

Patryk Studziński

Algorytmy geometryczne

Grupa: czwartek-16:15B

Ćwiczenie nr. 2

# Otoczka wypukła

## 1. Cel ćwiczenia

Wyznaczenie otoczki wypukłej danych zbiorów punktów za pomocą algorytmów Grahama i Jarvisa oraz graficzna prezentacja działania algorytmów.

## 2. Dane testowe

Dane i obliczenia zostały wykonane na:

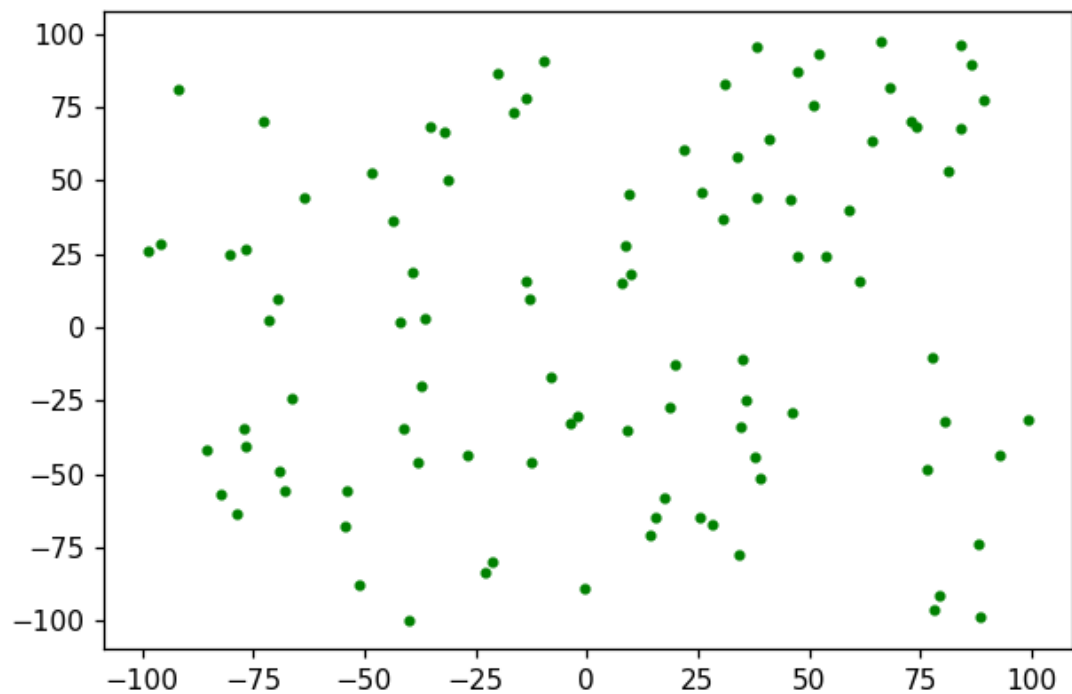
System operacyjny: 64 bitowy system operacyjny Win 10 Home

Procesor: Intel x64

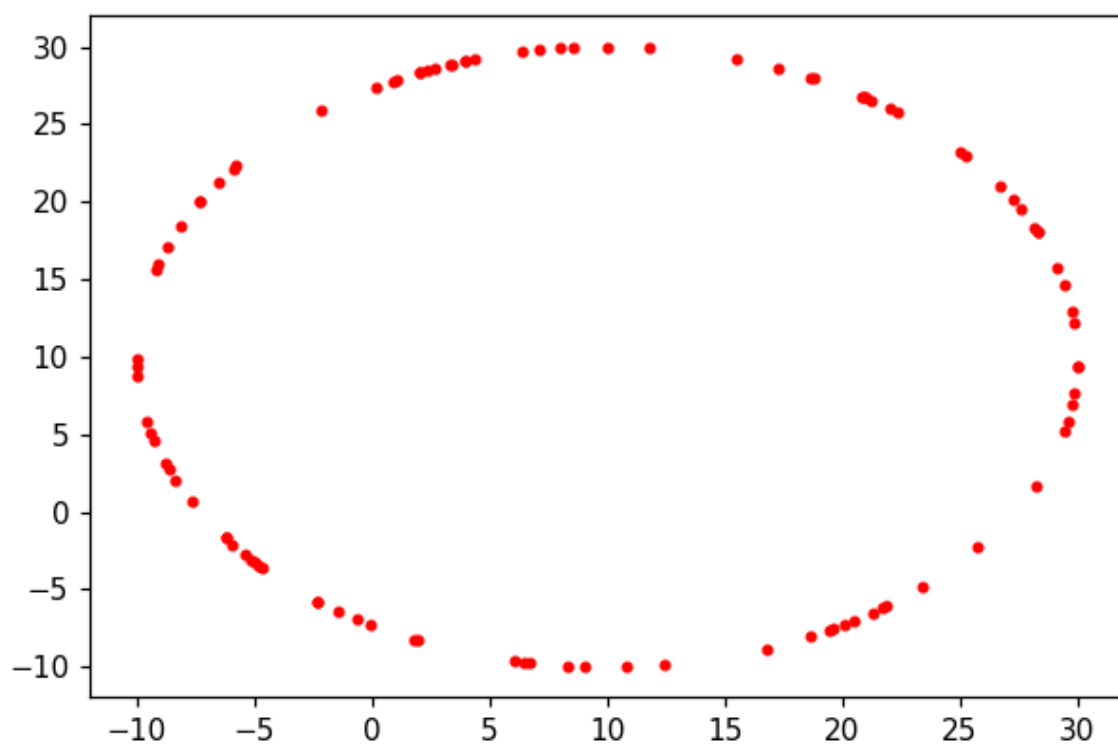
- Zbiór A zawierający losowo wygenerowane punkty o współrzędnych z określonego przedziału
- Zbiór B zawierający losowo wygenerowane punkty leżących na okręgu o zadanym środku i promieniu
- Zbiór C zawierający losowo wygenerowane punkty leżące na bokach prostokąta o zadanych wierzchołkach
- Zbiór D zawierający wierzchołki kwadratu oraz punkty wygenerowane na przekątnych i bokach kwadratu leżących na osiach

### 3. Graficzna prezentacja zbiorów

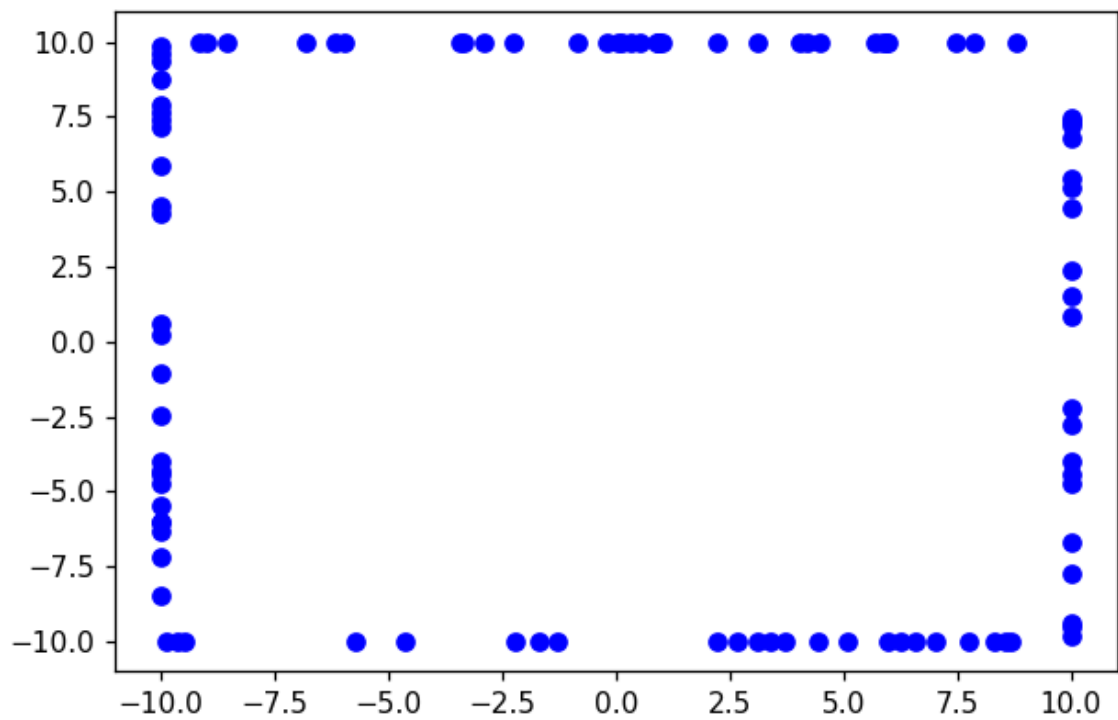
Zbiór A:



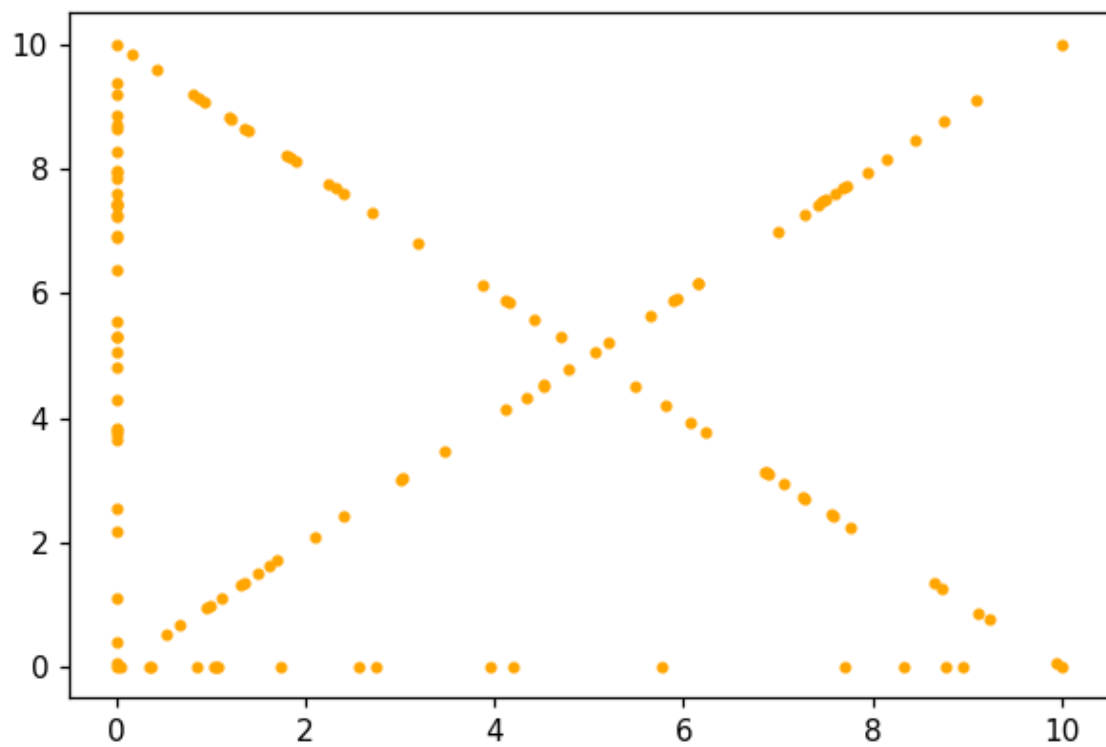
Zbiór B:



Zbiór C:



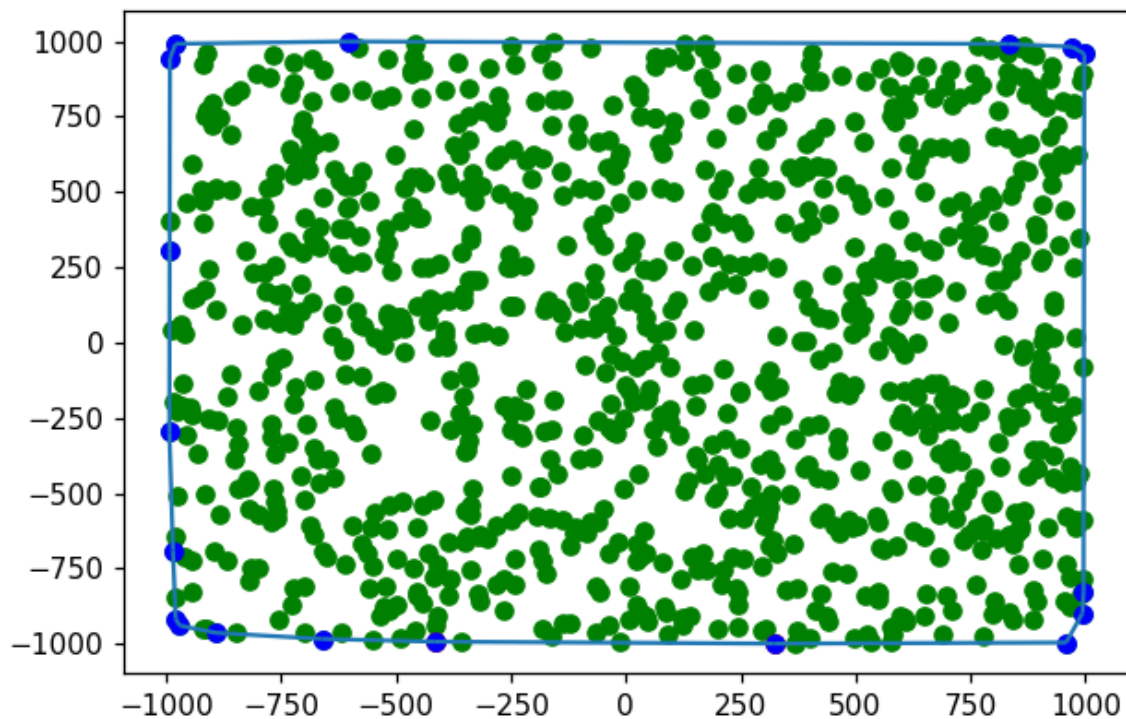
Zbiór D:



## 4. Wyniki działania programu

Zbiór A zawiera najogólniejsze przypadki do testowania, znajdowało się w nim mało współliniowych punktów. Algorytm Grahama wyznaczał otoczkę szybciej na tym zbiorze, co pokazuje poniższa tabela.

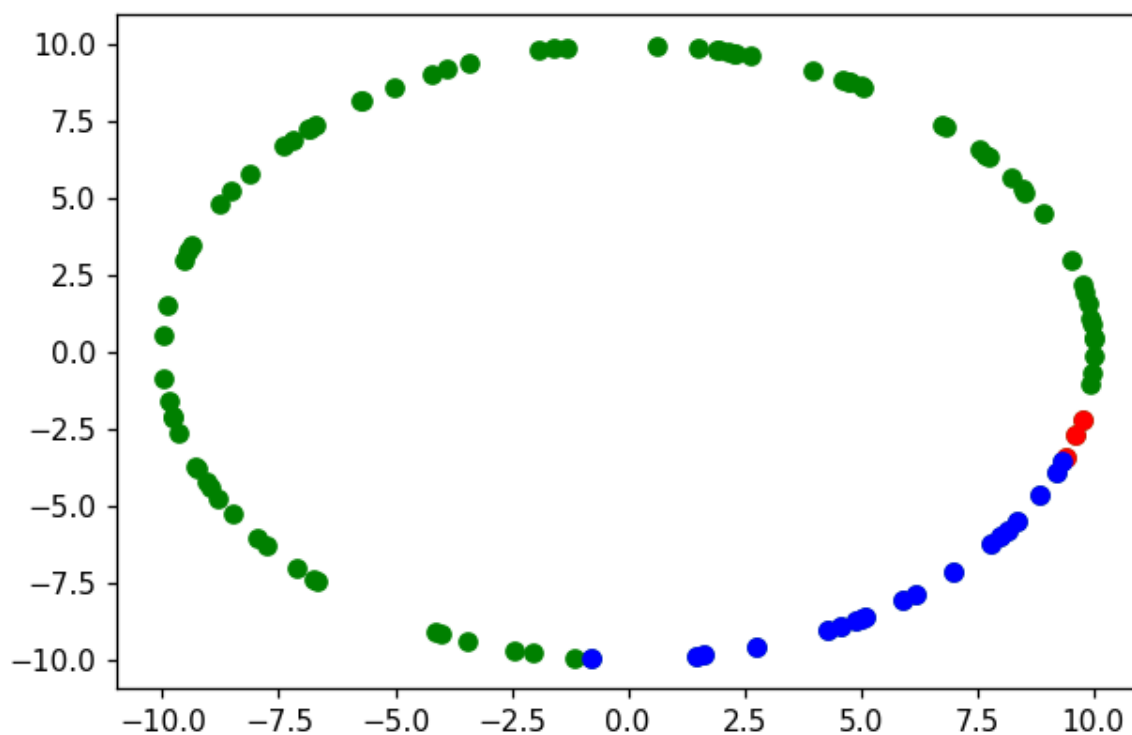
	Ilość punktów	czas Graham	Czas Jarvis
Zbiór A	100	0,99 ms	0,99 ms
	1000	11,97ms	16,98ms
	10000	145,59ms	207,45ms



Otoczka wypukła zbioru A wyznaczona za pomocą algorytmu Jarvisa.

W zbiorze B algorytmy musiały poradzić sobie z punktami położonymi na okręgu, więc wszystkie punkty wchodziły w skład otoczki. Tutaj zdecydowanie szybciej zadziałał algorytm Grahama, gdyż jest to pesymistyczny przypadek dla algorytmu Jarvisa, który działa tutaj w czasie  $n^2$ )

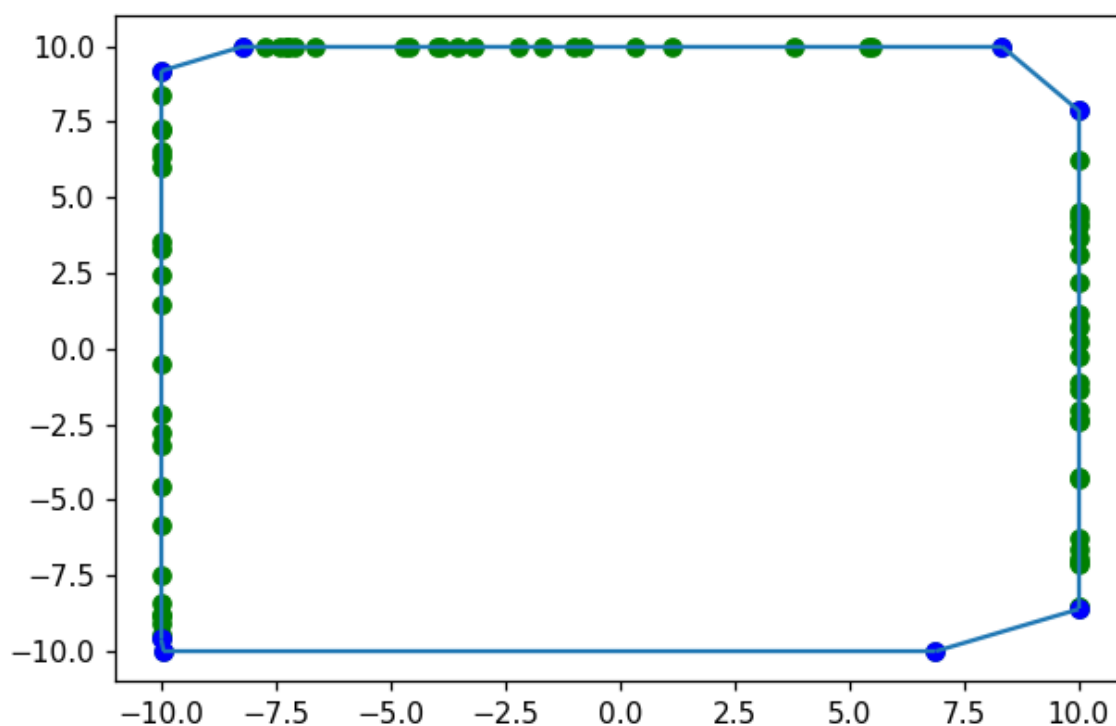
	Ilość punktów	czas Graham	Czas Jarvis
Zbiór B	100	0,99 ms	11,96 ms
	1000	10,97ms	962,60ms
	3000	37,41ms	8796,4ms



Tworzenie otoczki za pomocą algorytmu Grahama (punkty czerwone – aktualnie rozpatrywane, niebieskie – aktualnie w otoczce, zielone – pozostałe).

Zbiór C zawierał dużo punktów współliniowych, dlatego przy zbyt małej tolerancji dla 0 algorytmy nie działały poprawnie. Z racji na mniejszą liczbę punktów w otoczce algorytm Jarvisa działał w tym przykładzie szybciej.

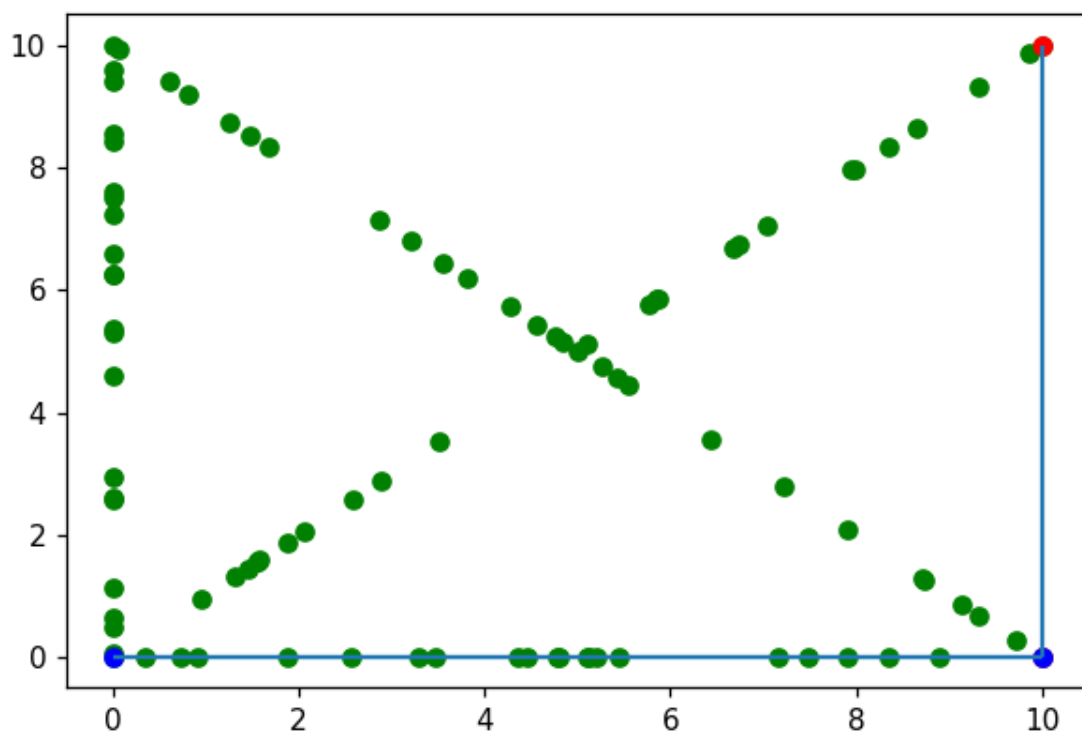
	Ilość punktów	czas Graham	Czas Jarvis
Zbiór C	100	0,99 ms	0,99 ms
	5000	68,8ms	52,8ms
	20000	295,41ms	209,44ms



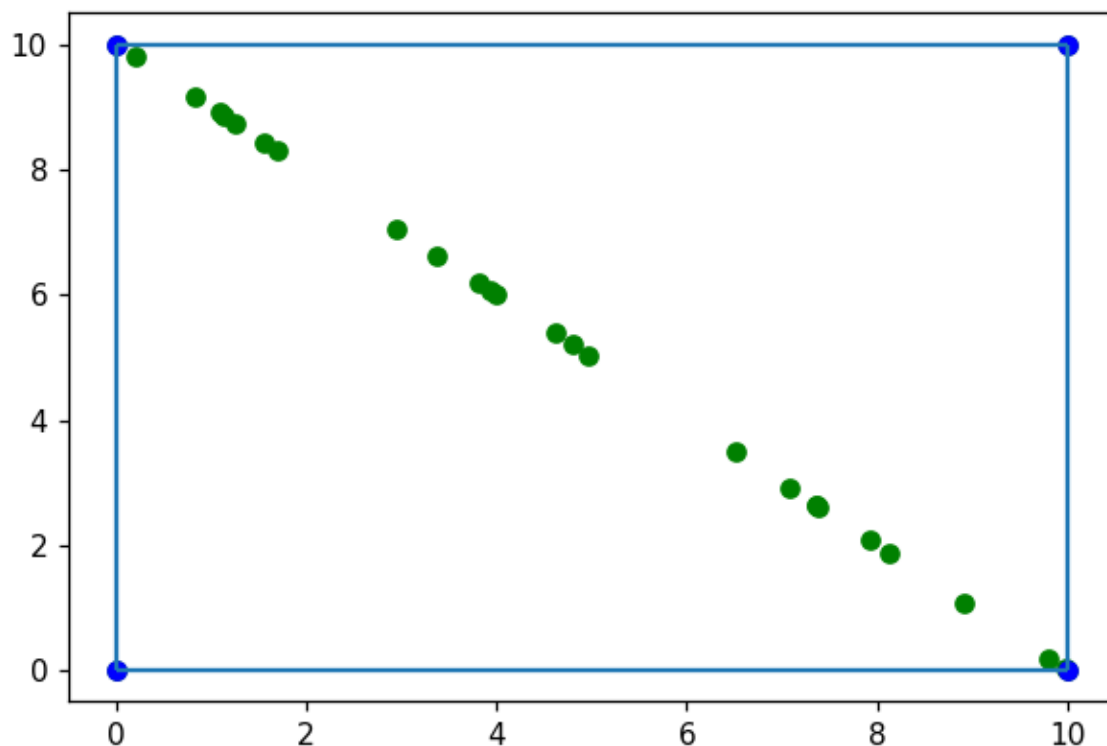
Otoczka wyznaczona przez algorytm Grahama dla zbioru C

Ostatni zbiór również zawierał dużą liczbę współliniowych punktów, z którymi algorytmy miały problemy przy małych tolerancjach  $\epsilon$ . Punkty w zbiorze D tworzą kwadrat, dlatego otoczka zawiera tylko 4 punkty. Zgodnie z oczekiwaniami tutaj ponownie lepiej poradził sobie algorytm Jarvisa, który szybciej wyznacza otoczkę.

	Ilość punktów	czas Graham	Czas Jarvis
Zbiór D	100	0,99ms	0,0 ms
	1000	6,01ms	4,95ms
	10000	66,07ms	53,85ms



Wyznaczanie otoczki wypukłej zbioru D za pomocą algorytmu Jarvisa.



Otoczka wypukła zbioru D wyznaczona algorytmem Grahama.

## 5. Wnioski

Zbiory punktów zostały dobrane tak, aby pokazać wady, zalety i różnice pomiędzy algorytmami. Algorytm Jarvisa działa zdecydowanie lepiej gdy otoczkę tworzy mała ilość punktów, niestety w odwrotnym przypadku, gdy otoczkę tworzą wszystkie lub duża ilość punktów działa znacznie wolniej. Drugi z algorytmów dla wszystkich danych działa bardzo podobnie, dlatego w większości przypadków, gdzie punkty będą rozmieszczone losowo, lepiej sprawdzi się algorytm Grahama, gdyż nie ma on pesymistycznych przypadków.