1. Poprawność algorytmów

Poprawność działania algorytmów Grahama i Jarvisa zależy od dokładności jaką wybierzemy. Epsilon równy 10^{-11} (czyli taki jaki jest w moich algorytmach) pozwalał algorytmom poprawnie znajdować punkty leżące na jednej prostej. Niewykluczone, że wtedy również punkty nie będące na jednej prostej zostaną uznane za współliniowe (powodem tych problemów jest dokładność wyznacznika). Oznacza to, że algorytmy powinny działać dobrze dla zbioru danych z podpunktu d, ale może występować pomijanie niektórych wierzchołków w zbiorach danych a, b, c, ponieważ zostaną uznane za współliniowe.

2. Porównanie czasu działania algorytmów dla różnych zbiorów

Zbiory a (punkty losowo w przedziale [-100, 100]:

Wielkość zbioru [liczba punktów], liczba powtórzeń algorytmu	Algorytm Grahama [s]	Algorytm Jarvisa [s]
10, 5000	2.3228026120013965	2.7164485649991548
100, 100	0.8582590799996979	1.1259706219989312
1000, 20	2.1907365059996664	2.9375181539999176
10000, 5	7.295819648999895	10.760283062998496
50000, 1	8.279267199999595	14.281790332001037

Tabela 1. Porównanie czasu działania algorytmów dla zbiorów takich jak w treści zadania 1a

Zatem dla zbioru a algorytm Grahama okazał się być szybszy.

Zbiory b (okrąg o środku w (0,0) i promieniu długości 19):

Wielkość zbioru [liczba punktów], liczba powtórzeń algorytmu	Algorytm Grahama [s]	Algorytm Jarvisa [s]
10, 5000	2.2921645830010675	4.1012604879997525
100, 100	0.8124440050014528	8.340807022999797
1000, 20	1.0967500209990249	91.60472364399902
2000, 1	0.27490986100019654	36.57838087900018

Tabela 2. Porównanie czasu działania algorytmów dla zbiorów takich jak w treści zadania 1b

Ponieważ szybkość działania algorytmu Jarvisa zależy od liczby wierzchołków otoczki wypukłej, a w zbiorze b wszystkie punkty tworzą tę otoczkę, to jest on wyraźnie wolniejszy od Algorytmu Grahama, którego złożoność obliczeniowa jest zawsze nie gorsza niż O(nlogn).

Zbiory c (punkty leżące na bokach prostokąta o wierzchołkach (-10, 10), (-10, -10), (10, -10), (10, 10):

Wielkość zbioru [liczba punktów], liczba powtórzeń algorytmu	Algorytm Grahama [s]	Algorytm Jarvisa [s]
10, 5000	2.4702575610008353	2.978018609001083
100, 100	0.8962860559986439	0.7651736060015537
1000, 20	2.456866506001461	1.4837456850000308
10000, 5	7.6371898889992735	3.6007481859996915
50000, 1	8.829956400999436	3.678853209999943

Tabela 3. Porównanie czasu działania algorytmów dla zbiorów takich jak w treści zadania 1c

W tym przypadku otoczka wypukła ma co najwyżej 8 wierzchołków (skrajne punkty na każdym boku prostokąta) zatem algorytm Jarvisa jest szybszy od algorytmu Grahama dla większych zbiorów danych.

Zbiory d (punkty leżące na wierzchołkach kwadratu (0, 0), (0, 10), (10, 0), (10, 10), na bokach, które leżą na osiach układu współrzędnych oraz na przekątnych):

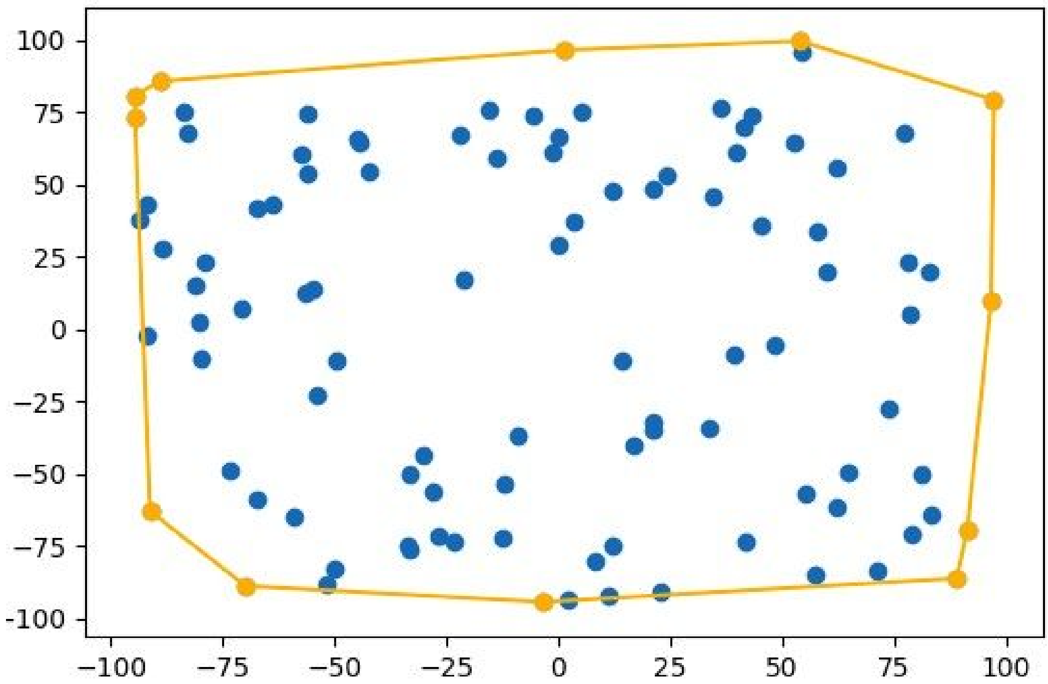
Liczba punktów na boku, liczba punktów na przekątnej, liczba powtórzeń algorytmu	Algorytm Grahama [s]	Algorytm Jarvisa [s]
5, 4, 5000	7.645719124000607	3.9936931709999044
25, 20, 100	1.4891500039993844	0.33599933500045154
250, 200, 10	7.2275983359995735	0.3790534870004194
2500, 2000, 1	70.92529695600024	0.3708012899987807

Tabela 4. Porównanie czasu działania algorytmów dla zbiorów takich jak w treści zadania 1d

Tutaj otoczka wypukła ma 4 wierzchołki (wierzchołki kwadratu) zatem algorytm Jarvisa jest szybszy od algorytmu Grahama dla dużych zbiorów.

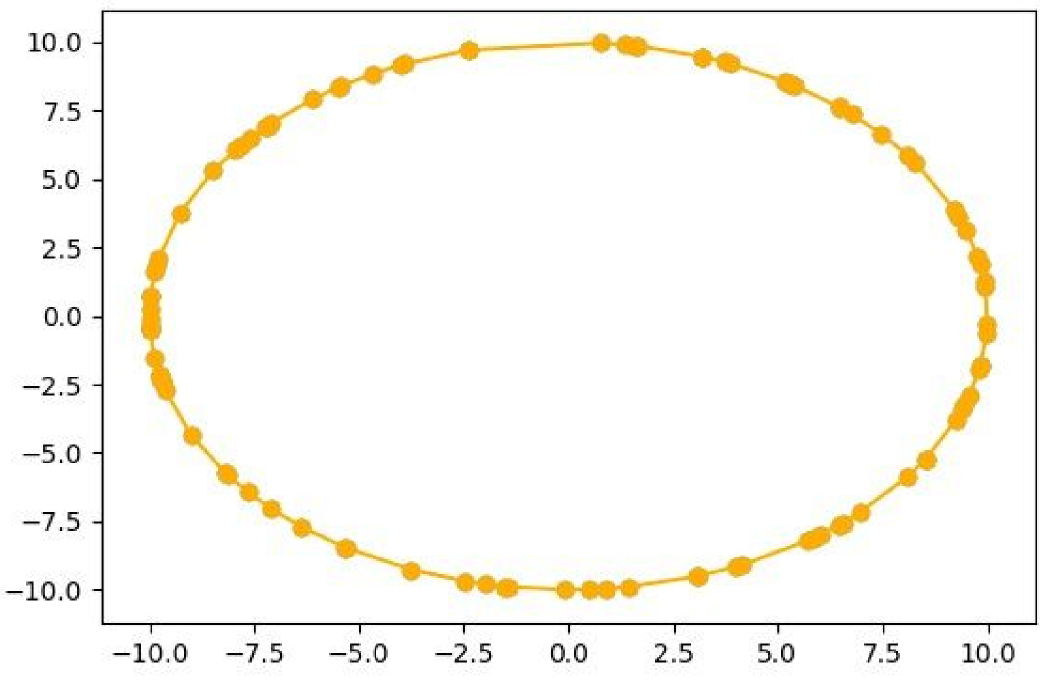
3. Dlaczego takie zbiory punktów?

Zbiór a jest losowym zbiorem punktów. Taki brak uporządkowania jest sytuacją najbardziej prawdopodobną.



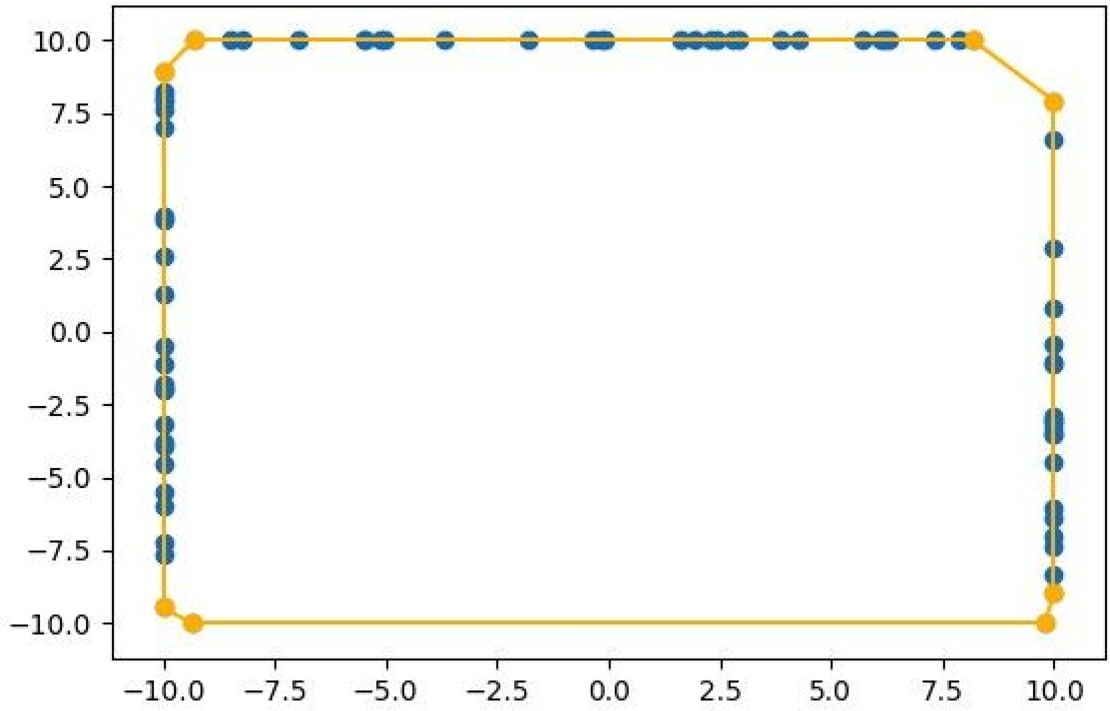
Rys. 1. Otoczka wypukła dla losowego zbioru punktów

Zbiór b zawiera punkty leżące na okręgu. Jest to sytuacja najbardziej niekorzystna dla algorytmu Jarvisa, ponieważ podwyższa jego złożoność obliczeniową do $O(n^2)$.

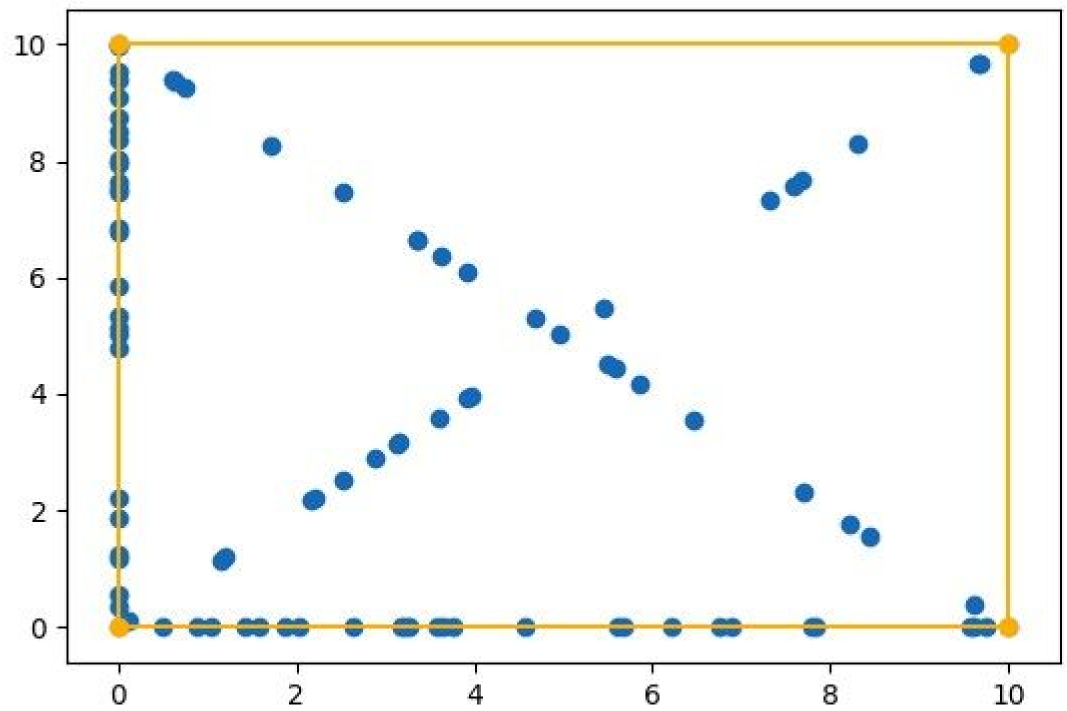


Rys. 2. Otoczka wypukła dla punktów na okręgu

Zbiory c i d pokazują sytuację, w której algorytm Jarvisa radzi sobie lepiej z uwagi na niewielką liczbę wierzchołków otoczki wypukłej.



Rys. 3. Otoczka wypukła dla punktów na bokach prostokąta



Rys. 4. Otoczka wypukła dla punktów na dwóch bokach i na przekątnych kwadratu

4. Wnioski

Różnica w szybkości działania algorytmów dla losowego zbioru punktów jest niewielka, ale algorytm Grahama okazuje się być szybszy. Dla zbioru punktów, w których liczba wierzchołków otoczki wypukłej jest duża, algorytm Grahama jest wydajniejszy. Dla zbiorów, w których liczba wierzchołków otoczki wypukłej jest niewielka to algorytm Jarvisa jest wydajniejszy.