

# Algorytmy geometryczne – laboratorium 2. Otoczka Wypukła

## 1. Dane techniczne:

- System operacyjny: Windows 11 Pro (x64)
- Procesor: AMD Ryzen 5 7530U (2.0 – 4.5 GHz)
- Pamięć RAM: 16GB (2133 MHz)
- Środowisko: Jupyter Notebook
- Język programowania: Python 3.13.7

Podczas przeprowadzenia doświadczenia użyto narzędzi do wizualizacji danych autorstwa Koła Naukowego BIT oraz bibliotek *numpy*, *pandas*, *matplotlib*. Użyta precyzaja danych zmiennoprzecinkowych to *float64*. By zagwarantować powtarzalność generowanych danych użyto polecenia `np.random.seed(42)`.

## 2. Cel ćwiczenia

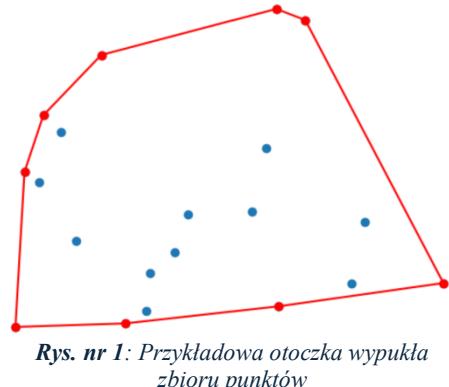
- Porównanie oraz graficzne przedstawienie działania algorytmów Grahama i Jarvisa.
- Określenie charakterystyki zbiorów, dla których zastosowanie powyższych algorytmów jest najefektywniejsze.

## 3. Wstęp teoretyczny

### 3.1. Definicja otoczki wypukłej

Otoczka wypukła  $\text{CH}(S)$  dowolnego niepustego zbioru zbioru punktów  $S$ , to najmniejszy zbiór wypukły zawierający wszystkie punkty z  $S$ . Na płaszczyźnie dwuwymiarowej taka otoczka sprowadza się do najmniejszego wielokąta wypukłego zawierającego wszystkie punkty ze zbioru  $S$ . Przykładowa otoczka wypukła przedstawiona została na rys. nr 1. Punkty oraz odcinki należące do otoczki przedstawione zostały kolorem czerwonym, a punkty nienależące, kolorem niebieskim.

Listę punktów otoczki zbioru  $S$  podaje się wg konwencji w porządku przeciwnym do ruchu wskazówek zegara (układ prawoskrętny).



W przypadku rozważania dodania punktów współliniowych do otoczki względem ostatniego punktu z  $\text{CH}(S)$  dodaje się tylko punkt najbardziej odległy względem uprzednio dodanego.

### 3.2. Algorytm Grahama

Algorytm Grahama jest jednym z klasycznych algorytmów służących do wyznaczania otoczki wypukłej dla zbioru punktów na płaszczyźnie.

Kroki algorytmu:

- Wybór punktu  $p_0$  o najmniejszej współrzędnej y (jeśli występuje kilka punktów o tej samej, najmniejszej współrzędnej y, to wybiera się ten o najmniejszej współrzędnej x).
- Posortowanie pozostałych punktów względem kąta, jaki tworzą z osią x i punktem odniesienia. Jeśli dwa punkty tworzą ten sam kąt, to sortuje się je po odległości od punktu początkowego  $p_0$ .
- Przechodzenie po punktach w kolejności kątowej i budowanie otoczki przy pomocy stosu. Dla każdego kolejnego punktu sprawdza się, czy skręt utworzony przez 2 ostatnie punkty na stosie oraz kolejnego punktu z posortowanej tablicy skierowany jest w lewo. Jeśli tak, to punkt zostaje dodany do stosu. Jeśli skręt jest w prawo lub punkty są współliniowe, usuwa się punkty ze szczytu stosu, aż do uzyskania skrętu w lewo. Następnie rozważany punkt zostaje dodany na stos. Takie podejście pozwala na pominięcie usuwania punktów współliniowych na etapie sortowania i wybierania tego punktu, który położony jest najdalej od  $p_0$ .

Złożoność obliczeniowa algorytmu Grahama wyznaczania  $\text{CH}(S)$  wynosi  $O(n \log n)$ , gdzie n to liczba punktów zbioru S.

Zależy ona od głównych operacji:

- wyboru punktu początkowego -  $O(n)$ ,
- posortowanie pozostałych punktów -  $O(n \log n)$ ,
- iteracja po posortowanych punktach oraz operacje na stosie -  $O(n)$ .

### 3.3. Algorytm Jarvisa

Algorytm Grahama także jest jednym z klasycznych algorytmów służących do wyznaczania otoczki wypukłej dla zbioru punktów na płaszczyźnie. Działa on w sposób odmienny od algorytmu Grahama i osiąga inną złożoność czasową.

Kroki algorytmu:

- Wybór punktu  $p_0$  o najmniejszej współrzędnej y.
- Wybór następnego punktu wchodzącego w skład otoczki. Początkowo przyjmuje się dowolny punkt. Następnie dla pozostałych punktów ze zbioru sprawdza się, czy tworzą one z ostatnim punktem z otoczki oraz wcześniejszym kandydatem kąt prawoskrętny. Jeśli tak, to kandydat zostaje podmieniony na aktualnie rozważany

punkt. W przypadku punktów 2 punktów wspólniowych wybiera się punkt, który znajduje się dalej od poprzedniego punktu z wcześniej wyznaczonego fragmentu otoczki. Proces powtarzany jest do momentu powrotu do punktu początkowego, co zamyka otoczkę wypukłą.

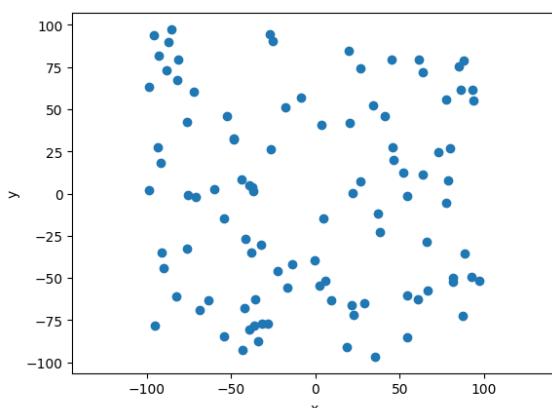
Złożoność obliczeniowa algorytmu Jarvisa wyznaczania CH(S) wynosi  $O(hn)$ , gdzie  $n$  to liczba punktów zbioru  $S$ , a  $h$  – liczba punktów należących do otoczki. W niektórych przypadkach złożoność czasowa tego algorytmu może sprowadzać się do  $O(n^2)$  – gdy liczność punktów otoczki osiąga liczbę zbioru  $S$ .

Zależy ona od głównych operacji:

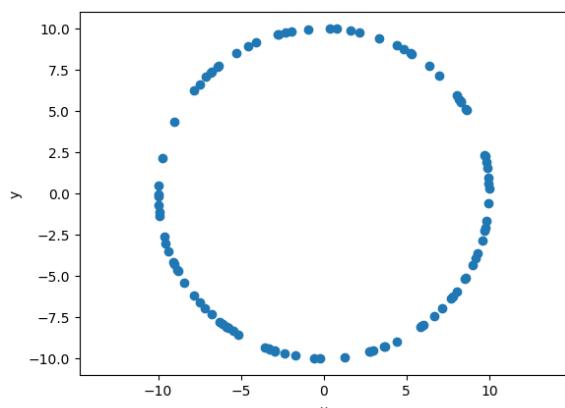
- wyboru punktu początkowego -  $O(n)$ ,
- wyboru  $h$  kolejnych punktów na otoczce ze zbioru  $S$  o liczności  $n$  -  $O(hn)$ .

## 4. Przebieg doświadczenia

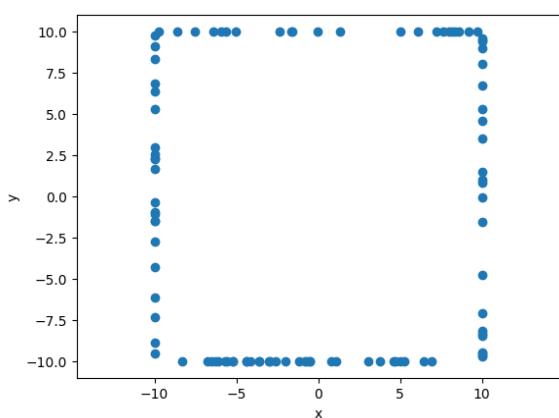
Doświadczenie rozpoczęto od wygenerowania (przy pomocy `np.random.uniform()` oraz `np.random.choice()`) 4 zbiorów testowych punktów typu `float64` na dwuwymiarowej przestrzeni euklidesowej oraz ich zwizualizowania. Zostały one przedstawione na rysunkach o nr od 2 do 5.



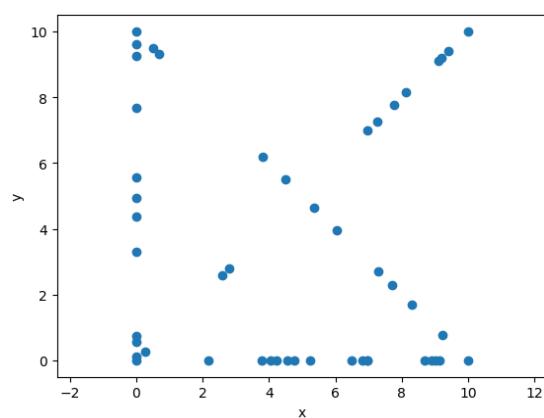
Rys. nr 2: Zbiór A - 100 losowych punktów o współrzędnych z przedziału  $x \in [-100, 100]$  oraz  $y \in [-100, 100]$



Rys. nr 3: Zbiór B - 100 losowych punktów leżących na okręgu o środku  $O = (0,0)$  i promieniu  $R = 10$



Rys. nr 4: Zbiór C - 100 losowych punktów leżących na obwodzie prostokąta wyznaczanego przez wierzchołki  $(-10, -10), (10, -10), (10, 10), (-10, 10)$



Rys. nr 5: Zbiór D - 25 losowych punktów leżących na dwóch bokach prostokąta leżących na osiach układu współrzędnych oraz 20 punktów na przekątnych tego prostokąta wyznaczanego przez wierzchołki  $(0,0), (10,0), (10,10), (0,10)$

Do wygenerowania punktów na bokach kwadratu użyto funkcji wyznaczającej punkty na prostej określonej przez jej punkt początkowy i końcowy, a także postaci parametrycznej prostej.

Następnie zaimplementowano algorytmy Grahama oraz Jarvisa do wyznaczania otoczki wypukłej. Zamiast funkcji wyznaczającej kąt użyto własnej implementacji funkcji obliczania wyznacznika macierzy  $3 \times 3$  określającej położenie 3 punktów względem siebie.

Wartość współczynnika tolerancji  $\varepsilon$  ustalono na  $10^{-24}$  ze względu na wskazania błędnych wyników dla większych wartości.

Kolejnym krokiem doświadczenia było zastosowania obu algorytmów wyznaczania otoczki wypukłej na każdym z 4 wygenerowanych zbiorów. Otrzymane wyniki przedstawiono na poniższych wykresach.

Ostatnim etapem ćwiczenia była analiza czasu działania implementacji tych algorytmów na zbiorach o większej liczbie punktów oraz o charakterystykach podobnych do zbiorów A, B, C oraz D. Tę analizę przedstawiono jako ostatnią.

## 5. Opracowanie wyników

### 5.1. Otoczka wypukła zbiorów A, B, C oraz D

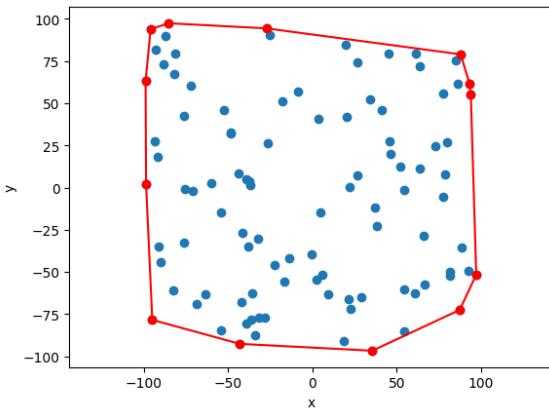
Wyznaczone otoczki były jednakowe dla obu stosowanych sposobów ich wyznaczania. Oznacza to, że implementacje algorytmów działają poprawnie dla każdego ze zbiorów. Liczności punktów wchodzących w ich skład przedstawiono w tabeli nr 1. Wizualizacje wyników algorytmów przedstawiono na rysunkach o nr od 6 do 9.

Wyniki zwizualizowano zgodnie z legendą:

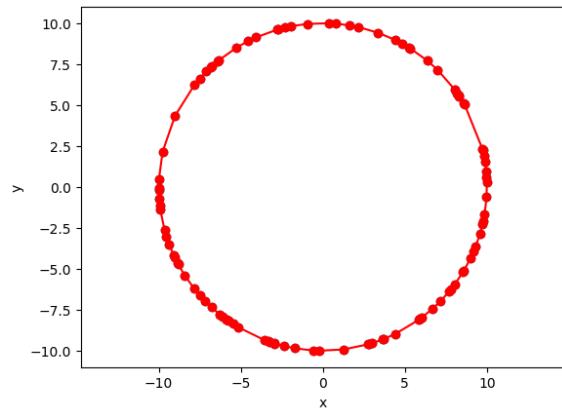
- **na czerwono** – punkty oraz odcinki należące do otoczki wypukłej
- **na niebiesko** – punkty ze zbioru nienależące do otoczki wypukłej

Zbiór punktów	Liczba punktów należących do otoczki wypukłej	
	Algorytm Graham	Algorytm Jarvisa
A	13	13
B	100	100
C	8	8
D	4	4

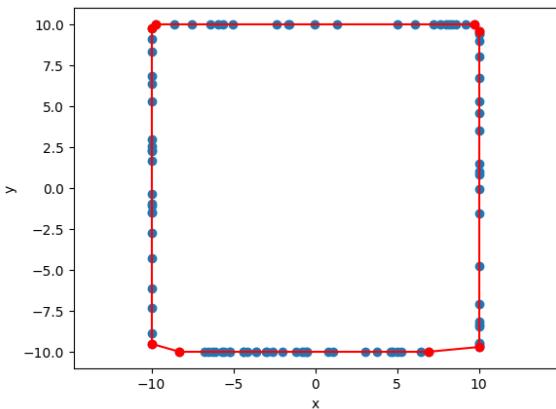
**Tabela nr 1:** Liczności punktów należących do otoczek wypukłych zbiorów A, B, C oraz D wyznaczonych algorytmami Grahama oraz Jarvisa



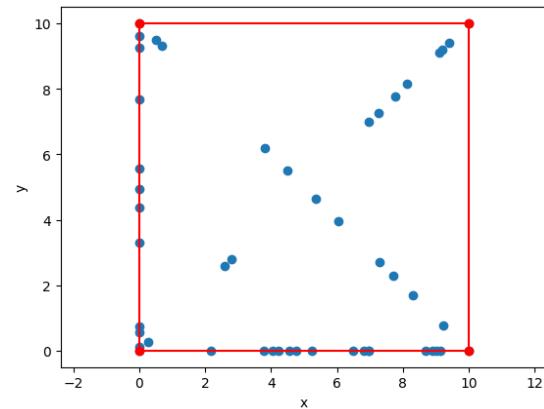
Rys. nr 6: Wyznaczona otoczka wypukła dla zbioru A



Rys. nr 7: Wyznaczona otoczka wypukła dla zbioru B



Rys. nr 8: Wyznaczona otoczka wypukła dla zbioru C



Rys. nr 9: Wyznaczona otoczka wypukła dla zbioru D

W przypadku zbioru A, 13 ze 100 punktów zostało zakwalifikowanych jako należące do otoczki wypukłej. Na poprawność jej wyznaczenia wskazuje widoczny fakt zawierania wszystkich punktów w obszarze ograniczonym przez nią.

Lista punktów zakwalifikowanych do otoczki wypukłej zbioru A dla:

#### algorytmu Grahma

```
[ (35.51287236845647, -96.68243421442877),
  (87.3459977473469, -72.49581117080135),
  (97.13009082212014, -51.58894569769992),
  (93.91692555291172, 55.02656467222292),
  (93.12640661491187, 61.67946962329222),
  (87.89978831283781, 78.96547008552977),
  (-27.2740795241412, 94.35641654419214),
  (-85.08987126404584, 97.37738732010345),
  (-95.88310114083951, 93.98197043239887),
  (-98.89557657527952, 63.09228569096683),
  (-98.60957389376186, 2.149460515513141),
  (-94.91617465118097, -78.42171460133912),
  (-43.031901124506476, -92.62261052909344)]
```

#### algorytmu Jarvisa

```
[ (35.51287236845647, -96.68243421442877),
  (87.3459977473469, -72.49581117080135),
  (97.13009082212014, -51.58894569769992),
  (93.91692555291172, 55.02656467222292),
  (93.12640661491187, 61.67946962329222),
  (87.89978831283781, 78.96547008552977),
  (-27.2740795241412, 94.35641654419214),
  (-85.08987126404584, 97.37738732010345),
  (-95.88310114083951, 93.98197043239887),
  (-98.89557657527952, 63.09228569096683),
  (-98.60957389376186, 2.149460515513141),
  (-94.91617465118097, -78.42171460133912),
  (-43.031901124506476, -92.62261052909344)]
```

W przypadku zbioru B każdy z punktów (100 punktów) został zakwalifikowany jako należący do otoczki wypukłej. Każdy z punktów leżących na okręgu będzie zawsze wchodził w skład otoczki wypukłej, ponieważ jest on najbardziej oddalonym punktem na prostej, która przebiega przez środek figury, w swoim kierunku. Pominięcie jakiegokolwiek z punktów w otoczce wiązałoby się z jego niezawieraniem się w obszarze ograniczonym przez otoczkę złożonej z innych punktów zbioru.

W przypadku zbioru C oraz D wszystkie punkty zostały prawidłowo zakwalifikowane. Przypadki wspólnotowości punktów zostały prawidłowo rozpatrzone, na co wskazuje poprawny wynik oraz wizualizacja. Metoda określania takiej relacji między punktami okazała się wystarczająco dokładna.

Lista punktów zakwalifikowanych do otoczki wypukłej zbioru C dla:

**algorytmu Grahama**

```
[(-8.30324571829616, -10.0),
(6.942862881911797, -10.0),
(10.0, -9.712130227404883),
(10.0, 9.59021057243017),
(9.693909194192305, 10.0),
(-9.720021276457416, 10.0),
(-10.0, 9.783246970394032),
(-10.0, -9.517041589250692)]
```

**algorytmu Jarvisa**

```
[(-8.30324571829616, -10.0),
(6.942862881911797, -10.0),
(10.0, -9.712130227404883),
(10.0, 9.59021057243017),
(9.693909194192305, 10.0),
(-9.720021276457416, 10.0),
(-10.0, 9.783246970394032),
(-10.0, -9.517041589250692)]
```

Punkty zakwalifikowane do otoczki wypukłej zbioru D dla:

**algorytmu Grahama**

```
[(0, 0),
(10, 0),
(10, 10),
(0, 10)]
```

**algorytmu Jarvisa**

```
[(0, 0),
(10, 0),
(10, 10),
(0, 10)]
```

W każdym z przypadków oba algorytmy wyznaczyły dokładnie te same punkty jako należące do otoczki wypukłej zbioru.

## 5.2. Generowanie dodatkowych zbiorów danych oraz pomiar czasu pracy algorytmów

Korzystając z wcześniej zdefiniowanych funkcji wygenerowano nowe zbiory o cechach zbiorów A, B, C oraz D. Ich kolejne liczności przedstawiono w tabeli nr 2.

Liczności zbiorów o charakterystyce B zostały ograniczone ze względu na zbyt długi czas pracy algorytmu Jarvisa. Punkty w tych zbiorach zostały umieszczone na okręgu o środku  $O = (0, 0)$  oraz promieniu  $R = 1000$ .

Połowa punktów w zbiorach o charakterystyce zbioru D znajduje się na lewym i dolnym boku prostokąta. Druga połowa znajduje się na jego przekątnych.

Charakterystyka zbioru	Kolejne liczności generowanych zbiorów
A, C, D	1000, 2500, 5000, 7500, 10000, 25000, 50000, 75000, 100000
B	1000, 2500, 5000, 7500, 10000, 25000, 50000

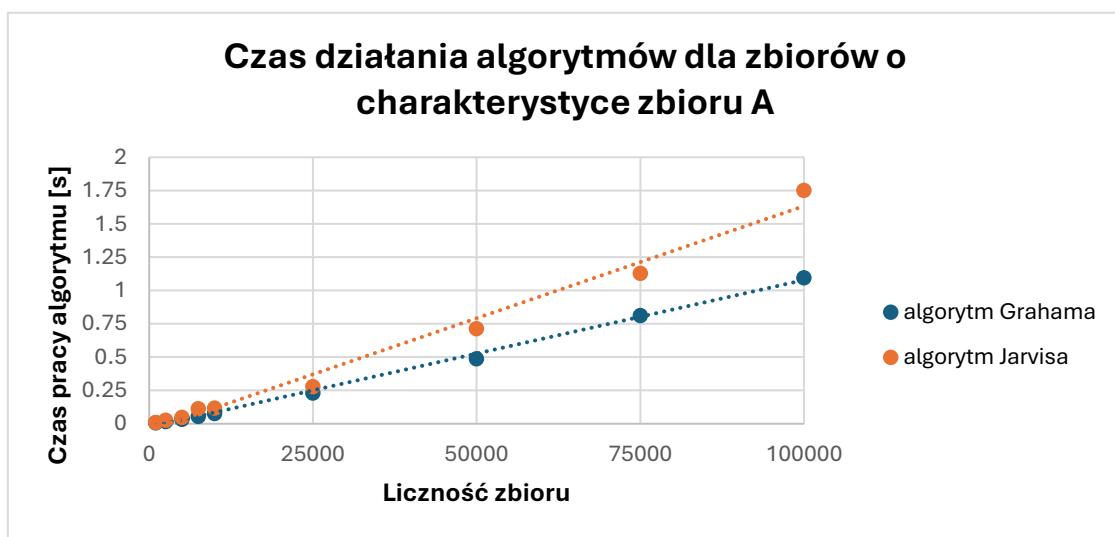
Tabela nr 2: Kolejne liczności generowanych zbiorów

### 5.2.1. Dodatkowe zbiory o charakterystyce zbioru A

Wyniki, czas pracy oraz jego wykres przedstawiono odpowiednio w tabeli nr 3 oraz na rysunku o nr 10.

	Czas pracy algorytmu [s]								
Liczność zbioru	1000	2500	5000	7500	10000	25000	50000	75000	100000
Liczność otoczki	20	23	21	32	25	24	30	30	31
Graham	0,0072	0,0167	0,0352	0,0552	0,0779	0,2318	0,4877	0,8120	1,0960
Jarvis	0,0095	0,0274	0,0491	0,1132	0,1199	0,2803	0,7151	1,1303	1,7521

Tabela nr 3: Czas pracy algorytmów dla zbiorów o charakterystyce zbioru A



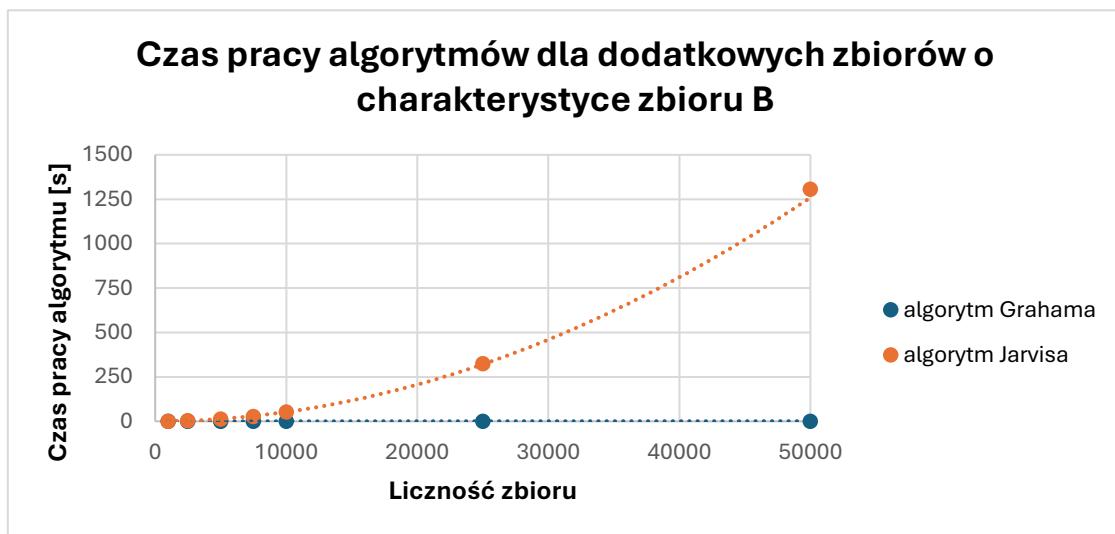
Rys. nr 10: Wykres czasu pracy algorytmów dla zbiorów o charakterystyce zbioru A

### 5.2.2. Dodatkowe zbiory o charakterystyce zbioru B

Wyniki, czas pracy oraz jego wykres przedstawiono odpowiednio w tabeli nr 4 oraz na rysunku o nr 11.

	Czas pracy algorytmu [s]						
Liczność zbioru	1000	2500	5000	7500	10000	25000	50000
Liczność otoczki	1000	2500	5000	7500	10000	25000	50000
Graham	0,0063	0,0230	0,0413	0,0543	0,0771	0,2460	0,5023
Jarvis	0,5741	3,6144	12,699	27,476	53,035	324,97	1306,8

Tabela nr 4: Czas pracy algorytmów dla zbiorów o charakterystyce zbioru B



Rys. nr 11: Wykres czasu pracy algorytmów dla zbiorów o charakterystyce zbioru B

W każdym z przypadków zbiorów o charakterystyce zbioru B oba algorytmy zwróciły otoczkę wypukłą, złożoną ze wszystkich punktów w rozważanym zbiorze.

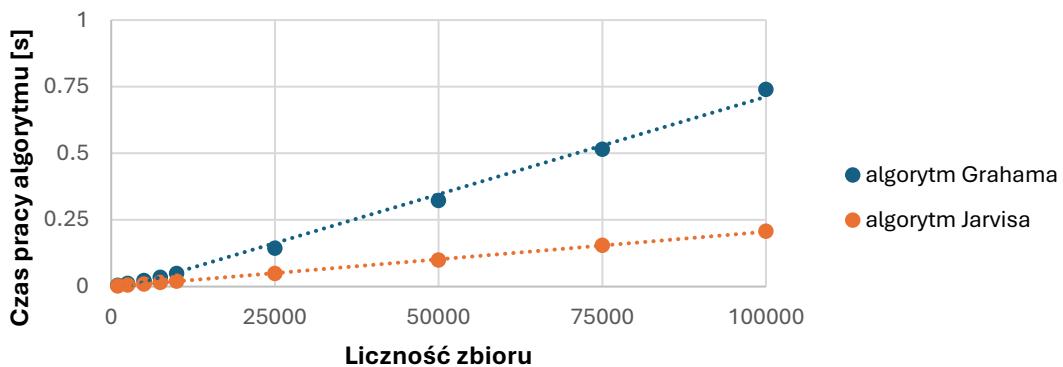
### 5.2.3. Dodatkowe zbiory o charakterystyce zbioru C

Wyniki, czas pracy oraz jego wykres przedstawiono odpowiednio w tabeli nr 5 oraz na rysunku o nr 12.

	Czas pracy algorytmu [s]								
Liczność zbioru	1000	2500	5000	7500	10000	25000	50000	75000	100000
Liczność otoczki	8	8	8	8	8	8	8	8	8
Graham	0,0039	0,0113	0,0220	0,0348	0,0493	0,1439	0,3221	0,5145	0,7398
Jarvis	0,0024	0,0049	0,0096	0,0145	0,0199	0,0484	0,0987	0,1549	0,2073

Tabela nr 5: Czas pracy algorytmów dla zbiorów o charakterystyce zbioru C

### Czas pracy algorytmów dla dodatkowych zbiorów o charakterystyce zbioru C



Rys. nr 12: Wykres czasu pracy algorytmów dla zbiorów o charakterystyce zbioru C

W każdym z przypadków zbiorów o charakterystyce zbioru C oba algorytmy zwróciły otoczkę wypukłą, złożoną z 8 punktów, 4 par wierzchołków leżących najbliżej wierzchołków wyznaczających prostokąt.

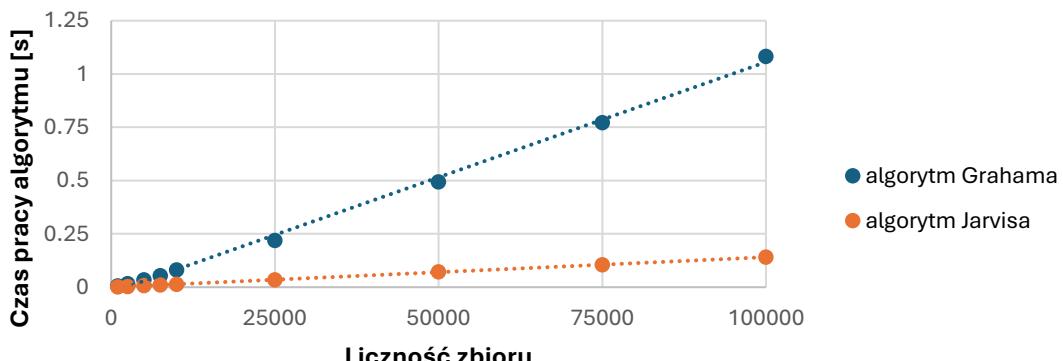
#### 5.2.4. Dodatkowe zbiorы o charakterystyce zbioru D

Wyniki, czas pracy oraz jego wykres przedstawiono odpowiednio w tabeli nr 6 oraz na rysunku o nr 13.

Liczność zbioru	Czas pracy algorytmu [s]								
	1000	2500	5000	7500	10000	25000	50000	75000	100000
Liczność otoczki	4	4	4	4	4	4	4	4	4
Graham	0,0057	0,0158	0,0346	0,0547	0,0804	0,2186	0,4941	0,7724	1,0831
Jarvis	0,0016	0,0033	0,0070	0,0102	0,0142	0,0349	0,0715	0,1053	0,1407

Tabela nr 6: Czas pracy algorytmów dla zbiorów o charakterystyce zbioru D

### Czas pracy algorytmów dla dodatkowych zbiorów o charakterystyce zbioru D



Rys. nr 13: Wykres czasu pracy algorytmów dla zbiorów o charakterystyce zbioru D

W każdym z przypadków zbiorów o charakterystyce zbioru D oba algorytmy zwróciły otoczkę wypukłą, złożoną z 4 punktów, wierzchołków wyznaczających prostokąt.

## 6. Wnioski

W przypadku zbiorów o charakterystyce zbioru A widać przewagę w efektywności działania algorytmu Grahama. Zwiększa się ona w przypadku zbiorów wzorowanych na zbiorze B. Algorytm Jarvisa, ze względu na charakterystykę zbioru osiąga złożoność czasową  $O(n^2)$ , co czyni go mało przydatnym w porównaniu do algorytmu Grahama. Zarówno w przypadku zbioru wzorowanym na zbiorze C oraz D, algorytm Jarvisa daje dużo lepsze rezultaty czasowe. Wykres jego czasu działania jest reprezentowany przez prostą o dużo mniejszym stopniu nachylenia, jego czas działania rośnie wolniej niż czas działania algorytmu Grahama dla coraz liczniejszych zbiorów.

Na podstawie analizy wyników doświadczenia nasuwają się następujące wnioski:

- 1) Dobór odpowiedniego algorytmu obliczania wyznacznika jest ważny dla osiągnięcia czasowo efektywnego wyznaczenia otoczki wypukłej zbioru. Jeśli występuje podejrzenie, że liczba punktów na otoczce będzie mała, to należy zastosować algorytm Jarvisa. W przeciwnym przypadku lepiej sprawdzi się algorytm Grahama.
- 2) Wybór algorytmu wyznaczania otoczki nie wpływa na samą poprawność wyniku, a jedynie na czas jego osiągnięcia.
- 3) Dobrą praktyką jest graficzne przedstawienie zbiorów, których otoczkę wypukłą należy wyznaczyć. Pomaga to określić charakterystykę zbioru, a tym samym, ułatwia wybór odpowiedniego algorytmu.