Adam Dyda grupa czwartek 16:15 A

Algorytmy geometryczne laboratorium 2

1.Cel ćwiczenia

Celem ćwiczenia jest zapoznanie się z algorytmami wyznaczania otoczki wypukłej (tj. algorytmem grahama i algorytmem jarvisa), oraz porównanie czasów działania tych algorytmów.

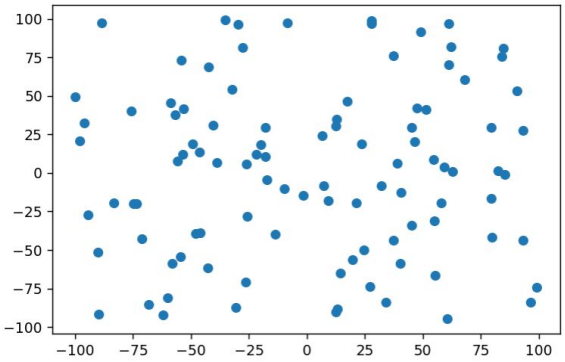
2.Wprowadzenie i przygotowanie do ćwiczenia

Do wykonania ćwiczenia wykorzystałem język python z następującymi bibliotekami, **math** - do implementacji funkcji matematycznych i losowania liczb, **numpy** - do losowania liczb, **matplotlib** - do wizualizacji otrzymanych wyników i danych. **functools** - biblioteka z narzędziami dla funkcji wyższego rzędu, użyta tutaj przy sortowaniu zbioru punktów.

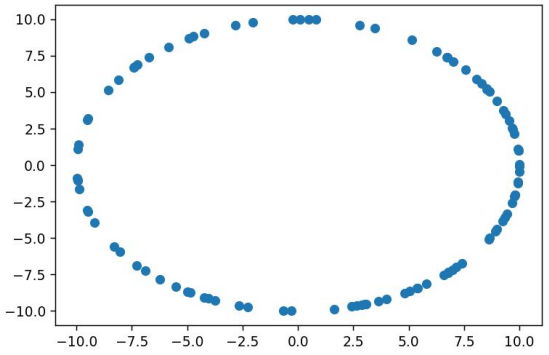
3.Wygenerowane zbiory danych na których pracowałem

zbiór A - 100 losowo wygenerowanych punktów o współrzędnych z przedziału [-100,100]
zbiór B - 100 losowo wygenerowanych punktów leżących na okręgu o środku (0,0) i promieniu R = 10
zbiór C - 100 losowo wygenerowanych punktów leżących na bokach prostokąta o wierzchołkach (-10, 10), (-10,-10), (10,-10), (10,10)
zbiór D - zawierający wierzchołki kwadratu (0, 0), (10, 0), (10, 10), (0, 10) oraz punkty wygenerowane losowo w sposób następujący: po 25 punktów na dwóch bokach kwadratu leżących na osiach i po 20 punktów na przekątnych kwadratu.

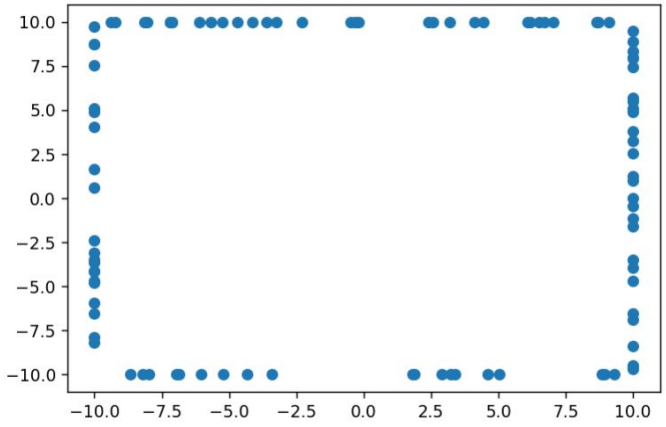
rys 1.1 punkty ze zbioru A



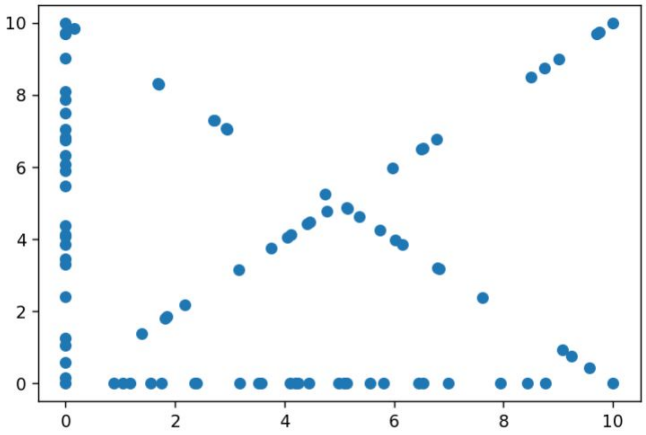
rys 1.2 punkty ze zbioru B



rys 1.3 punkty ze zbioru C



rys 1.4 punkty ze zbioru D

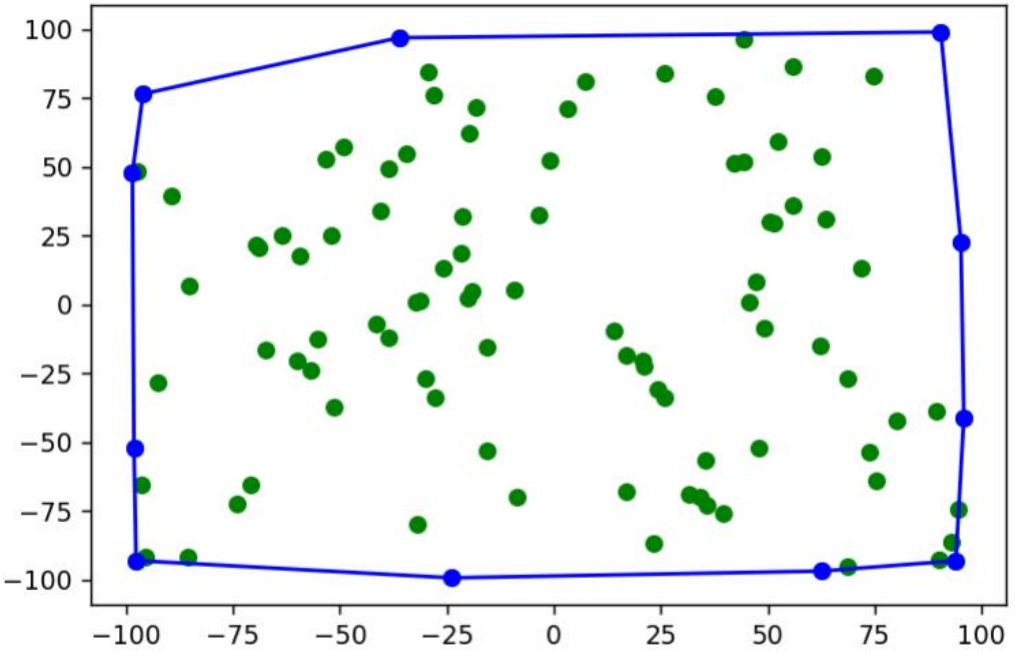


4. Prezentacja działania algorytmów dla powyższych zbiorów

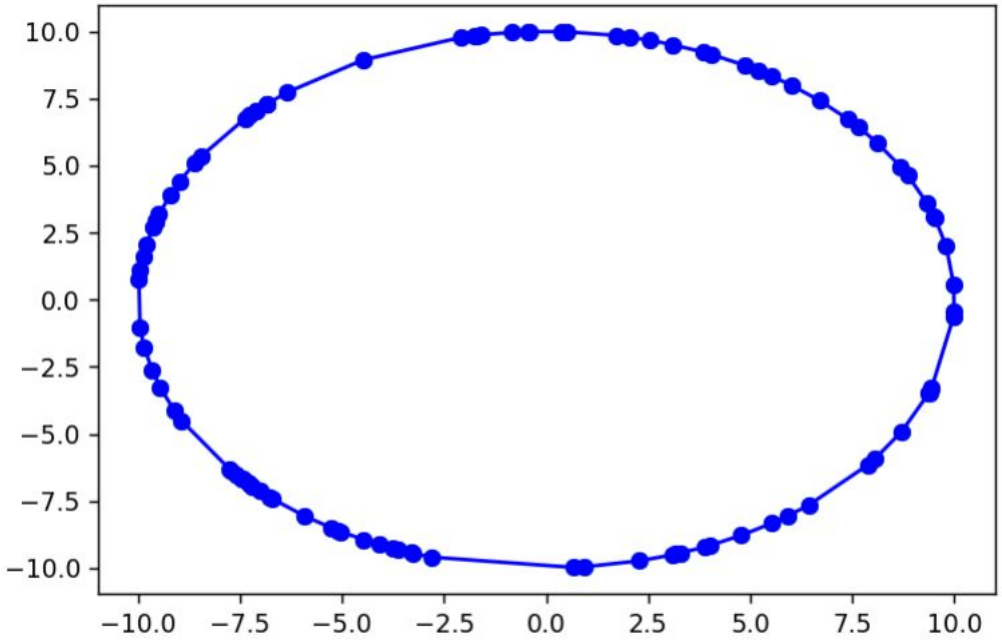
4.1 Algorytm Jarvisa

Poniższe rysunki przedstawiają otoczkę wyznaczoną na powyższych zbiorach za pomocą algorytmu Jarvisa. Niebieskim kolorem zaznaczone krawędzie otoczki

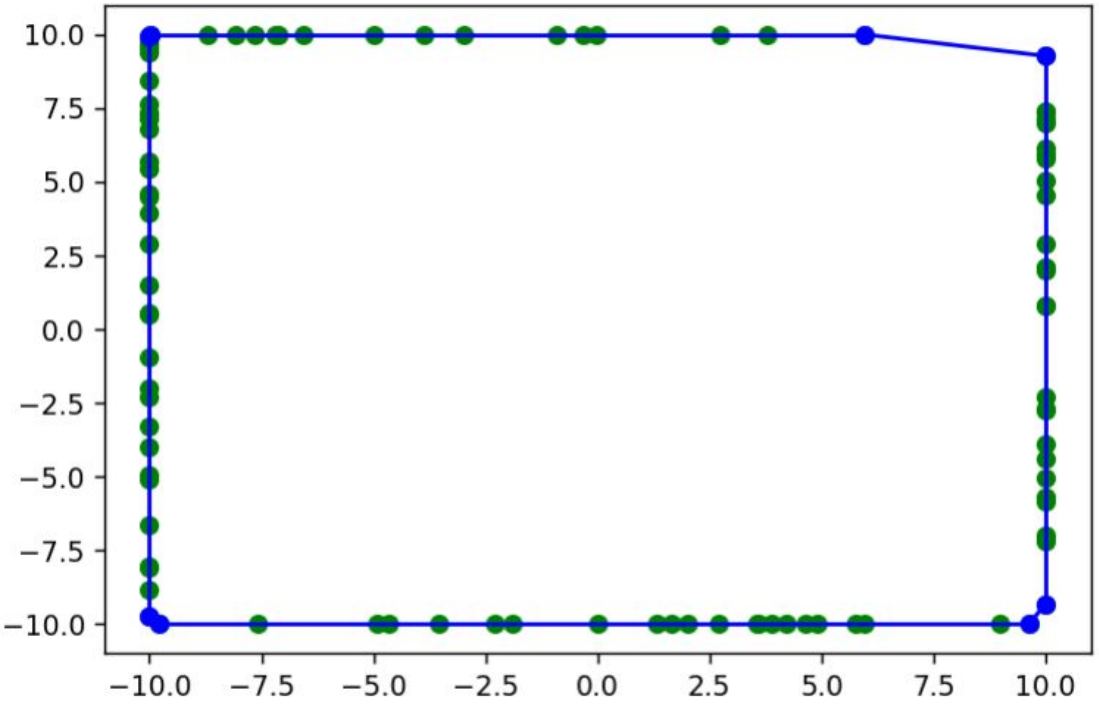
rys 2.1 działanie algorytmu Jarvisa na zbiorze A



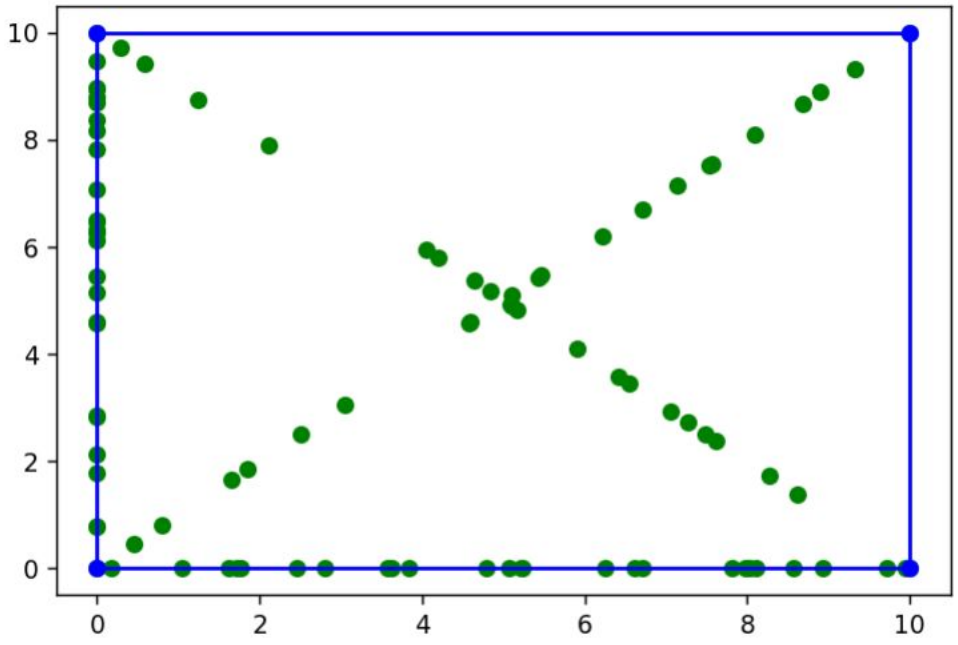
rys 2.2 działanie algorytmu Jarvisa na zbiorze B



rys 2.3 działanie algorytmu Jarvisa na zbiorze C



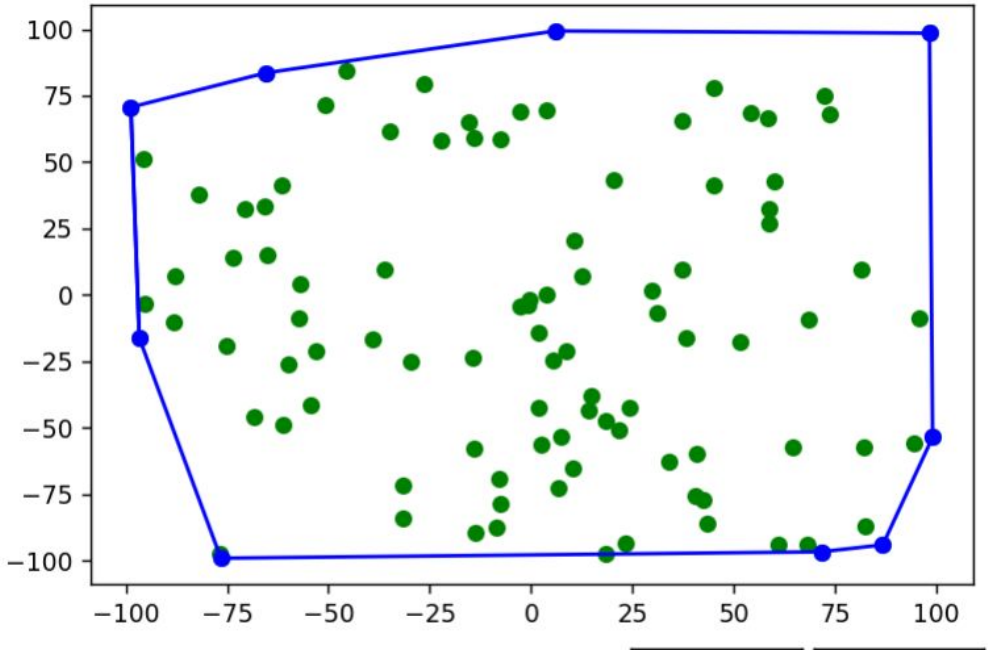
rys 2.4 działanie algorytmu Jarvisa na zbiorze D



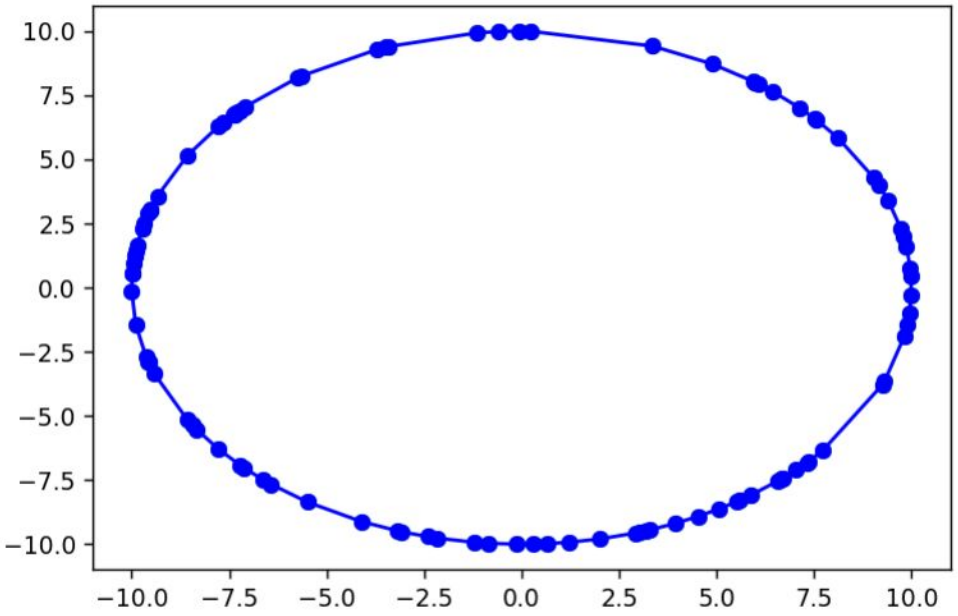
4.1 Algorytm Grahama

Poniższe rysunki przedstawiają otoczkę wyznaczoną na powyższych zbiorach za pomocą algorytmu Grahama. Niebieskim kolorem zaznaczone krawędzie otoczki.

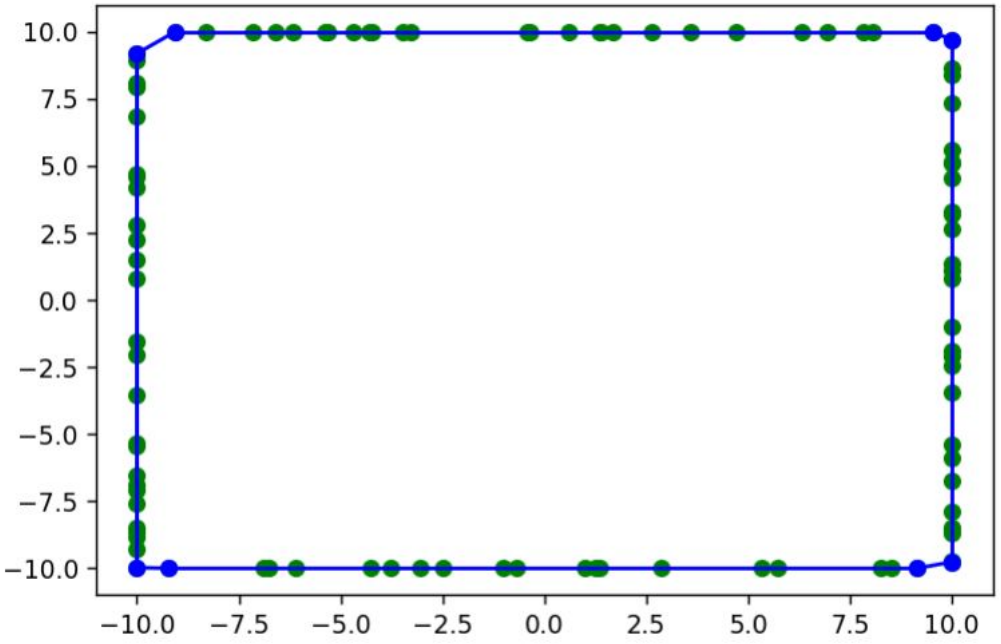
rys 2.5 działanie algorytmu Grahama na zbiorze A



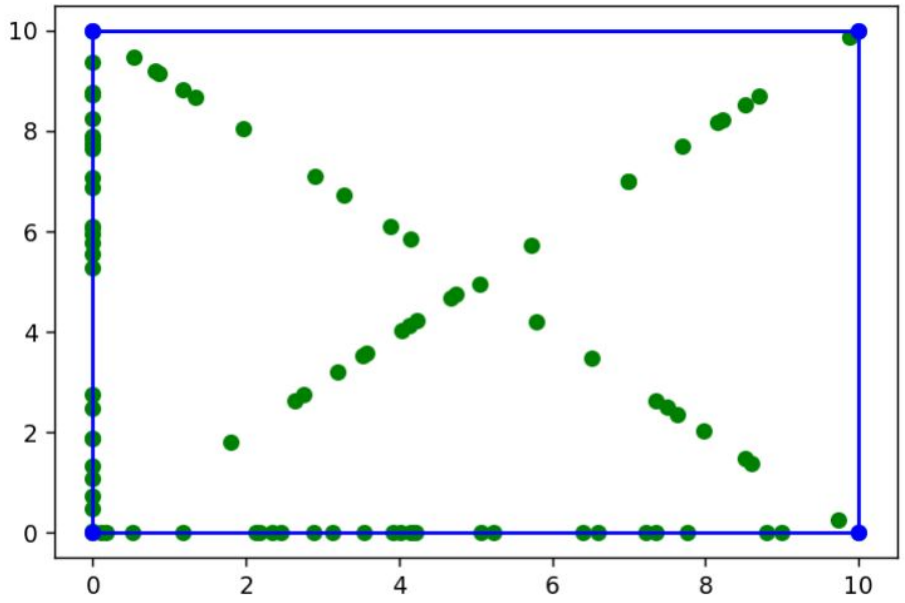
rys 2.6 działanie algorytmu Grahama na zbiorze B



rys 2.7 działanie algorytmu Grahama na zbiorze C



rys 2.8 działanie algorytmu Grahama na zbiorze D



5. Porównanie szybkości algorytmów na zbiorach o różnej wielkości

Jedynym parametrem który był modyfikowany w teście szybkości była ilość punktów w zbiorach. Czas był mierzony bez rysowania zbiorów aby uzyskać wyniki jak najbliższe rzeczywistej szybkości algorytmów.

5.1 Porównanie algorytmów na zbiorze A

tabela 1.1 porównania czasu działania algorytmów na zbiorze A

Ilość punktów w zbiorze	czas dla algorytmu Jarvisa [s]	czas dla algorytmu Grahama [s]
100	0.002001047134399414	0.0020194053649902344
1000	0.023998737335205078	0.02700519561767578
10000	0.19298529624938965	0.17099738121032715
100000	3.2080249786376953	2.489002227783203
500000	15.318148136138916	16.409810304641724
1000000	45.89803457260132	32.6230046749115

5.2 Porównanie algorytmów na zbiorze B

tabela 1.2 porównania czasu działania algorytmów na zbiorze B

Ilość punktów w zbiorze	czas dla algorytmu Jarvisa [s]	czas dla algorytmu Grahama [s]
100	0.014998435974121094	0.0029969215393066406
1000	0.7980203628540039	0.02097797393798828
10000	111.21293234825134	0.2479875087738037
100000	> 3 h *	2.3580055236816406
500000	> 3 dni *	15.046174049377441
1000000	> 1 tydz *	137.30884265899658

***** czasy szacowane za pomocą złożoności obliczeniowej

5.3 Porównanie algorytmów na zbiorze C

tabela 1.3 porównania czasu działania algorytmów na zbiorze C

Ilość punktów w zbiorze	czas dla algorytmu Jarvisa [s]	czas dla algorytmu Grahama [s]
100	0.0010018348693847656	0.0019996166229248047
1000	0.01600027084350586	0.028021812438964844
10000	0.0907280445098877	0.2100205421447754
100000	0.9450232982635498	3.22998046875
500000	5.903012275695801	31.04051423072815
1000000	12.922511100769043	102.02523350715637

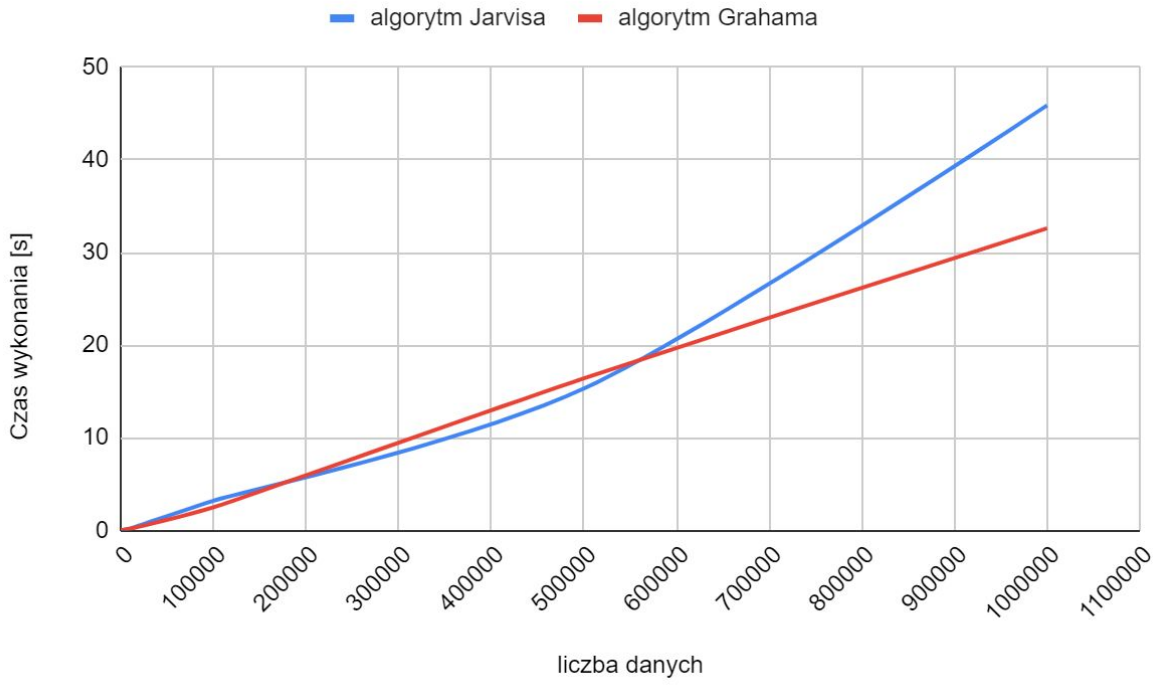
5.4 Porównanie algorytmów na zbiorze D

tabela 1.4 porównania czasu działania algorytmów na zbiorze D

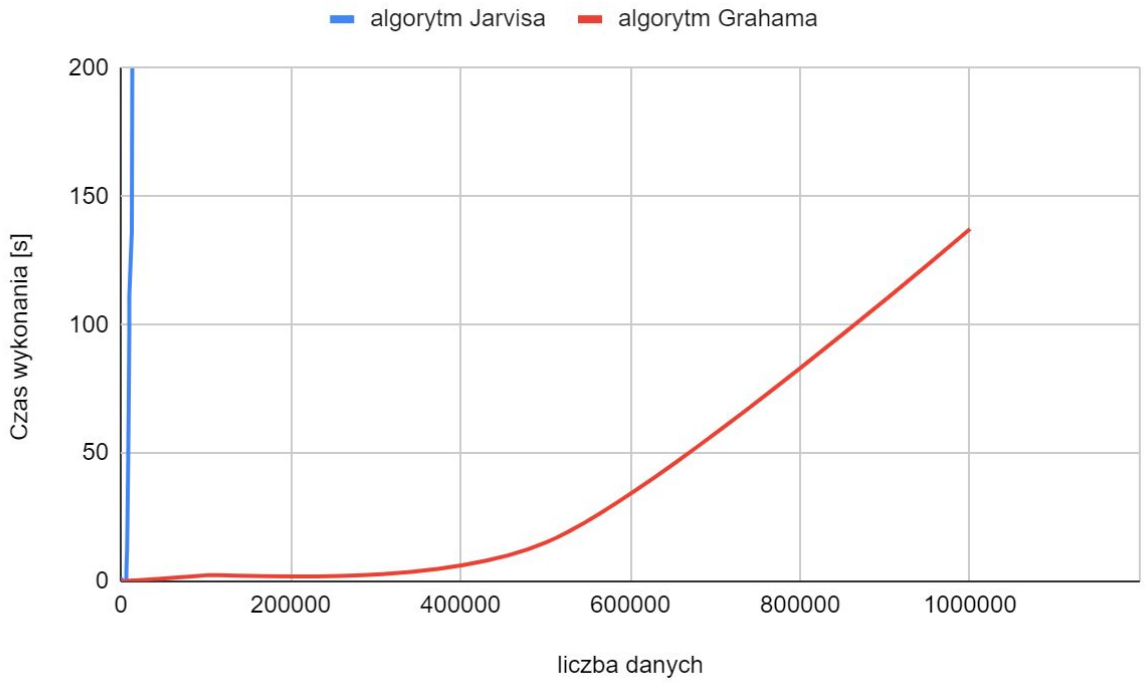
Ilość punktów na bokach	Ilość punktów na przekątnych	czas dla algorytmu Jarvisa [s]	czas dla algorytmu Grahama [s]
25	25	0.0	0.00199317932128906
250	250	0.00800156593322753	0.01600170135498047
2500	2500	0.06097579002380371	0.10617685317993164
25000	25000	0.48302221298217773	1.807999849319458
2500000	250000	5.321664810180664	122.89200735092163

6. Ilustracja czasów wykonania algorytmów

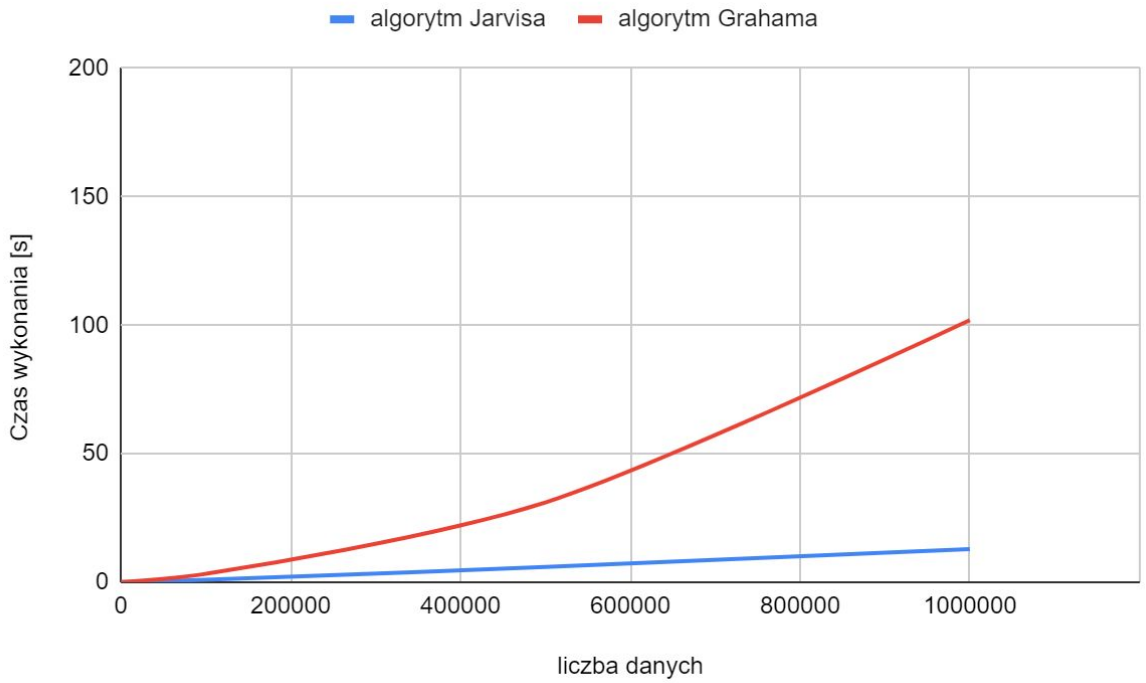
rys 3.1 wykres czasów wykonania od ilości punktów, dla zbioru A



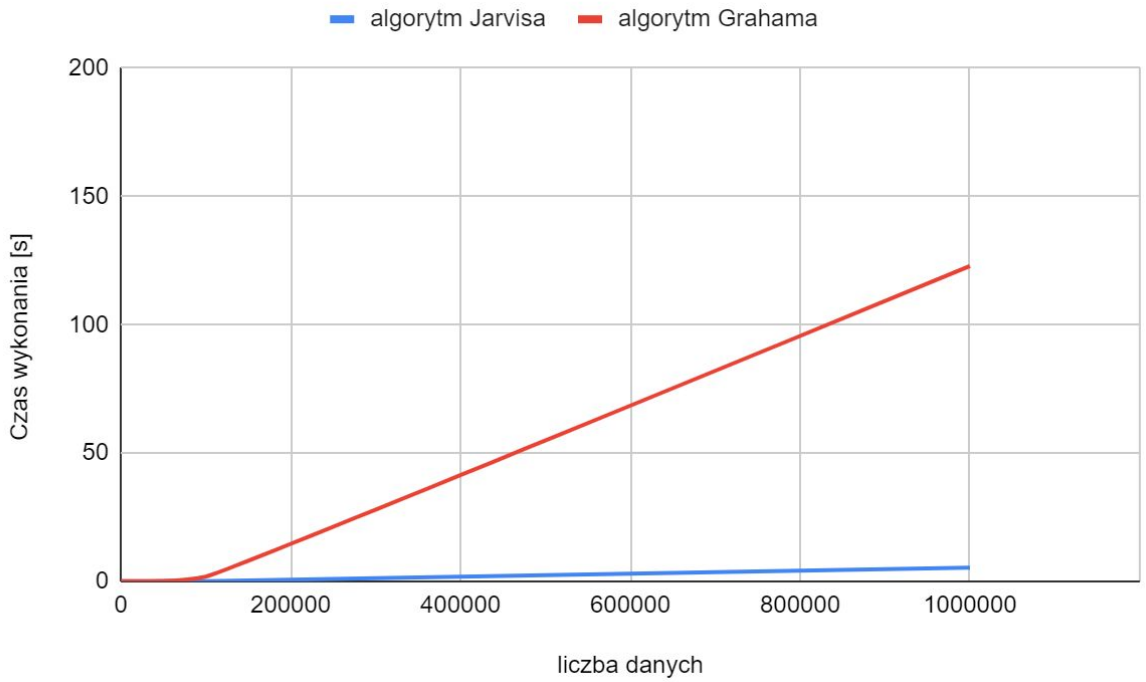
rys 3.2 wykres czasów wykonania od ilości punktów, dla zbioru B



rys 3.3 wykres czasów wykonania od ilości punktów, dla zbioru C



rys 3.4 wykres czasów wykonania od ilości punktów, dla zbioru D



7. Wnioski

Z uzyskanych wyników wynika to czego mogliśmy się spodziewać, algorytm Grahama radzi sobie lepiej dla dużych zbiorów danych w których punkty rozmieszczone są losowo (u jest jest to zbiór A). Wynika to oczywiście ze złożoności obliczeniowej algorytmu Grahama która wynosi O(n log n) w porównaniu do złożoności algorytmu Jarvisa O(n*k). Jednak w zbiorach w których mamy bardzo mało punktów należących do otoczki w porównaniu do wszystkich punktów zbioru (u nas są to zbiory C i D) to algorytm Jarvisa jest dużo szybszy, wynika to z czynnika k w złożoności obliczeniowej tego algorytmu. Warto też zwrócić uwagę na wyniki dla zbioru B jest to zbiór w którym wszystkie punkty należą do otoczki, wynikiem tego jest ogromny czas wykonania algorytmu Jarvisa.