Laboratorium 2 - Otoczka wypukła

Mateusz Podmokły

18 październik 2023

1 Specyfikacja użytego środowiska

Specyfikacja:

- Środowisko: Jupyter Notebook,
- Język programowania: Python,
- System operacyjny: Microsoft Windows 11,
- Architektura systemu: x64.

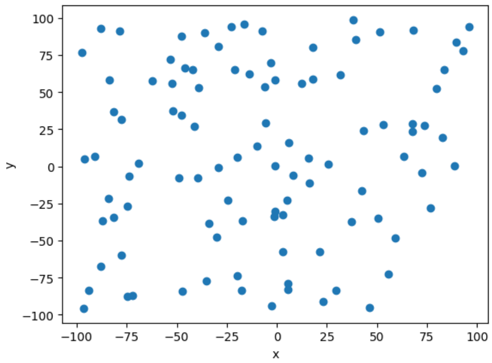
2 Przebieg ćwiczenia

Ćwiczenie polega na zaimplementowaniu algorytmów Grahama i Jarvisa obliczających otoczkę wypukłą oraz na analizie ich wyników.

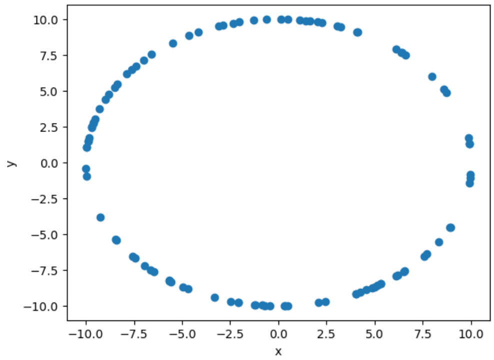
2.1 Losowanie punktów

Wylosowane zostały następujące zbiory punktów:

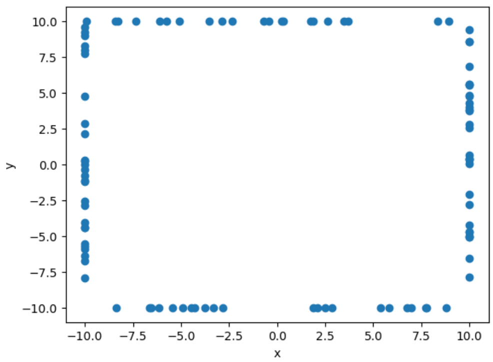
1. losowo wygenerowane punkty o współrzędnych z przedziału $[-100, 100]$,
2. losowo wygenerowane punkty leżące na okręgu o środku $(0, 0)$ i promieniu $R = 100$,
3. losowo wygenerowane punkty leżące na bokach prostokąta o wierzchołkach $(-10, 10)$, $(-10, -10)$, $(10, -10)$, $(10, 10)$,
4. punkty na wierzchołkach kwadratu $(0, 0)$, $(10, 0)$, $(10, 10)$, $(0, 10)$ oraz punkty wygenerowane losowo na dwóch bokach kwadratu leżących na osiach i na przekątnych kwadratu.



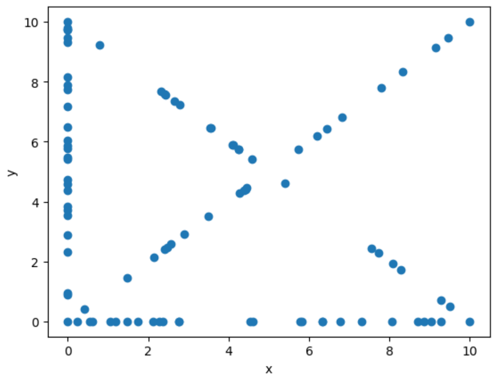
Rysunek 1: Zbiór 1. - obszar kwadratowy.



Rysunek 2: Zbiór 2. - okrąg.



Rysunek 3: Zbiór 3. - boki prostokąta.



Rysunek 4: Zbiór 4. - wierzchołki, boki i przekątne kwadratu.

2.2 Znajdowanie otoczki wypukłej

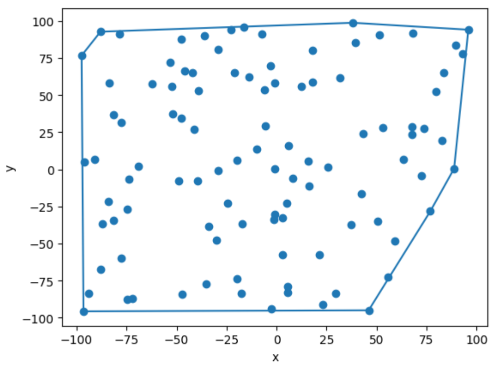
2.2.1 Wykorzystane algorytmy i rodzaje pomiarów

Dla każdego zbioru obliczona została otoczka wypukła z użyciem algorytmu Grahama oraz algorytmu Jarvisa dla trzech różnych liczbach punktów w celu zwiększenia wiarygodności pomiarów. Zmierzony został czas każdego wykonania algorytmu.

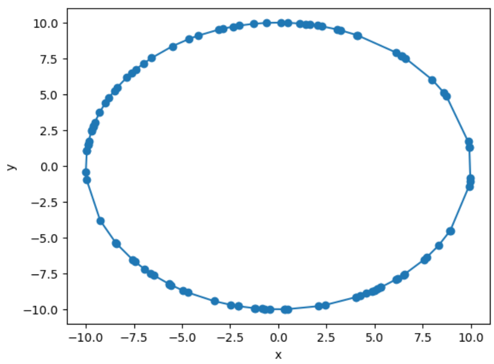
Złożoności obliczeniowe algorytmów:

- Algorytm Grahama - $O(nlogn)$
- Algorytm Jarvisa - $O(kn)$, k - liczba punktów otoczki

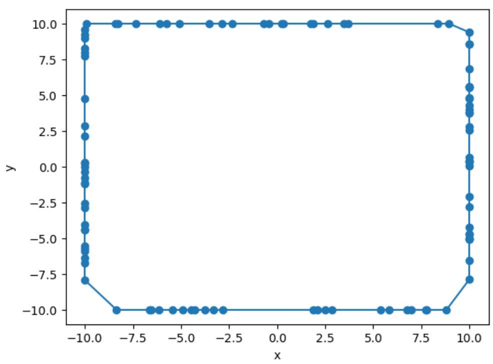
2.2.2 Otoczki wypukłe przykładowych zbiorów



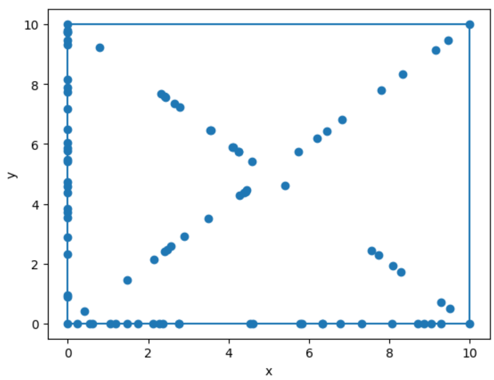
Rysunek 5: Zbiór 1.



Rysunek 6: Zbiór 2.



Rysunek 7: Zbiór 3.



Rysunek 8: Zbiór 4.

2.2.3 Liczności zbiorów użytych w poszczególnych pomiarach

- Zbiór 1. - 10 000 punktów, 100 000 punktów, 1 000 000 puntków,
- Zbiór 2. - 1000 punktów, 5000 punktów, 10 000 punktów,
- Zbiór 3. - 10 000 punktów, 100 000 punktów, 1 000 000 puntków,
- Zbiór 4. - 10 000 punktów, 100 000 punktów, 1 000 000 puntków.

3 Analiza wyników

3.1 Zestawienie czasów działania algorytmów

		Czas działania algorytmu		
		Pomiar I	Pomiar II	Pomiar III
Zbiór 1.	Algorytm Grahama	0.07 s	0.74 s	11.02 s
	Algorytm Jarvisa	0.09 s	1.06 s	12.86 s
Zbiór 2.	Algorytm Grahama	0.01 s	0.04 s	0.09 s
	Algorytm Jarvisa	0.53 s	15.01 s	59.03 s
Zbiór 3.	Algorytm Grahama	0.09 s	1.15 s	14.45 s
	Algorytm Jarvisa	0.05 s	0.44 s	4.33 s
Zbiór 4.	Algorytm Grahama	0.64 s	3.56 s	41.3 s
	Algorytm Jarvisa	0.11 s	0.42 s	3.59 s

Tabela 1: Czasy działania dla poszczególnych zbiorów i algorytmów.

3.2 Analiza otrzymanych czasów

Z Tabeli 1. wynika, że Algorytm Jarvisa osiąga lepszy czas niż Algorytm Grahama w przypadku zbioru 3. i 4. (odpowiednio punkty na bokach prostokąta i na bokach oraz przekątnych kwadratu). Natomiast w zbiorze 1. i 2. (odpowiednio losowa chmura punktów i okrąg) przewagę uzyskuje Algorytm Grahama, szczególnie dla zbioru, w którym punkty zlokalizowane są na okręgu różnica jest znacząca.

4 Wnioski

Szybkość porównywanych algorytmów

Czas działania Algorytmu Jarvisa jest zależny od liczby punktów należących do otoczki, natomiast Algorytm Grahama zawsze działa w czasie $O(nlogn)$. Otoczka wypukła zbioru 3. może składać się z maksymalnie ośmiu punktów (po dwa na każdy bok prostokąta), a zbioru 4. zawsze z dokładnie czterech (jeden w każdym rogu kwadratu), dlatego czas wykonania Algorytmu Jarvisa będzie tutaj znacznie krótszy. Zbiór, w którym wszystkie punkty położone są na okręgu (2.) charakteryzuje to, że każdy punkt wchodzi w skład otoczki. Wtedy złożoność rośnie do $O(n^2)$, więc o wiele szybszy będzie Algorytm Grahama. Zbiór 1. to losowa chmura punktów rozmieszczonych wewnątrz prostokąta. Czas obydwu algorytmów jest tutaj zbliżony, z lekką przewagą Algorytmu Grahama ze względu na to, że liczba punktów otoczki takiej chmury punktów jest zazwyczaj większa od $log_2 n$, gdzie n to liczba wszystkich punktów.

Podsumowanie

Algorytm Grahama lepiej sprawdza się w przypadku, gdy znaczna część puntków należy to otoczki wypukłej zbioru ze względu na to, że jego złożoność nie zależy od ich liczby. Z kolei Algorytm Jarvisa warto użyć jeżeli wiemy, że otoczka składa się z niewielkiej liczby punktów, dokładnie jeżeli $k < log_2 n$.