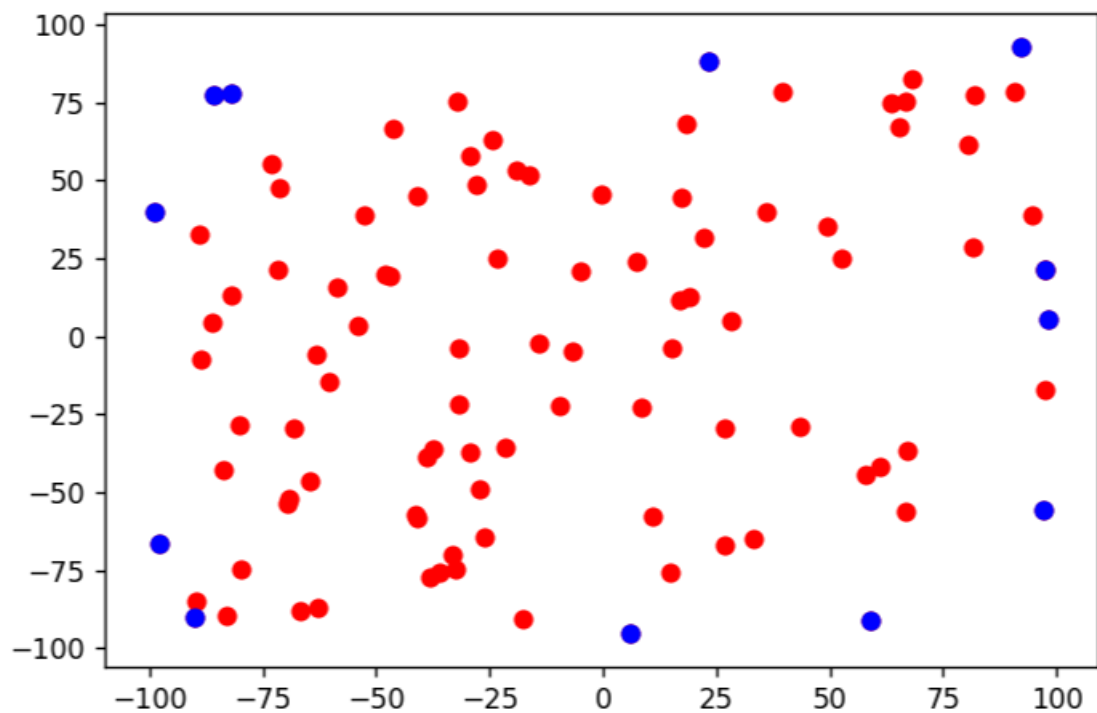
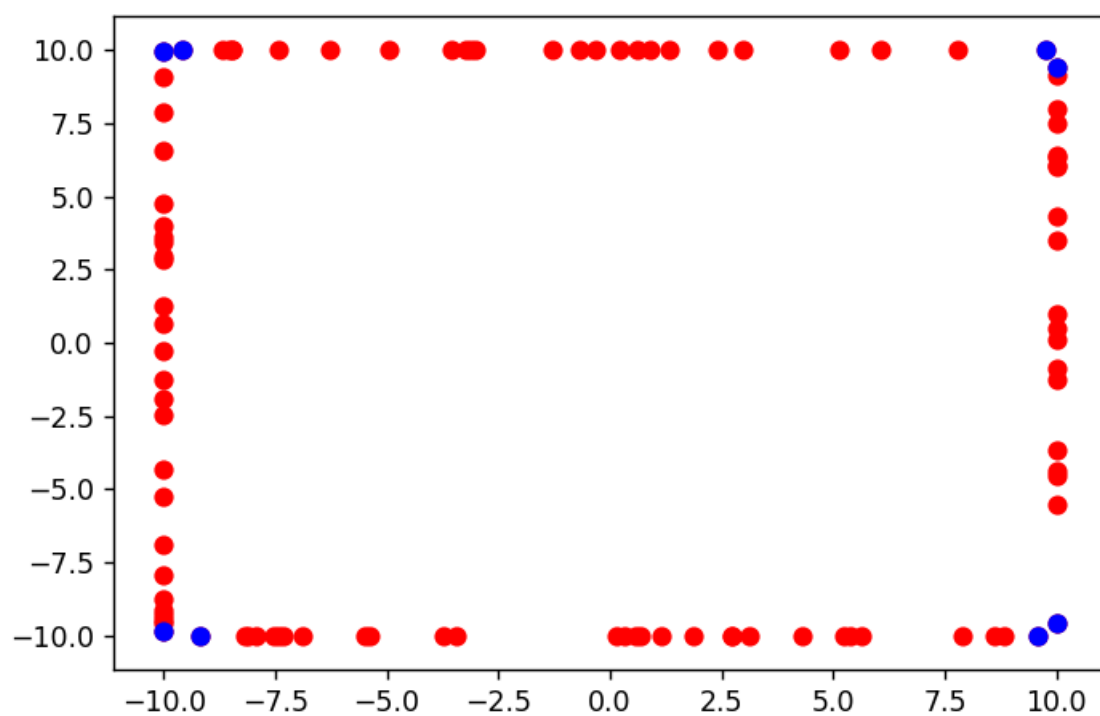
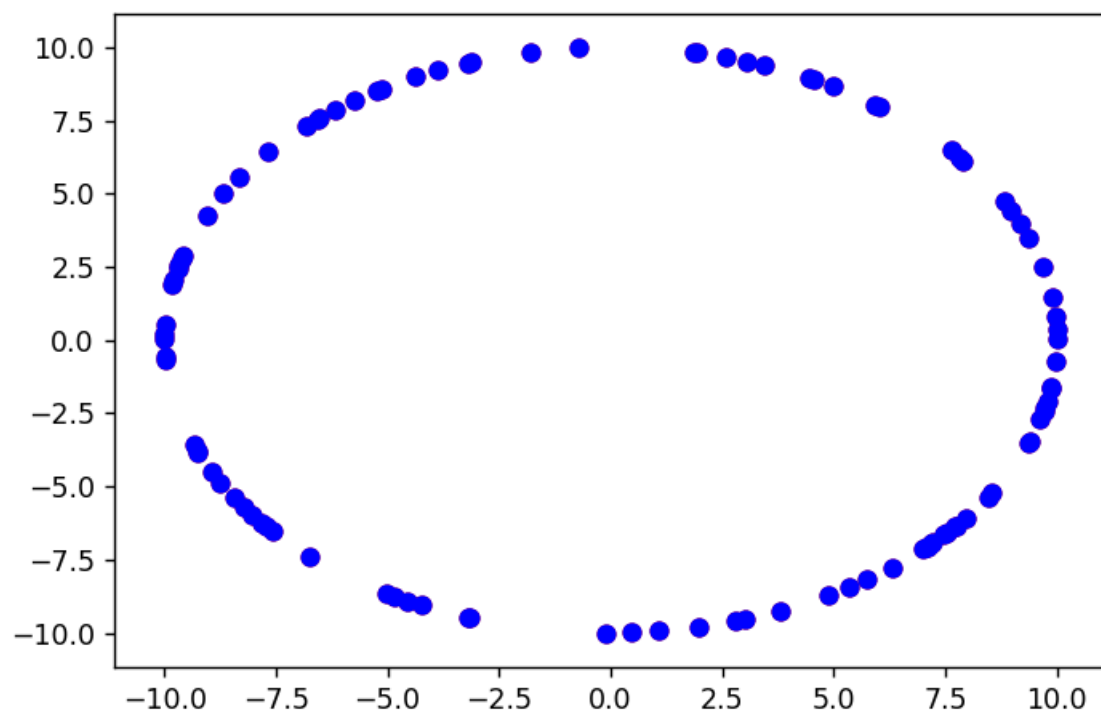


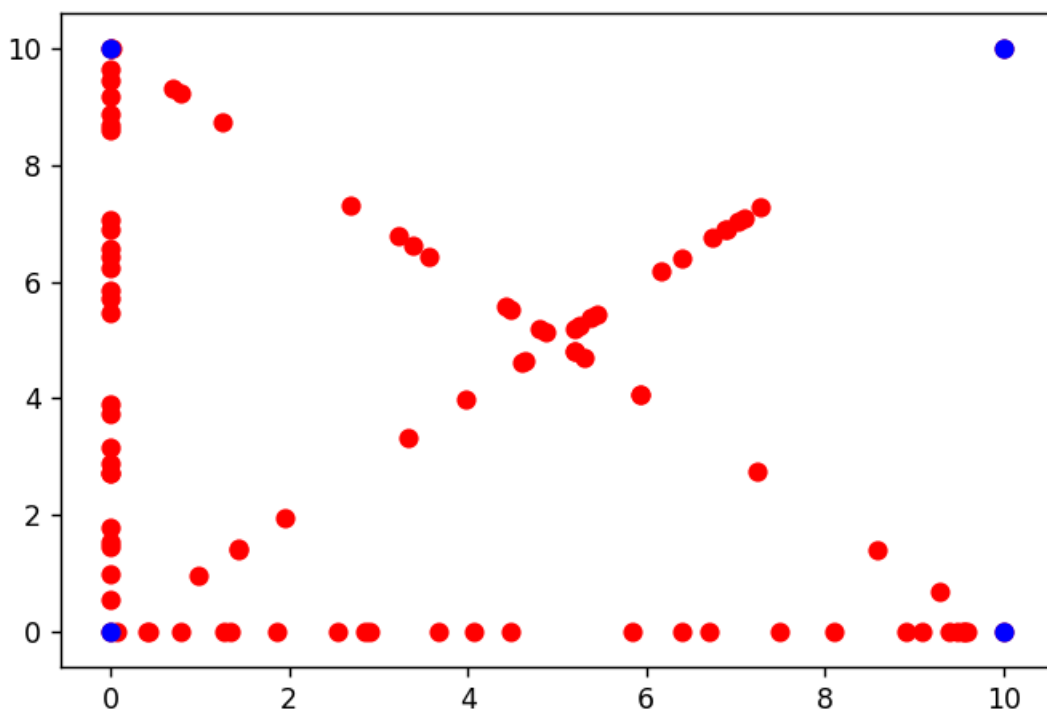
# Otoczka wypukła – opracowanie wyników

## 1) Algorytm Grahama

Algorytm dla wszystkich zbiorów danych zadziałał poprawnie. Wymagało to jednak ustawienia odpowiedniego marginesu błędu (epsilon) przy określaniu po której stronie odcinka znajduje się dany punkt (bądź też które punkty tworzą taki sam kąt). Znalezione otoczki (zaznaczone kolorem niebieskim) dla zbiorów z podpunktu 1):







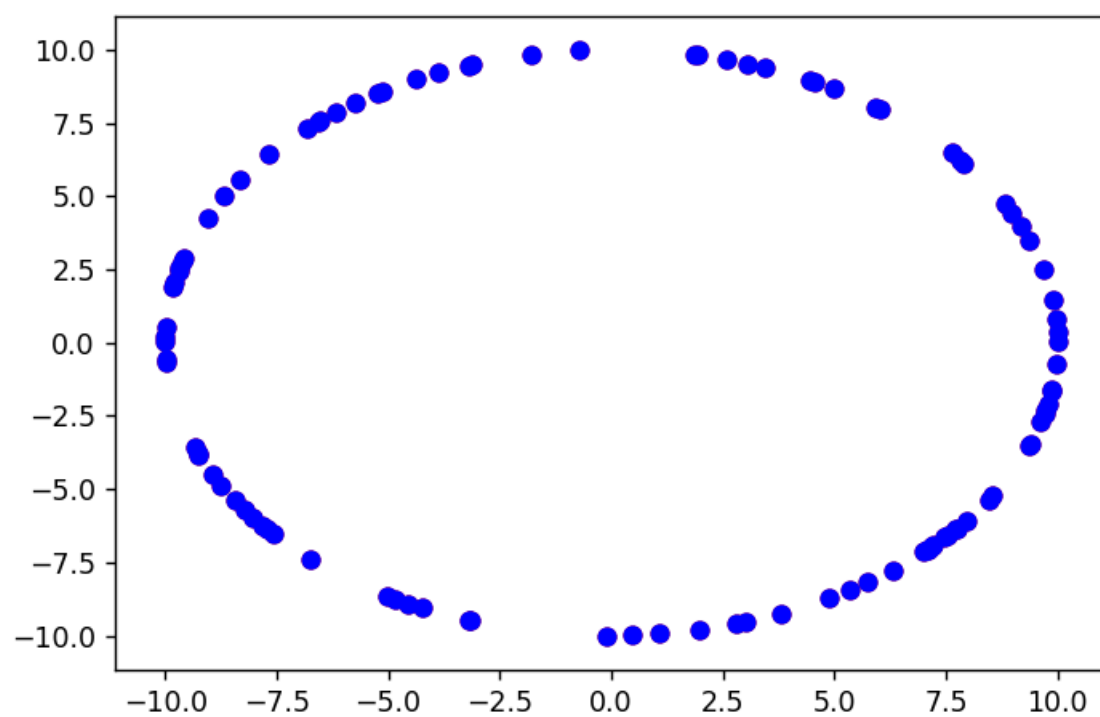
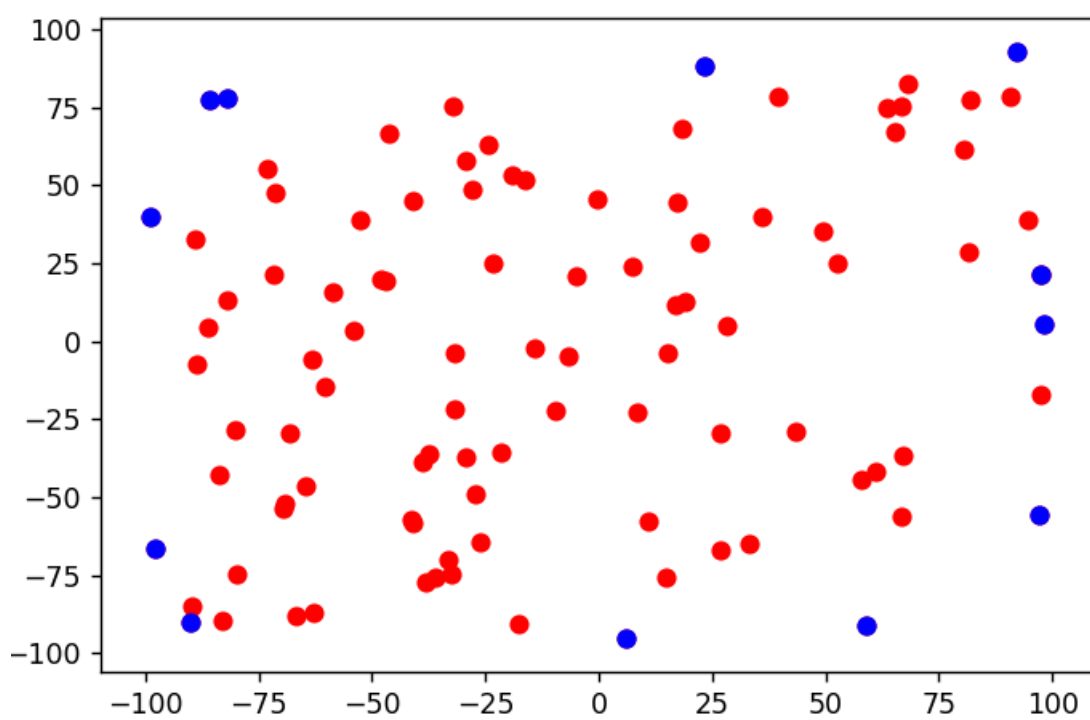
Co więcej algorytm działał dość szybko, poniżej przedstawiam czasy wykonania dla powyższych zbiorów:

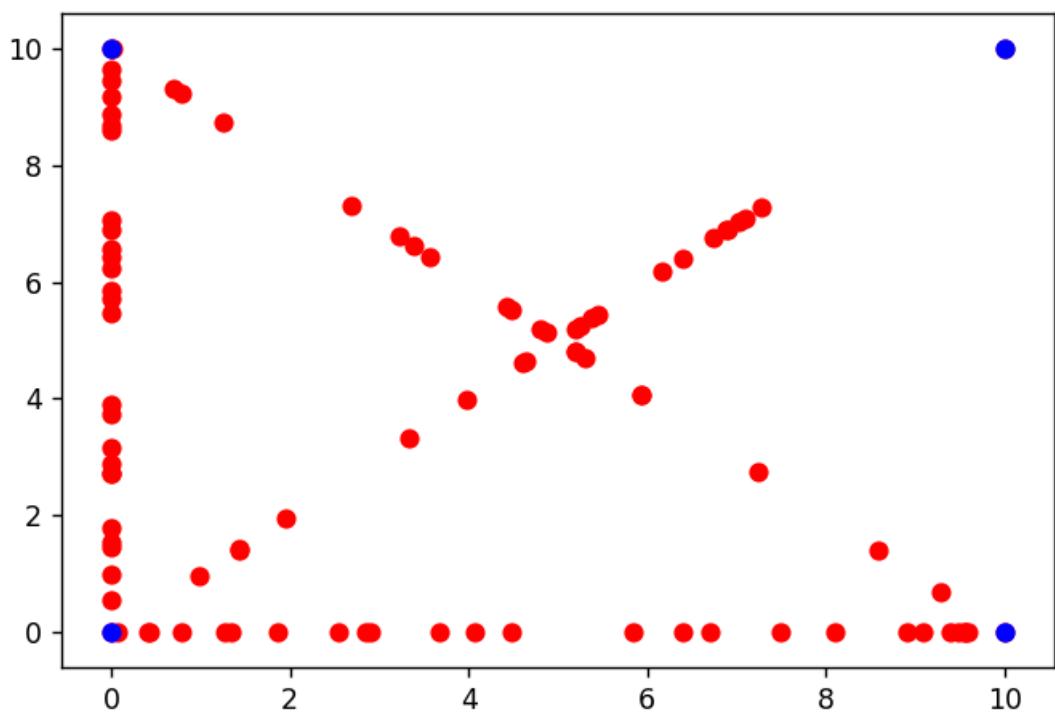
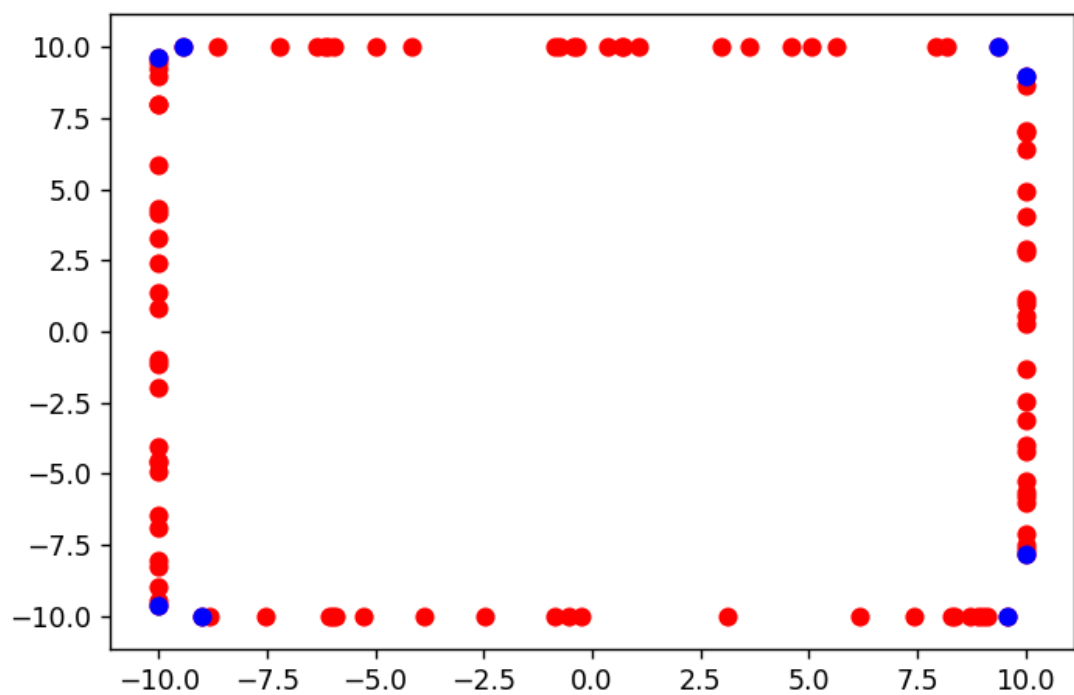
```
Starting graham algorithm
Finished graham algorithm
It took 0.002017974853515625s
Starting graham algorithm
Finished graham algorithm
It took 0.0009975433349609375s
Starting graham algorithm
Finished graham algorithm
It took 0.0010099411010742188s
Starting graham algorithm
Finished graham algorithm
It took 0.0s
```

## 2) Algorytm Jarvisa

Algorytm Jarvisa również dla wszystkich zbiorów działał poprawnie. Nie obyło się jednak bez komplikacji. W przypadku prostokąta i kwadratu trzeba było rozważyć sytuację, gdy istnieją dwa lub więcej punktów o takim samym kącie. W takim wypadku postępowaliśmy podobnie jak w algorytmie Grahama – braliśmy ten punkt, który jest najbardziej oddalony od ostatniego punktu należącego do otoczki. Po dodaniu tej modyfikacji, dostaliśmy poprawny wynik dla wszystkich zbiorów.

Wyznaczone otoczki zamieszczam poniżej:





Jak widać wyniki są identyczne jak te, wyznaczone przez algorytm Grahama. Jednak już na tym etapie możemy zaobserwować przewagę algorytmu Grahama pod względem czasu wykonania programu.

```

Starting jarvis algorithm
Finished jarvis algorithm
It took 0.006967782974243164s
Starting jarvis algorithm
Finished jarvis algorithm
It took 0.01198720932006836s
Starting jarvis algorithm
Finished jarvis algorithm
It took 0.0019757747650146484s
Starting jarvis algorithm
Finished jarvis algorithm
It took 0.000997781753540039s

```

### 3) Wybór zbiorów

Moim zdaniem zbiory z zadania zostały wybrane nieprzypadkowo:

- a) Losowy zbiór punktów: typowa sytuacja, najlepszy przypadek do wizualizacji działania algorytmu.
- b) Okrąg: sytuacja brzegowa, w której każdy punkt zbioru powinien jednocześnie należeć do jego otoczki. Najbardziej pesymistyczny przypadek dla algorytmu Jarvisa.
- c) Prostokąt: Sytuacja w której wiele punktów jest współliniowych, wiemy jednak, że jedynie punkty w rogach prostokąta powinny należeć do otoczki
- d) Kwadrat z przekątnymi: zaledwie 4 punkty powinny należeć do otoczki (wierzchołki). Dobry przykład dla zawężania zbioru punktów przez algorytm Grahama.

### 4) Porównanie czasu wykonania algorytmów

Poniżej zamieszczam porównanie czasów wykonania dla dużych zbiorów (10000 – 100000 punktów). Widać wyraźnie, że algorytm Grahama wygrywa w tym zestawieniu, nawet dla kwadratu (najlepszy przypadek dla Jarvisa, zaledwie 4 punkty w otoczce).

|                            |                               |
|----------------------------|-------------------------------|
| random points              | random circle                 |
| Starting jarvis algorithm  | Starting jarvis algorithm     |
| Finished jarvis algorithm  | Finished jarvis algorithm     |
| It took 19.52378821372986s | It took 1.087092399597168s    |
| Starting graham algorithm  | Starting graham algorithm     |
| Finished graham algorithm  | Finished graham algorithm     |
| It took 2.075512409210205s | It took 0.009018898010253906s |

|                             |                             |
|-----------------------------|-----------------------------|
| random rectangle            | random square               |
| Starting jarvis algorithm   | Starting jarvis algorithm   |
| Finished jarvis algorithm   | Finished jarvis algorithm   |
| It took 4.747373342514038s  | It took 0.5516774654388428s |
| Starting graham algorithm   | Starting graham algorithm   |
| Finished graham algorithm   | Finished graham algorithm   |
| It took 0.7949461936950684s | It took 0.2494058609008789s |