musimy przechowywać w pamięci wszystkie kombinacje. Jednakże takie podejście daje możliwość nieznacznego przyspieszenia kodu. Gdyż dla kolejnych instancji zawierających taką samą ilość wierzchołków nie musimy na nowo generować sekwencji odwiedzania. Tylko iterujemy po już wygenerowanej co nieznacznie skraca czas obliczeniowy.

Kolejnym krokiem jest już iterowanie po wszystkich sekwencjach i sprawdzanie czy nowa sekwencja jest lepsza od poprzednich. Realizowane jest to w następujący sposób. Przypisujemy pierwszą utworzoną sekwencję obliczamy jej koszt i uznajemy za najlepszą a następnie sprawdzamy czy kolejna jest lepsza, jeśli tak to nową uznajemy za najlepszą.

Poniżej przedstawiono algorytm generowania wszystkich możliwych sekwencji (Rysunek 1) jak i algorytm Brute Force wykorzystujący wygenerowane sekwencje (Rysunek 2).



Rysunek 1



Rysunek 2

4. Dane testowe

Do sprawdzenia poprawności działania programu wykorzystano dane testowe dr Jarosława Mierzwy. Na ich podstawie uzyskano następujące wyniki:

tsp_6_1.txt

Trip :1 2 3 4 5 6

Total Cost = 132

tsp_6_2.txt

Trip :1 6 2 3 4 5 1

Total Cost = 80

tsp_10.txt

Trip :1 4 5 3 9 8 7 10 2 6 1

Total Cost = 212

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 (format pliku: Tryb (0 – Pomiar automatyczny / 1 – Pomiar na podstawie danych z pliku); Ilość Pomiarów; Ilość Powtórzeń Pomiaru; Plik Wyjściowy; Plik Wejściowy)

Przy wyborze trybu automatycznego program odczytuje ilość pomiarów, ilość powtórzeń i lokalizację wyjściową.

Przy wyborze trybu pomiaru z pliku program odczytuje lokalizację wejściową i wyjściową

Do pomiarów wykorzystano automatyczne generowanie instancji zawierających od 3 do 11 wierzchołków i wagi odległości z zakresów 1-10, 1-1 000, 1-1 000 000. Dla każdego zakresu wygenerowano 200 instancji a następnie dla każdej instancji wykonano 40 pomiarów.

Dane wyjściowe dla trybu automatycznego to lista zawierające informacje o numerze pomiaru, próbie badawczej np.:(1-10), powtórzeniu pomiaru konkretnej instancji, Czasie zapisanym w [ms] oraz Liczbie wierzchołków w następujący sposób:

Numer pomiaru; Próba badawcza; Numer pomiaru instancji; Czas[ms]; Liczba wierzchołków

Dane wyjściowe dla trybu pomiaru z pliku to zapis kolejności odwiedzanych wierzchołków oraz całkowity koszt odwiedzania w następującym formacie:

Trip :1 4 5 3 9 8 7 10 2 6

Total Cost = 173

6. Wyniki

Wyniki pomiarów zostały zgromadzone w plikach:

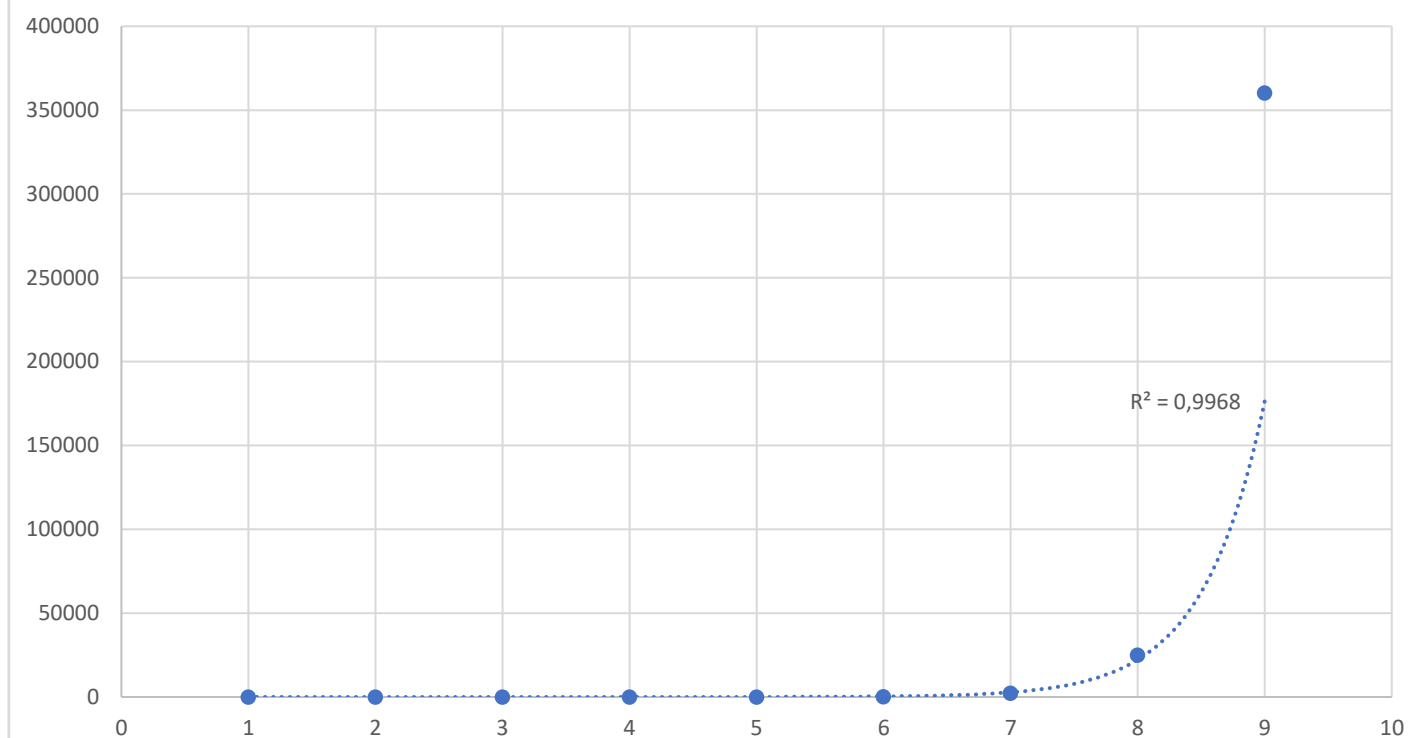
„BruteForceTest 3-11.txt” – plik zawiera dane pomiarowe dla liczby wierzchołków od 3 do 11, przy czym dla każdej liczby wierzchołków wygenerowano 200 instancji. Dla każdej instancji dokonano po jednym pomiarze czasu dla danej instancji. Z wyłączeniem pomiarów dla 11 wierzchołków dla którego dokonano tylko 17 pomiarów ze względu na czas pomiarowy.

„BruteForceTest 3-9 x40.txt” – plik zawiera dane pomiarowe dla liczby wierzchołków od 3 do 9, przy czym dla każdej liczby wierzchołków wygenerowano 200 instancji. Dla każdej instancji dokonano po 40 pomiarze czasu dla danej instancji.

Liczba wierzchołków	Wartość drogi		
	1-10	1-1000	1-1000000
3	0,0248	0,0247	0,0245
4	0,1008	0,0992	0,0992
5	0,510	0,509	0,508
6	3,34	3,33	3,33
7	30,7	30,5	30,2
8	242,1	241,6	242,4
9	2308,1	2309,1	2305,8
10	25026,4	25083,5	25019,3
11	360273,0	361451,0	367476,4

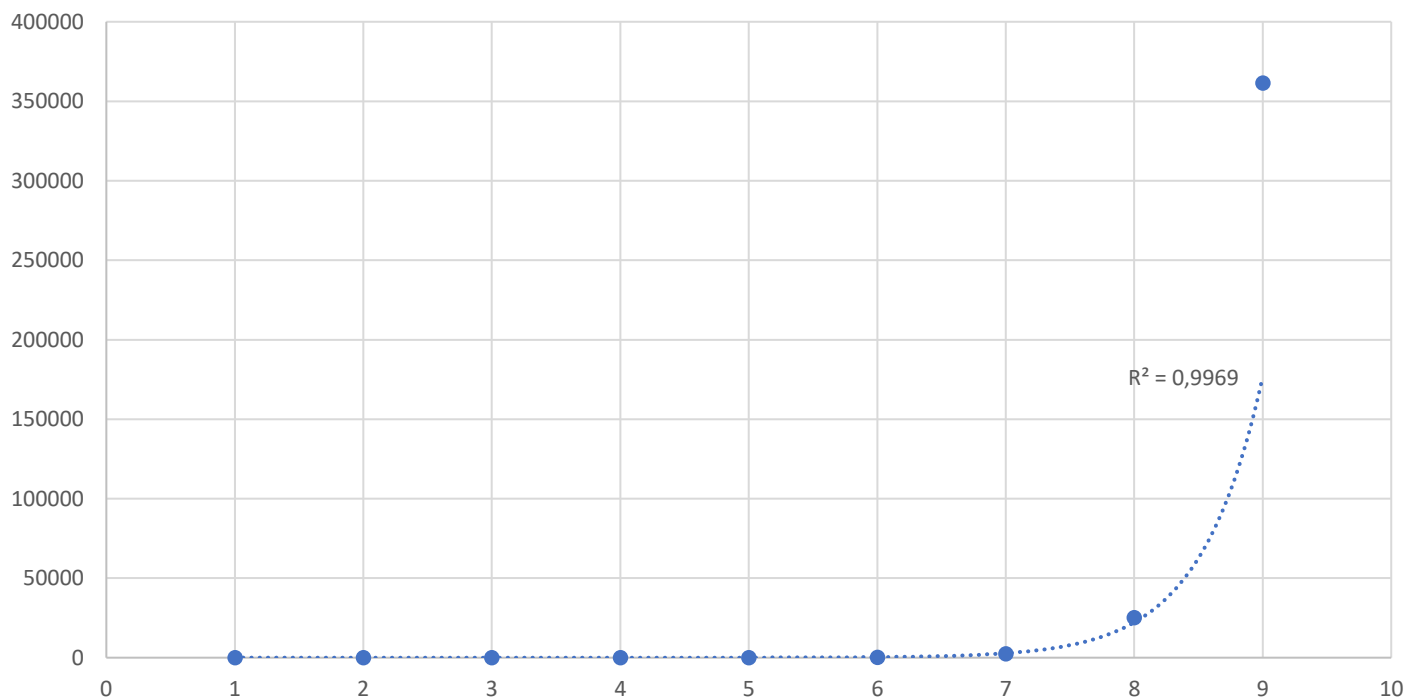
Tabela 1 – Wartości średnie czasu w zależności od liczby wierzchołków zapisany w milisekundach dla danych pomiarowych z „BruteForceTest 3-11.txt”

Wykres czasu wykonywania algorytmu do liczby wierzchołków o odległości z przedziału 1-10



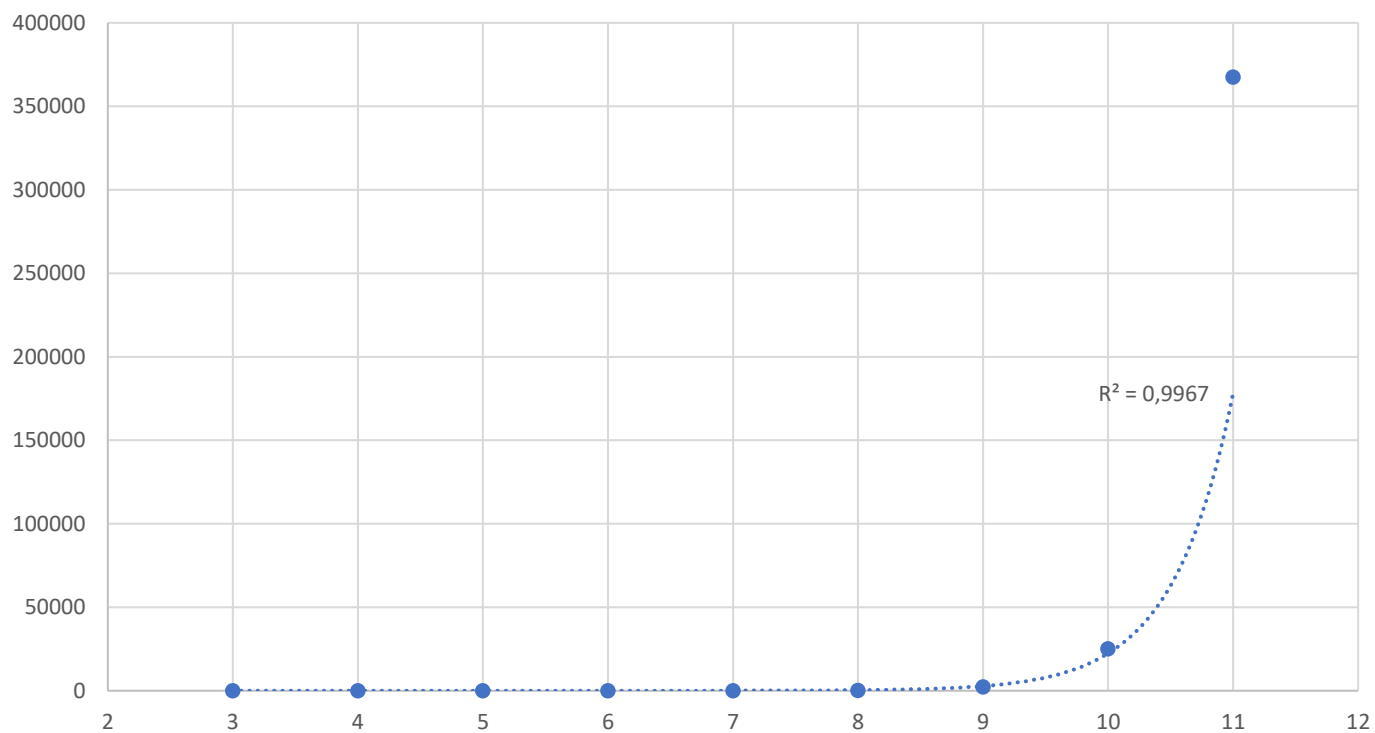
Wykres 1

Wykres czasu wykonywania algorytmu do liczby wierzchołków o odległości z przedziału 1-1000



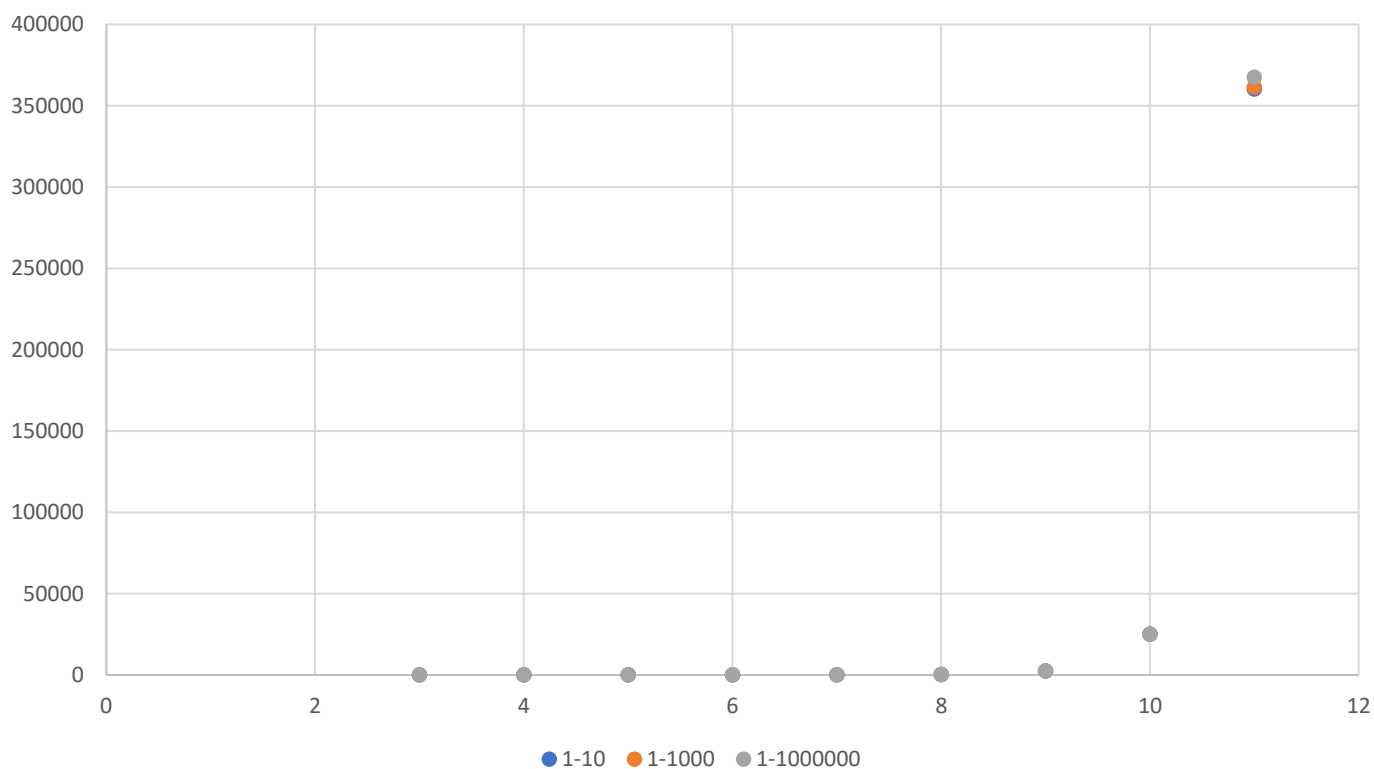
Wykres 2

Wykres czasu wykonywania algorytmu do liczby wierzchołków o odległości z przedziału 1-1000000



Wykres 3

Wykres czasu wykonywania algorytmu do liczby wierzchołków



Wykres 4

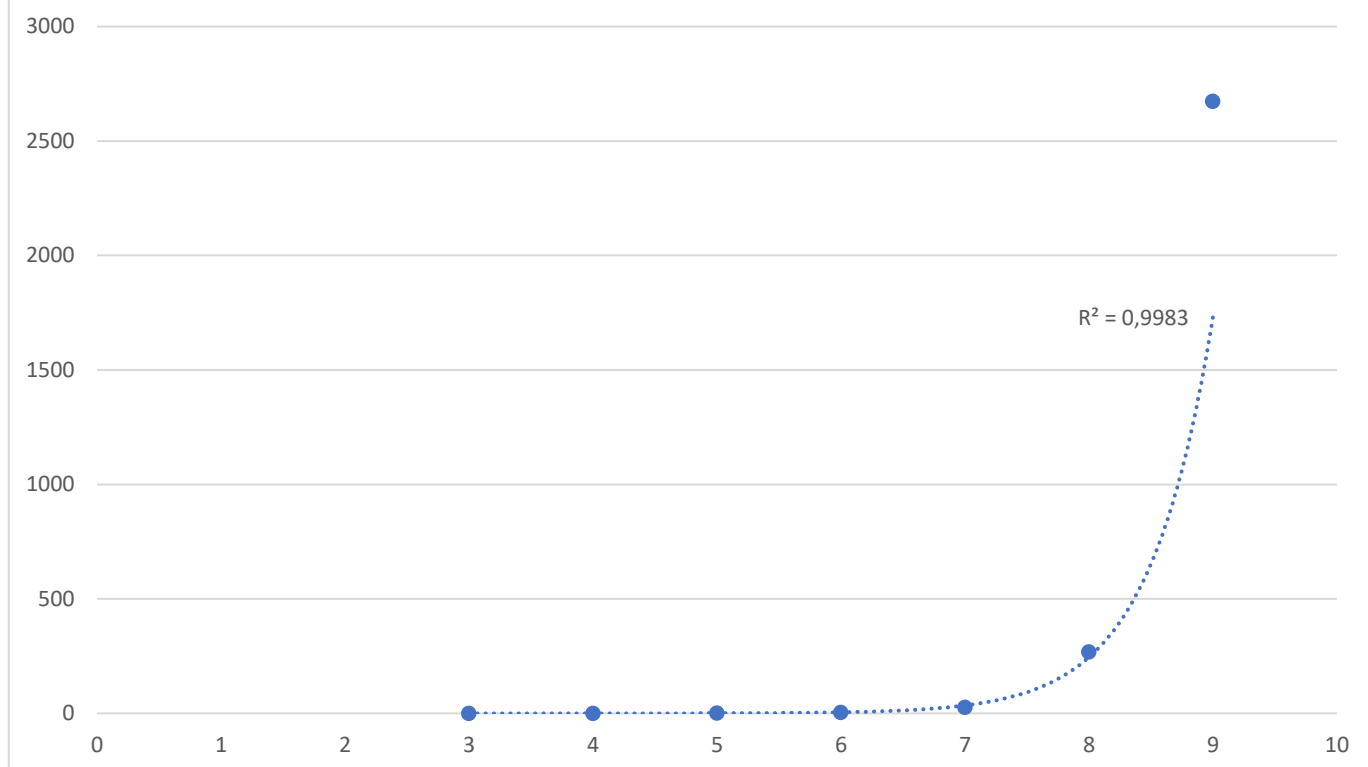
Wykresy 1-3 przedstawiają dane pomiarowe z pliku „BruteForceTest 3-11.txt” i reprezentują rozkład średniego czasu wykonywanego algorytmu dla kolejnych ilości wierzchołków. Wykresy te odróżnia jedynie wartości odległości między wierzchołkami.

Wykres 4 przedstawia nałożone na siebie wykresy 1-3

Liczba wierzchołków	Wartość drogi		
	1-10	1-1000	1-1000000
3	0,0240	0,0236	0,0239
4	0,0929	0,0939	0,0934
5	0,521	0,524	0,527
6	3,29	3,30	3,29
7	25,8	25,8	25,8
8	268,5	268,3	268,2
9	2672,9	2668,6	2670,4

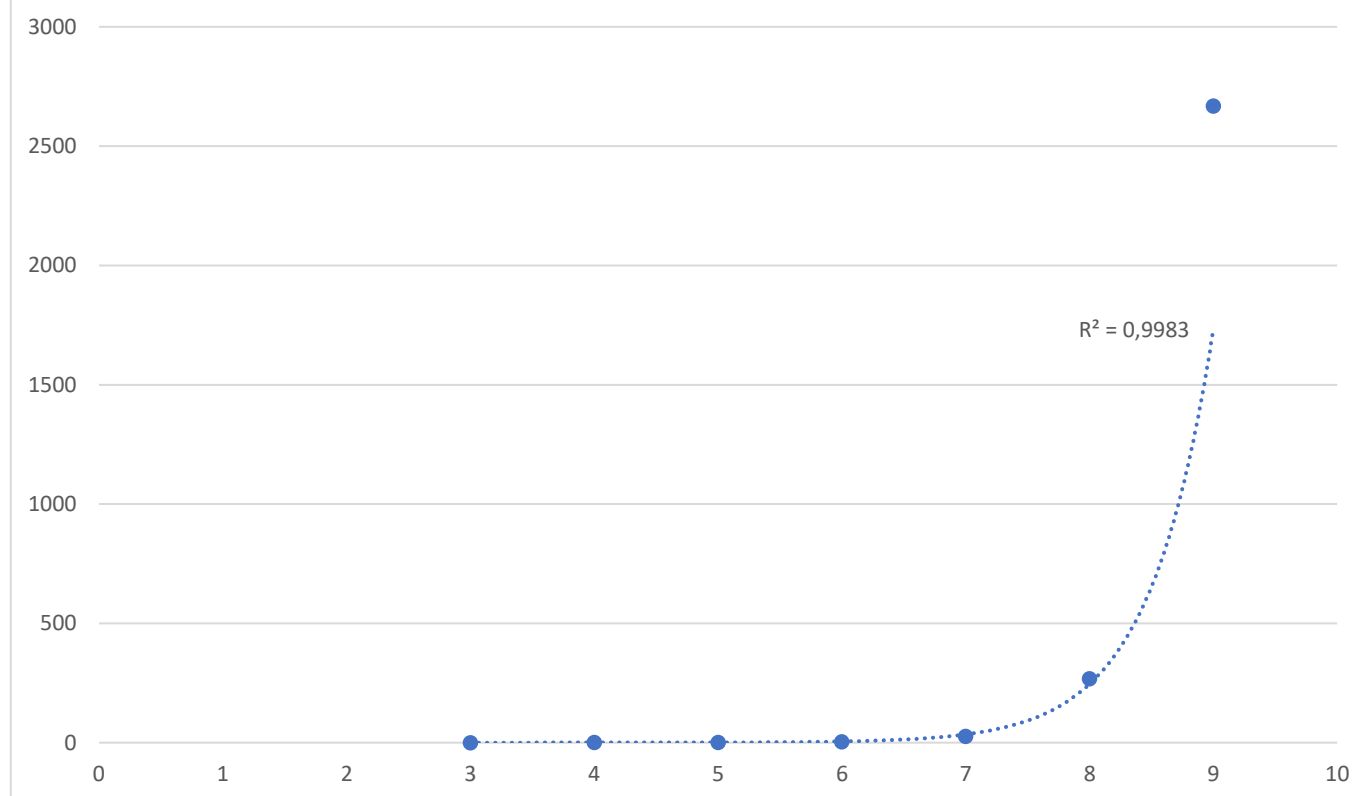
Tabela 2 - Wartości średnie czasu w zależności od liczby wierzchołków zapisany w milisekundach dla danych pomiarowych z „BruteForceTest 3-9 x40.txt”

Wykres czasu wykonywania algorytmu do liczby wierzchołków o
odległości z przedziału 1-10

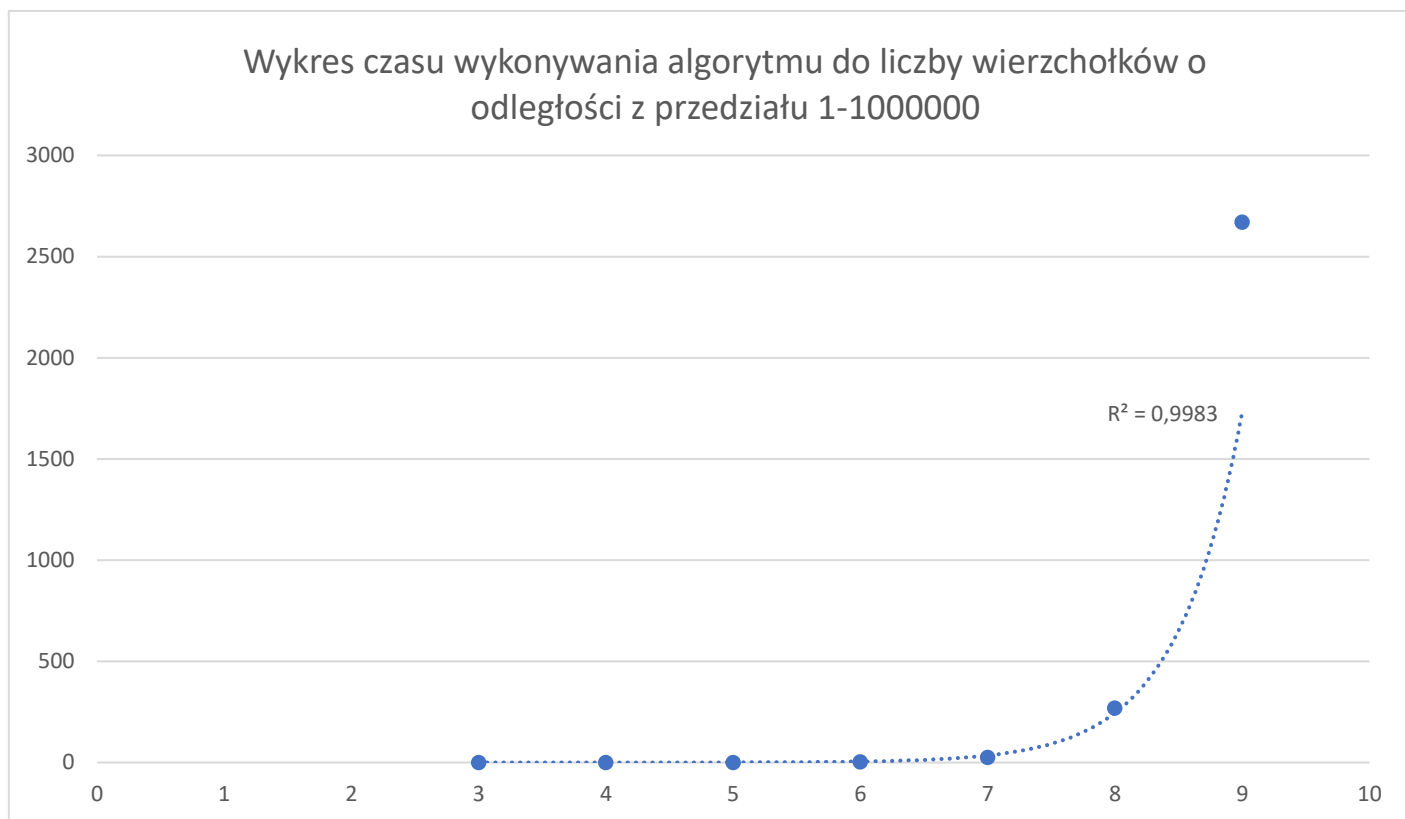


Wykres 5

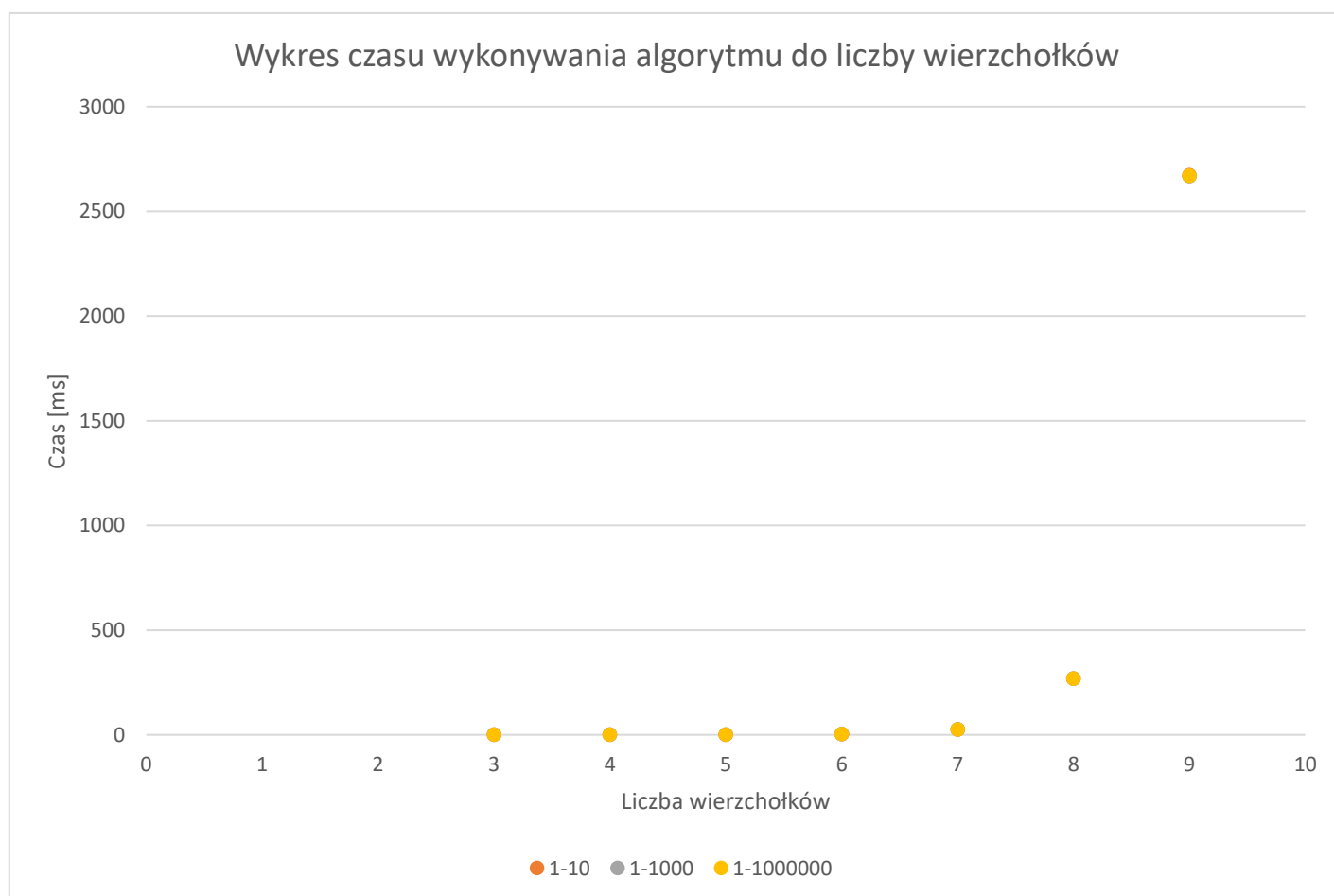
Wykres czasu wykonywania algorytmu do liczby wierzchołków o
odległości z przedziału 1-1000



Wykres 6



Wykres 7



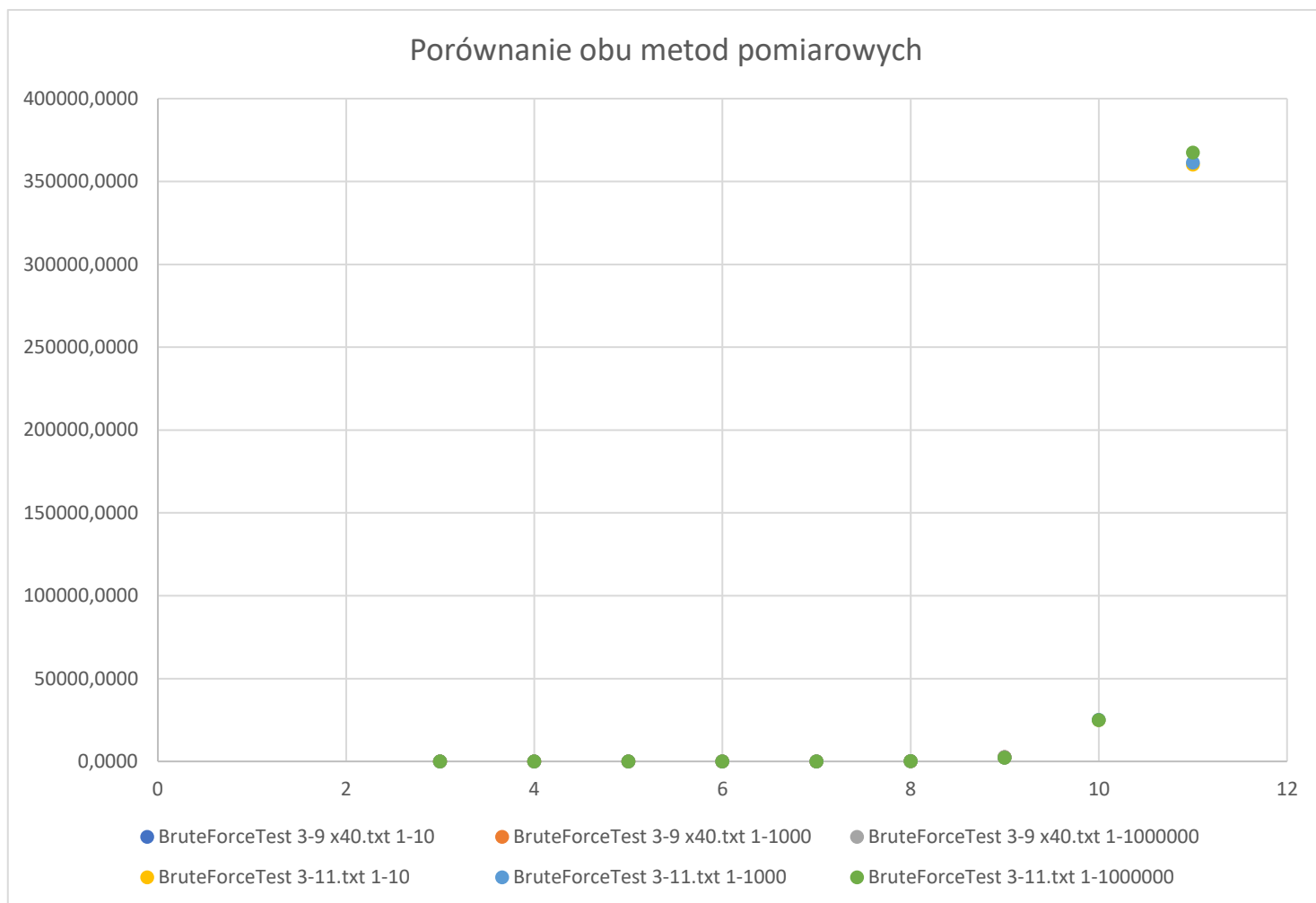
Wykres 8

Wykresy 5-7 przedstawiają dane pomiarowe z pliku „BruteForceTest 3-9 x40.txt” i reprezentują rozkład średniego czasu wykonywanego algorytmu dla kolejnych ilości wierzchołków. Wykresy te odróżnia jedynie wartości odległości między wierzchołkami.

Wykres 8 przedstawia nałożone na siebie wykresy 5-7

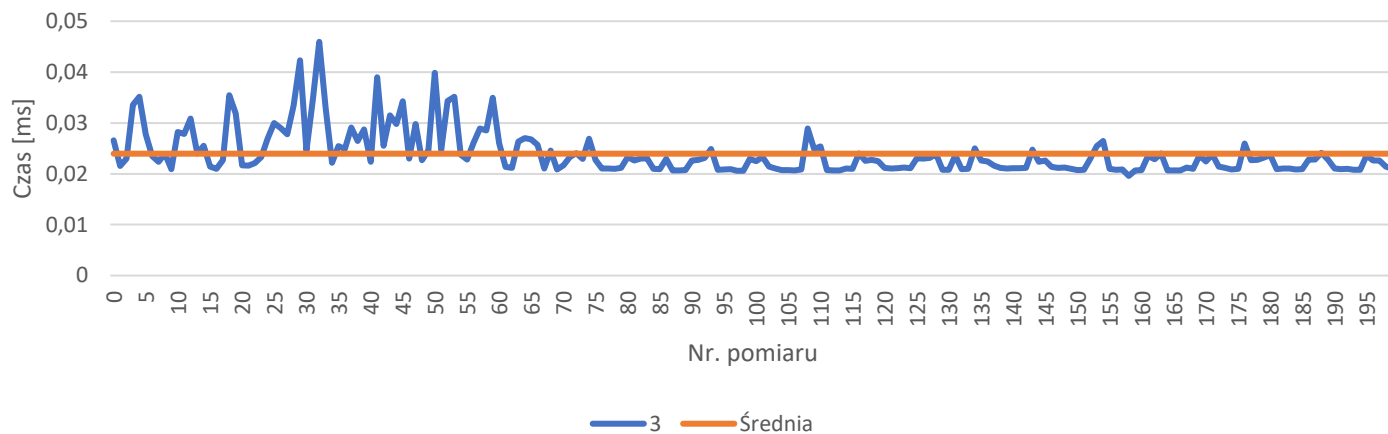
Liczba wierzchołków	BruteForceTest 3-9 x40.txt			BruteForceTest 3-11.txt		
	1-10	1-1000	1-1000000	1-10	1-1000	1-1000000
3	0,0240	0,0236	0,0239	0,0248	0,0247	0,0245
4	0,0929	0,0939	0,0934	0,1008	0,0992	0,0992
5	0,521	0,524	0,527	0,510	0,509	0,508
6	3,29	3,30	3,29	3,34	3,33	3,33
7	25,8	25,8	25,8	30,7	30,5	30,2
8	268,5	268,3	268,2	242,1	241,6	242,4
9	2672,9	2668,6	2670,4	2308,1	2309,1	2305,8
10	-	-	-	25026,4	25083,5	25019,3
11	-	-	-	360273,0	361451,0	367476,4

Tabela 3 – Zestawienie danych z Tabeli 1 i 2



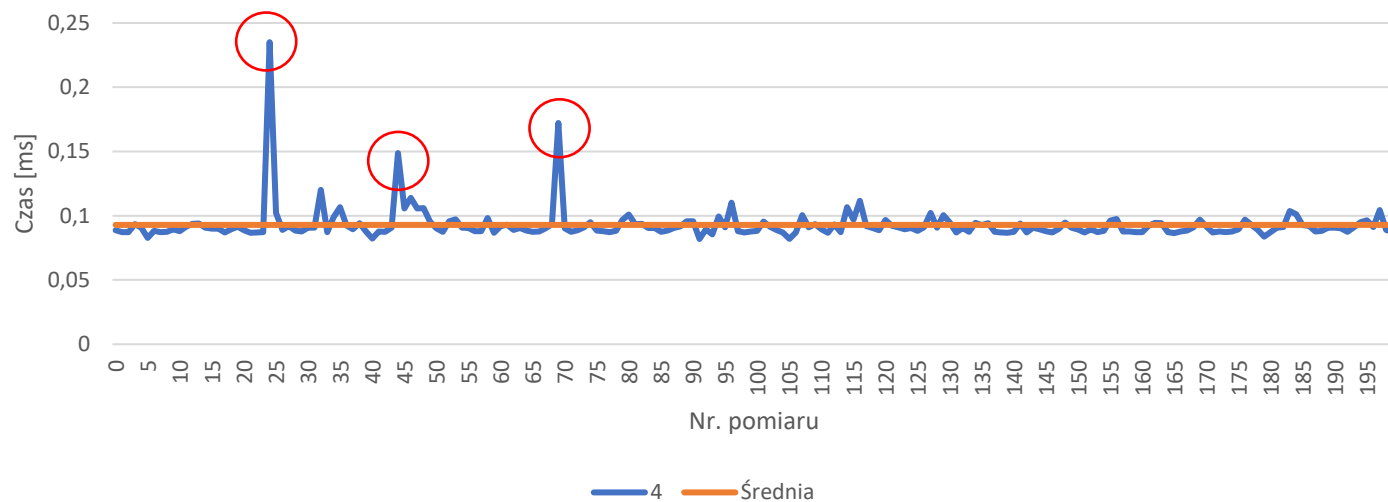
Wykres 9 – przedstawia dane zestawione z wykresu 4 i 8

Liczba wierzchołków 3 dla próby 1-10



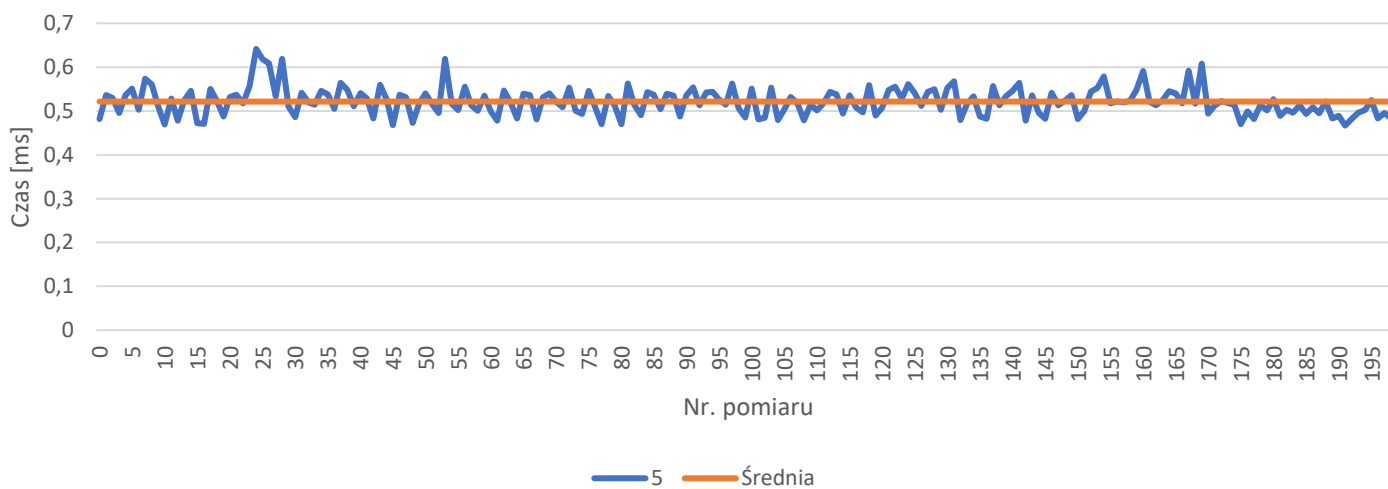
Wykres 10

Liczba wierzchołków 4 dla próby 1-10



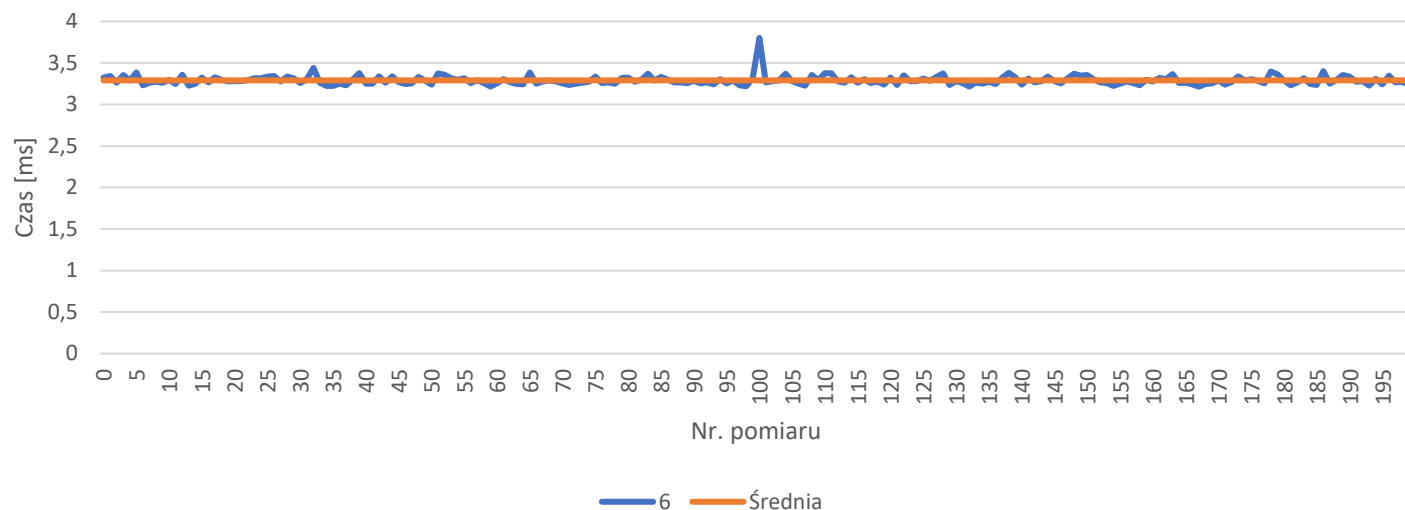
Wykres 11

Liczba wierzchołków 5 dla próby 1-10



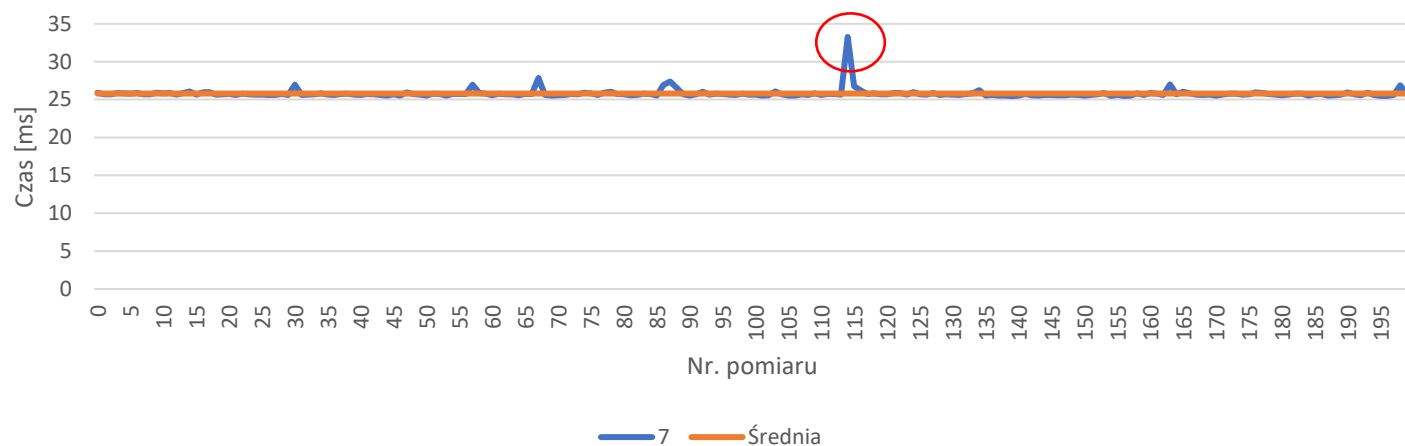
Wykres 12

Liczba wierzchołków 6 dla próby 1-10



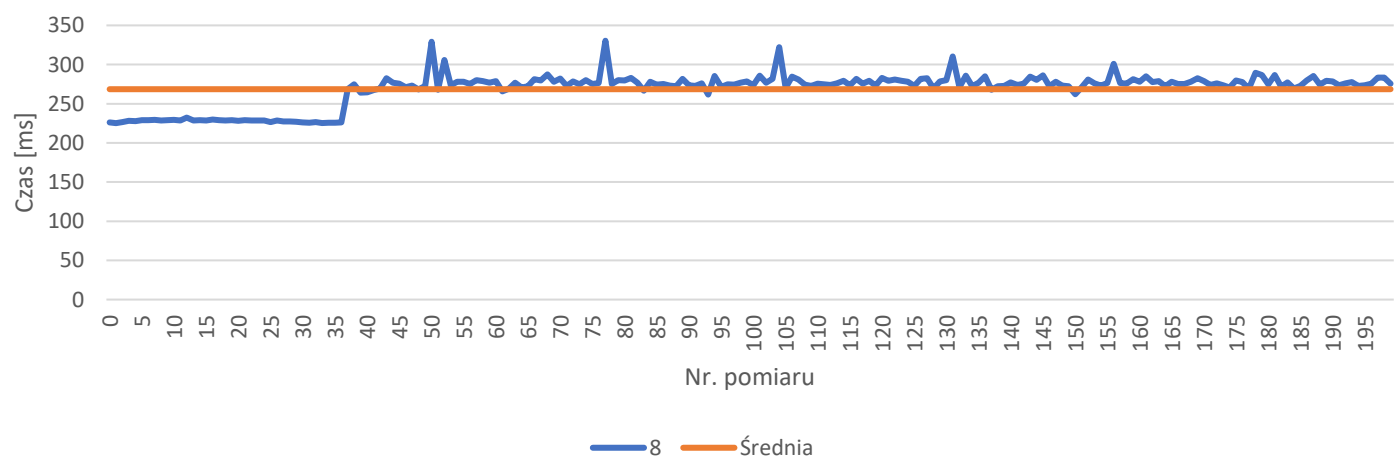
Wykres 13

Liczba wierzchołków 7 dla próby 1-10

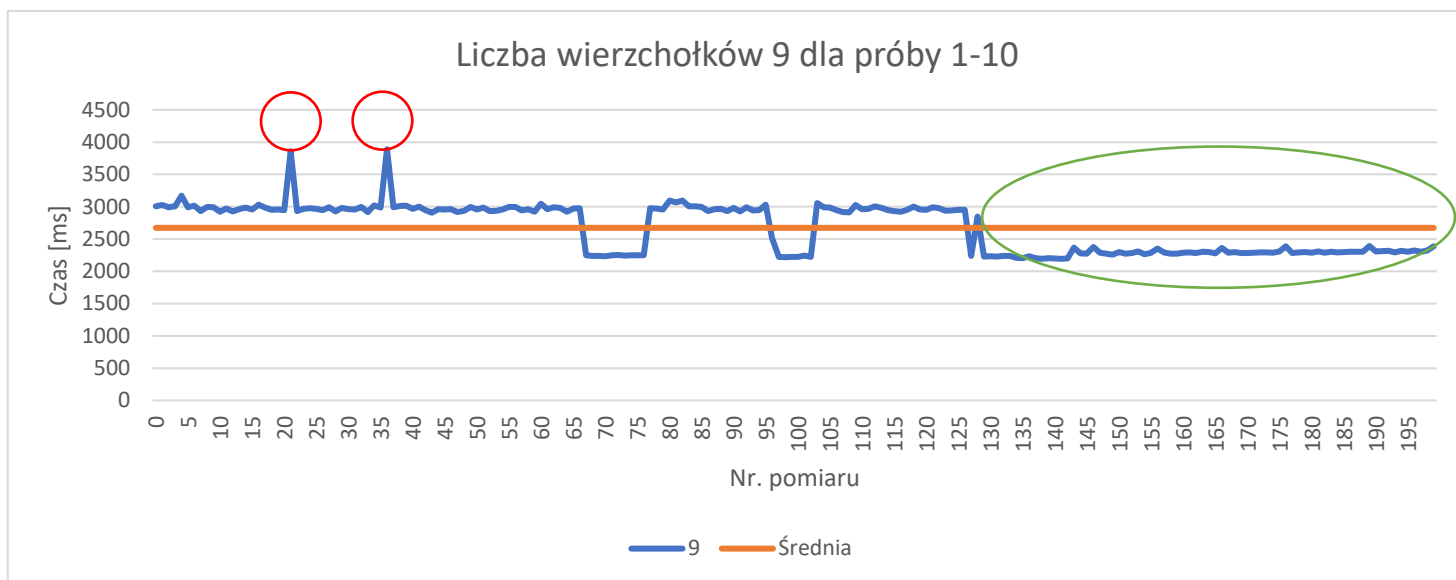


Wykres 14

Liczba wierzchołków 8 dla próby 1-10



Wykres 15



Wykres 16

Wykresy do 10-16 zawierają dane pomiarowe z pliku „BruteForceTest 3-9 x40.txt” i przedstawiają średni czas 40 pomiarów dla poszczególnych instancji czas pomiaru poszczególnych instancji do średniego czasu pomiaru wszystkich instancji.

	Liczba wierzchołków						
	3	4	5	6	7	8	9
Średnia	0,023959	0,092884	0,521334	3,289628	25,7994	268,529	2672,916
Min	0,01959	0,0819	0,466643	3,213393	25,46093	225,2071	2193,595
Max	0,04595	0,235065	0,641363	3,800105	33,26991	330,4104	3889,984
Min %	18,24%	11,83%	10,49%	2,32%	1,31%	16,13%	17,93%
Max %	91,78%	153,07%	23,02%	15,52%	28,96%	23,04%	45,53%

Tabela 4 – Przedstawia Wartości średnie, minimalne i maksymalne dla danych pomiarowych „BruteForceTest 3-9 x40.txt” i odległości wierzchołków 1-10

	Liczba wierzchołków						
	3	4	5	6	7	8	9
Średnia	0,023959	0,091498	0,521334	3,289628	25,7994	268,529	2660,8099
Min	0,01959	0,0819	0,466643	3,213393	25,46093	225,2071	2193,5948
Max	0,04595	0,12	0,641363	3,800105	33,26991	330,4104	3171,7578
Min %	18,24%	10,49%	10,49%	2,32%	1,31%	16,13%	17,56%
Max %	91,78%	31,15%	23,02%	15,52%	28,96%	23,04%	19,20%

Tabela 5 – Przedstawia Wartości średnie, minimalne i maksymalne dla danych pomiarowych „BruteForceTest 3-9 x40.txt” i odległości wierzchołków 1-10 po odrzuceniu błędów grubych zaznaczonych czerwonymi okręgami na wykresach dla wierzchołka 4, 7 i 9

7. Analiza wyników i wnioski

Analizując powyższe dane możemy wyciągnąć wiele ciekawych wniosków.

Po pierwsze możemy zauważyć, że wartość wagi pomiędzy wierzchołkami nie wpływa na działanie algorytmu Brute Force. Możemy wysnuć taki wniosek, ponieważ po nałożeniu wykresów na siebie (wykresy 1-3 i 5-7) co obrazują wykresy 4 i 8 dane niemal w 100% pokrywają się ze sobą.

Kolejnym wnioskiem jaki możemy wysunąć analizując powyższe dane jest to, że niezależnie od instancji czas wykonywania algorytmu dla tej samej liczby wierzchołków jest bardzo zbliżony. Taki wniosek można wysunąć analizując wykresy od 10 do 16.

Co bardzo ważne niezależnie czy wykonywano 40 pomiarów poszczególnych instancji czy tylko 1 pomiar pomiary były bardzo zbliżone. Jednakże dzięki takiemu podejściu w znaczącym stopniu udało się zniwelować błędy grube biorąc do analizy średnie z tych 40 pomiarów.

Jednakże wachania wciąż się pojawiały w szczególności dla bardzo małych instancji. Najprawdopodobniej spowodowane bardzo krótkim czasem obliczeniowym co możemy zaobserwować dla wszystkich instancji o 3, 4 i 5 wierzchołkach. Jest to prawdopodobnie spowodowane bardzo krótkim czasem pomiarowym poniżej 1 ms co mogło spowodować, że jakkolwiek operacja w tle mogła chwilowo obciążyć nieco bardziej procesor lub pamięć RAM i spowodować znaczący wzrost czasu. Przyczyna ta może odnosić się również do dość znaczącego odchylenia od wartości średniej w pomiarze dla 7 wierzchołków, gdzie czas pojedynczego pomiaru wynosił zaledwie kilka milisekund. Natomiast w przypadku pomiaru dla 9 wierzchołków bardziej prawdopodobną przyczyną błędu jest „throttling termiczny” procesora, który w wyniku wysokiej temperatury mógł obniżyć taktowanie zegara.

Co również można zauważyć, że procesor po wykonaniu w jednej serii pomiarów od samego początku to jest w jednym cyklu pracy programu wykonywał pomiary od 3 do 9 wierzchołków i zakończył pracę na 9 wierzchołkach dla 132 pomiaru. W dniu kolejnym kontynuowano pomiar na chłodnym procesorze przez co kolejne pomiary wykonywane były o kilkanaście procent szybciej. Co obserwujemy na wykresie 16.

Dodatkowo analizując wykresy i odrzucając błędy grube rozbieżność pomiędzy wartością średnią a skrajnymi znacząco się zmniejszyła i dla instancji powyżej 5 wierzchołków nie przekraczają nawet 30%. Co obrazują tabele 4 i 5.