17.11.2023 r.

Mateusz Król

Otoczka wypukła

Ćwiczenie nr 2

Spis treści

1. Realizacja ćwiczenia

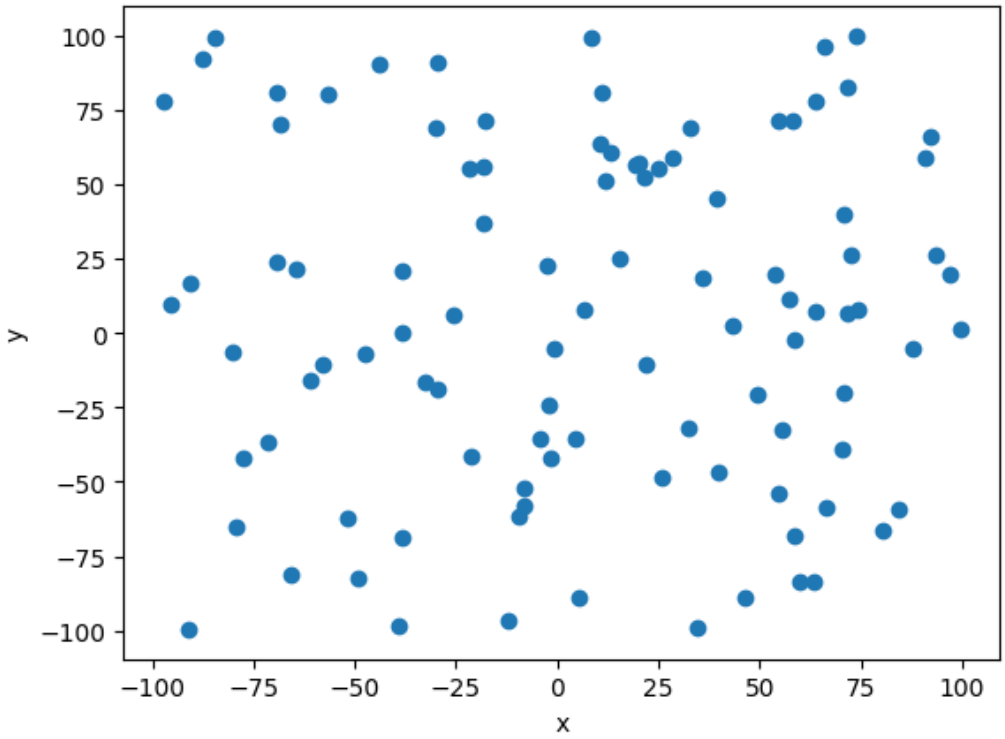
W ramach ćwiczenia drugiego, generuję różne typy zbiorów punktów o różnym rozmiarze, w celu przetestowania mojej implementacji algorytmów Grahama i Jarvisa.

Wykorzystuję bibliotekę **time**, żeby zmierzyć czasy działania algorytmów na różnych zbiorach danych.

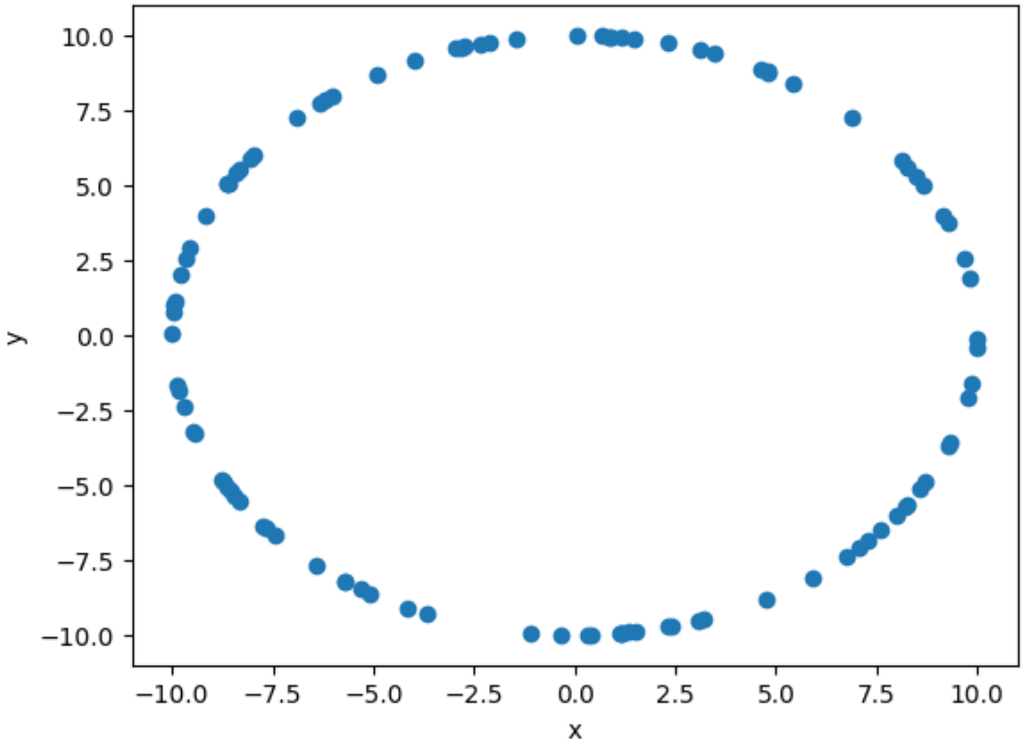
Pomiary czasowe będę przeprowadzał na zbiorach podstawowych zamieszczonych w poleceniu z laboratorium oraz na własnoręcznie wygenerowanych zbiorach o zdecydowanie większej ilości punktów.

Zbiory podstawowe:

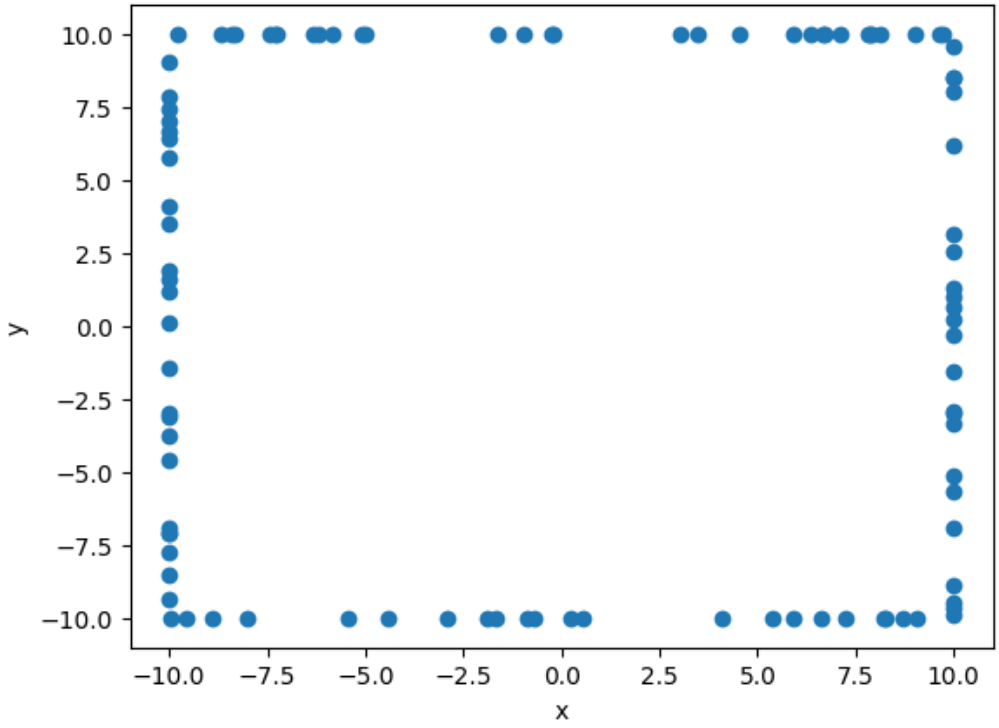
- Zbiór A: wygenerowane losowo punkty na całej powierzchni zadanego prostokąta – na iloczynie kartezjańskim dwóch przedziałów



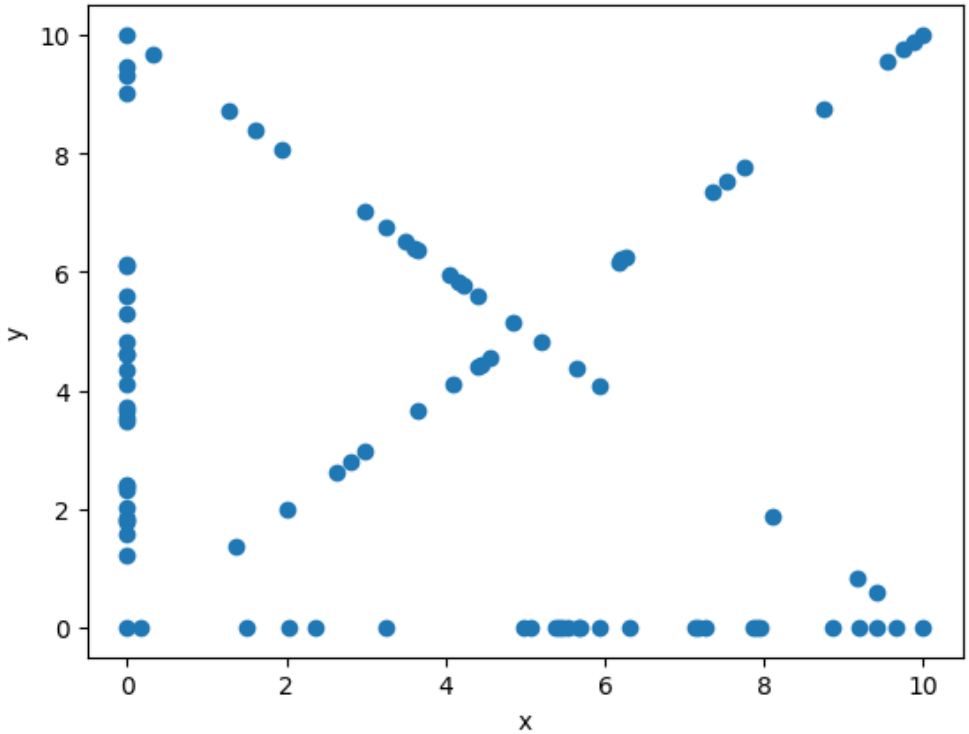
- Zbiór B: wygenerowane losowo punkty leżące na okręgu



- Zbiór C: wygenerowane losowo punkty leżące na obwodzie prostokąta



- Zbiór D: punkty w wierzchołkach kwadratu wraz z wygenerowanymi losowo punktami leżącymi na dwóch z boków kwadratu i na jego przekątnych



2. Wyniki i ich analiza

2.1 Dla zbiorów punktów podstawowych obserwujemy następujące wyniki:

Algorytm	Zbiór A	Zbiór B	Zbiór C	Zbiór D
Grahama	0.001s	0.0s	0.0s	0.0s
Jarvisa	0.0s	0.006s	0.0s	0.001s

2.2 Dla zbiorów o zmiennej ilości generowanych punktów

- Zbiór A: wygenerowane losowo 10^n punktów na przedziale [-1000; 1000]
- Zbiór B: wygenerowane losowo 100n punktów leżących na okręgu o promieniu 100
- Zbiór C: wygenerowane losowo 10^n punktów leżących na obwodzie prostokąta o wymiarach 2000 na 2000
- Zbiór D: punkty w wierzchołkach kwadratu wraz z wygenerowanymi losowo 10^n punktami leżącymi na dwóch z boków kwadratu i 10^n punktów leżących na przekątnych kwadratu o wymiarach 1000 na 1000

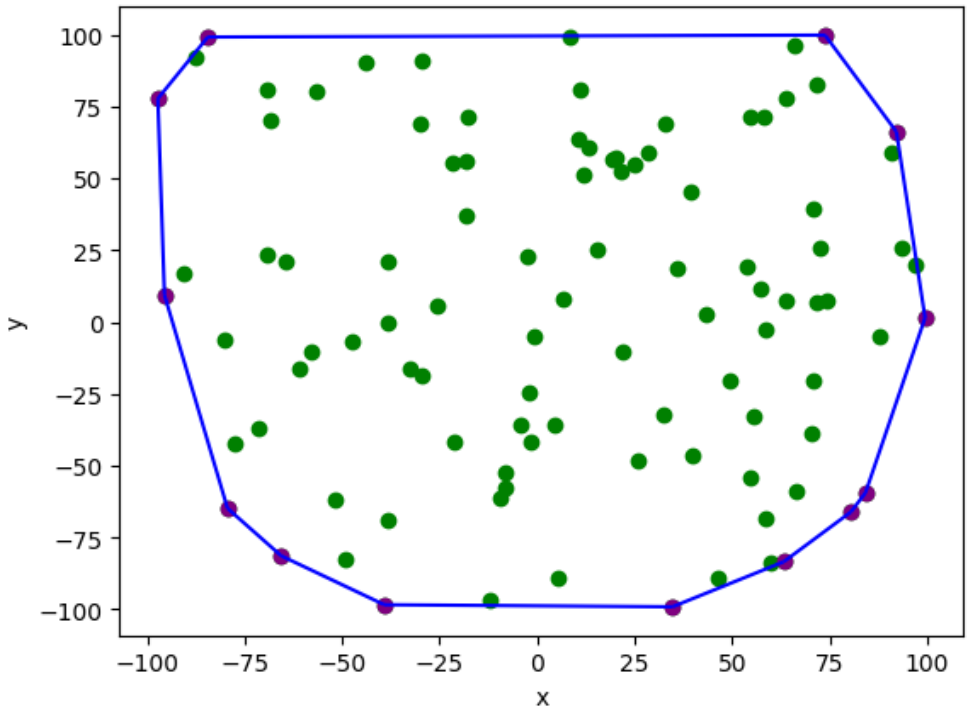
	Algorytm Grahama				Algorytm Jarvisa			
n	Zbiór A	Zbiór B	Zbiór C	Zbiór D	Zbiór A	Zbiór B	Zbiór C	Zbiór D
3	0.004s	0.001s	0.0s	0.015s	0.009s	0.09s	0.0s	0.015s
4	0.046s	0.004s	0.014s	0.131s	0.113s	0.161s	0.002s	0.179s
5	0.622s	0.0s	0.288s	1.929s	2.629s	0.265s	0.017s	1.826s
6	8.299s	0.003s	2.029s	18.408s	16.159s	0.365s	0.147s	15.106s

.

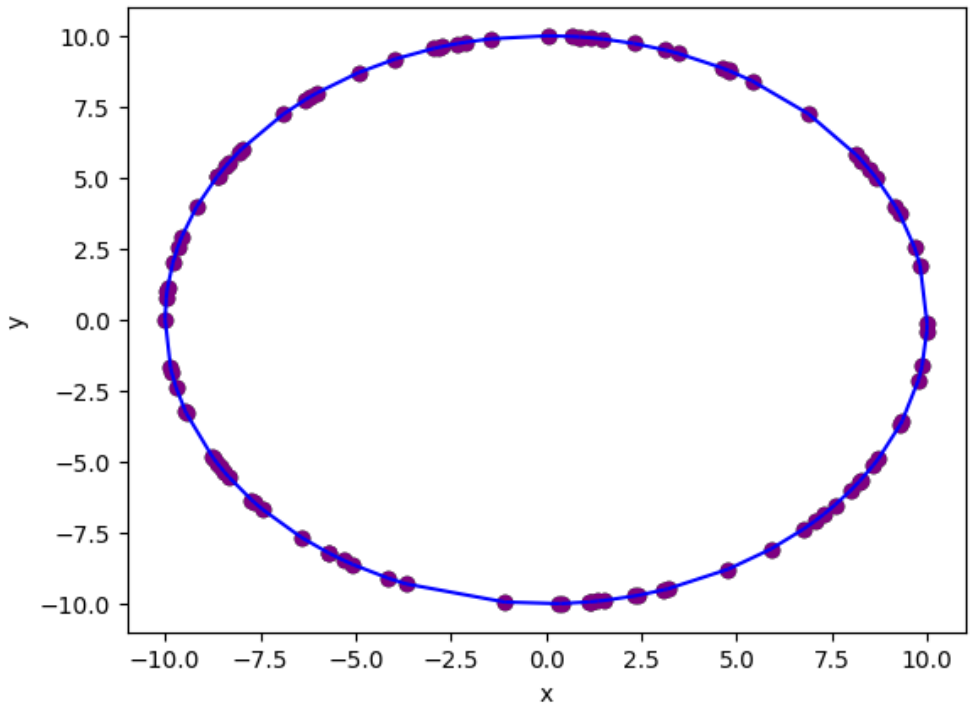
Algorytm Grahama jest szybszy dla zbiorów A i B, ze względu na dużą ilość punktów, które trafiają do otoczki, algorytm Jarvisa ma złożoność $O(n^2)$. Natomiast dla zbiorów C i D, algorytm Jarvisa jest generalnie szybszy – ilość wierzchołków, które znajdą się w otoczce wypukłej jest o wiele mniejsza niż n, więc algorytm Jarvisa ma złożoność liniową $O(k \cdot n)$, gdzie k to ilość wierzchołków w otoczce wypukłej.

3. Graficzne przedstawienie otoczki wypukłej

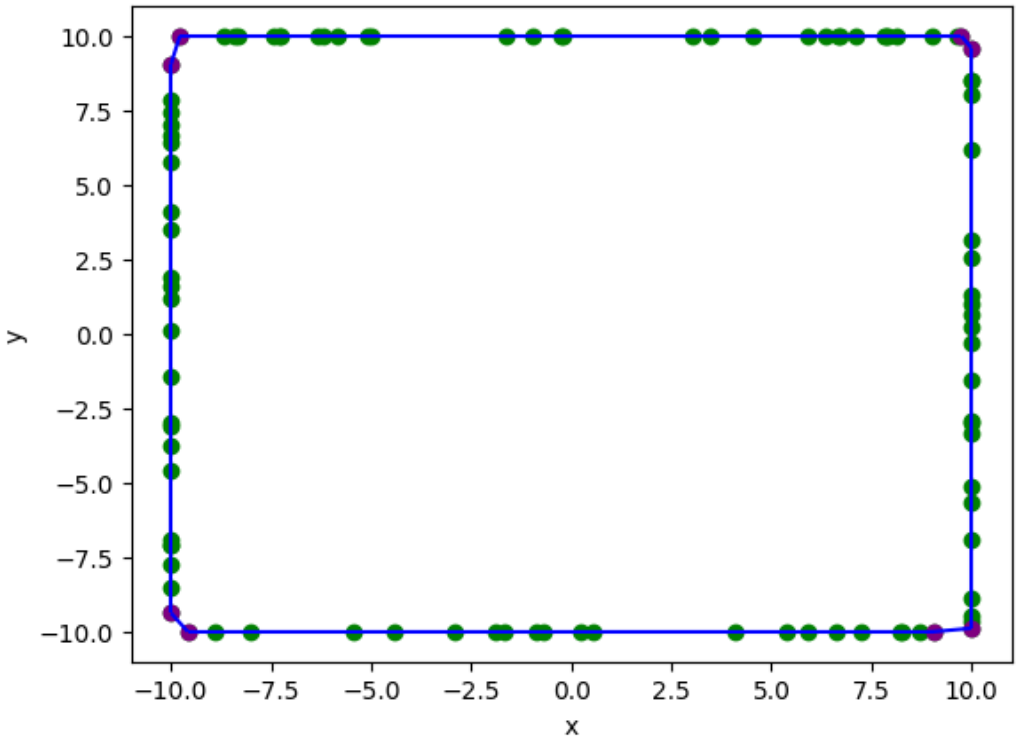
- Zbiór A:



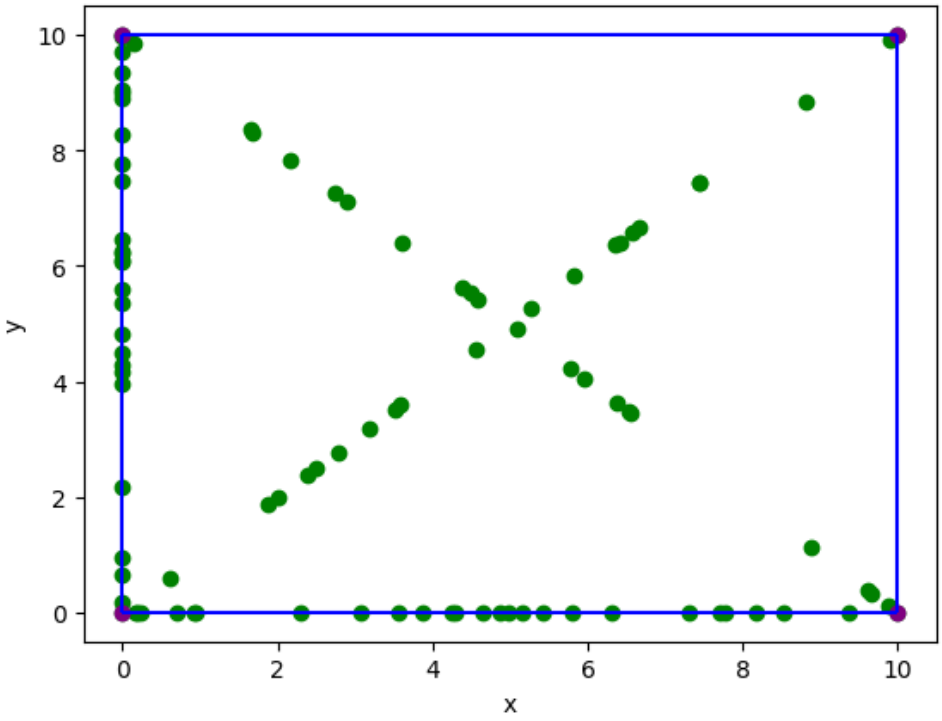
- Zbiór B:



- Zbiór C:



- Zbiór D:



4. Wnioski

Zaimplementowane przeze mnie algorytmy przechodzą wszystkie testy zadane w pliku z laboratorium. Na podstawie animacji działania algorytmu dla różnych zbiorów danych, jestem w stanie stwierdzić, że algorytm funkcjonuje poprawnie.

Zaproponowane zbiory punktów bardzo dobrze pokazują, że w różnych przypadkach opłaca się wykorzystywać różne algorytmy obliczające otoczkę wypukłą – algorytm Jarvisa czasami jest o wiele wolniejszy od algorytmy Grahama, ale dla zbiorów, dla których wiemy, że otoczka wypukła będzie składać się z mocno ograniczonej ilości punktów, algorytm Jarvisa okazuje się być o wiele lepszym wyborem.

Największy problem w działania algorytmów sprawiał zbiór B, przy testowaniu działania algorytmu Jarvisa – ze względu na to, że otoczka wypukła składa się ze wszystkich punktów – złożoność algorytmu wynosi $O(n^2)$