

PROJEKT ALGORYTMY GEOMETRYCZNE

Adam Dyda
Norbert Wolniak

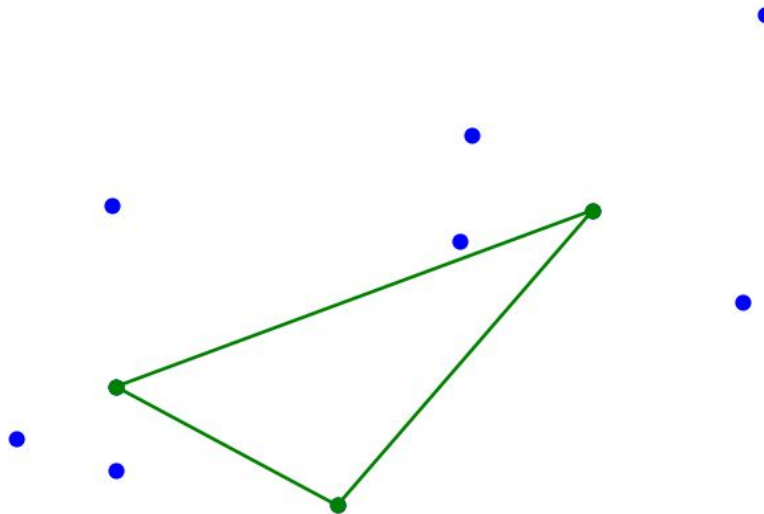
Algorytmy wyznaczania otoczki wypukłej

1. Wprowadzenie.

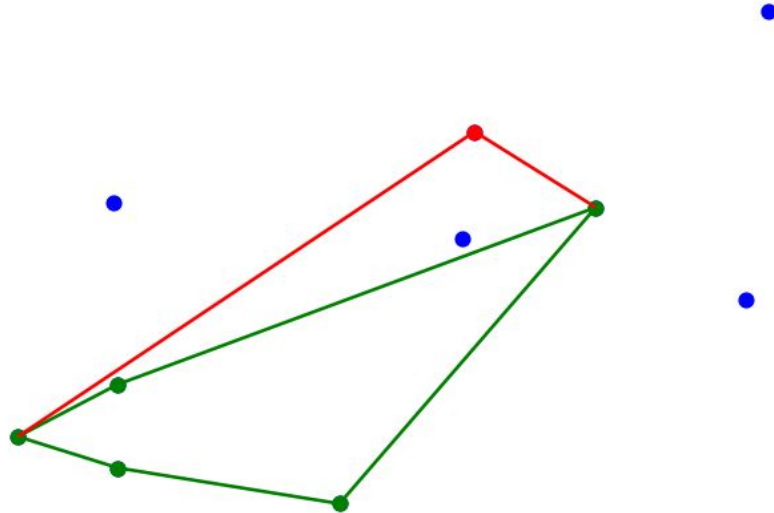
W ramach projektu zaimplementowaliśmy algorytmy do wyznaczania otoczki wypukłej w przestrzeni dwuwymiarowej. Algorytmy : Przyrostowy , Górna i dolna otoczka, Grahama, Jarvisa, Dziel i rządź, QuickHull i Chana.

2. Algorytm Przyrostowy

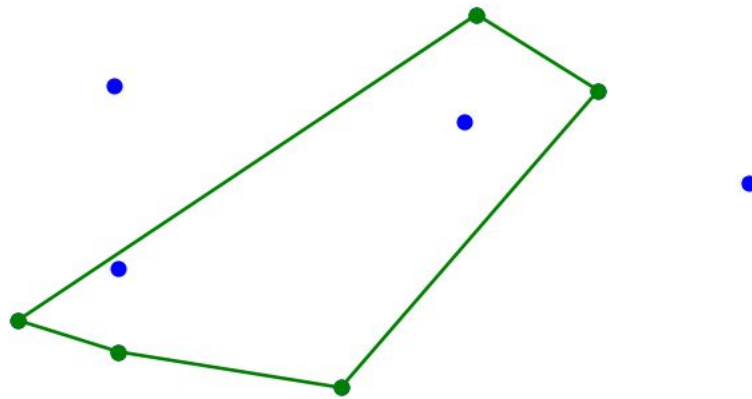
- Weź 3 pierwsze punkty ze zbioru S i stwórz z nich otoczkę $CH(S)$



- Dla reszty punktów jeśli punkt nie należy do wnętrza $CH(S)$
- Znajdź styczne tego punktu do otoczki $CH(S)$

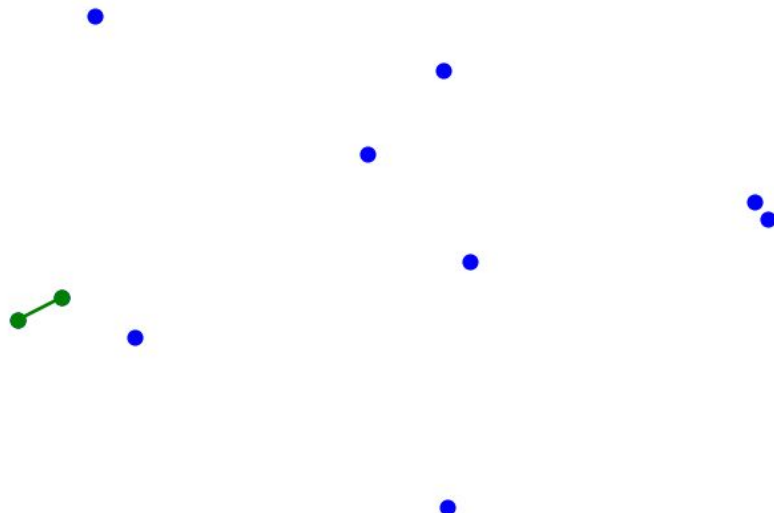


- Zaktualizuj otoczkę CH(S)

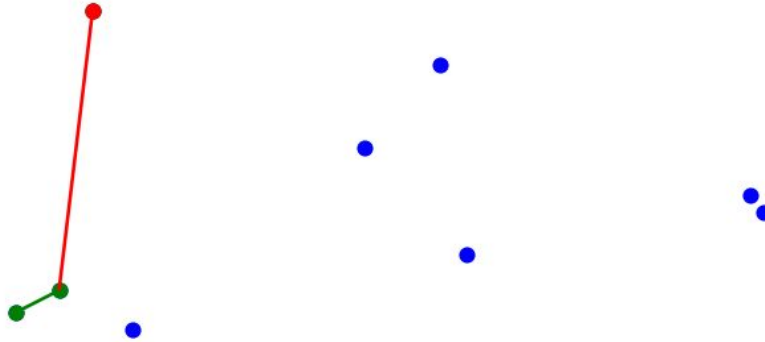


3. Algorytm Górnej i Dolnej otoczki

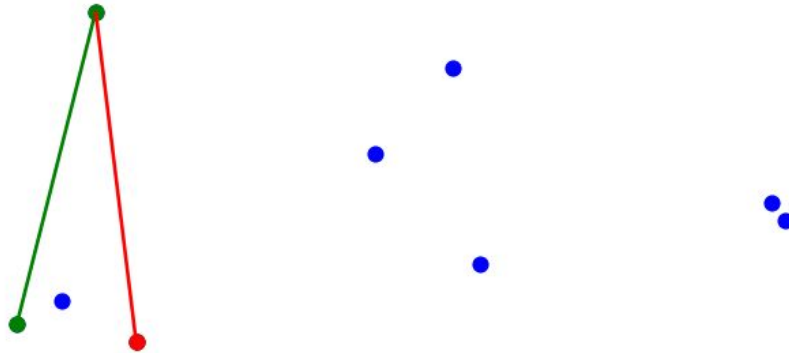
- Uporządkuj punkty w porządku leksykograficznym.
- Stwórz górny łańcuch $L_{\text{Górna}}$ z początkowych dwóch punktów p_1 , p_2



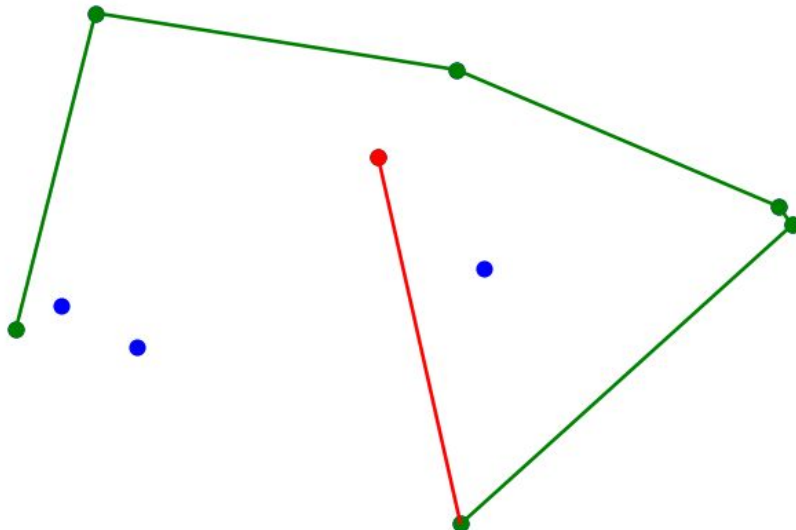
- For $i = 3$ to n
- Dodaj p_i do górnego łańcucha



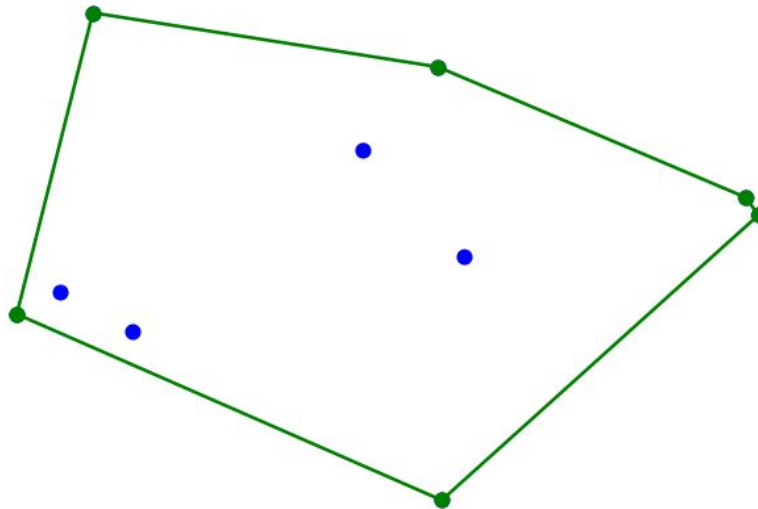
- Dopóki górny łańcuch zawiera więcej niż dwa punkty i trzy ostatnie nie tworzą skrętu w prawo : usuń środkowy z ostatnich trzech punktów $L_{\text{Górna}}$



- Analogicznie wyznacz dolny łańcuch L_{Dolny}

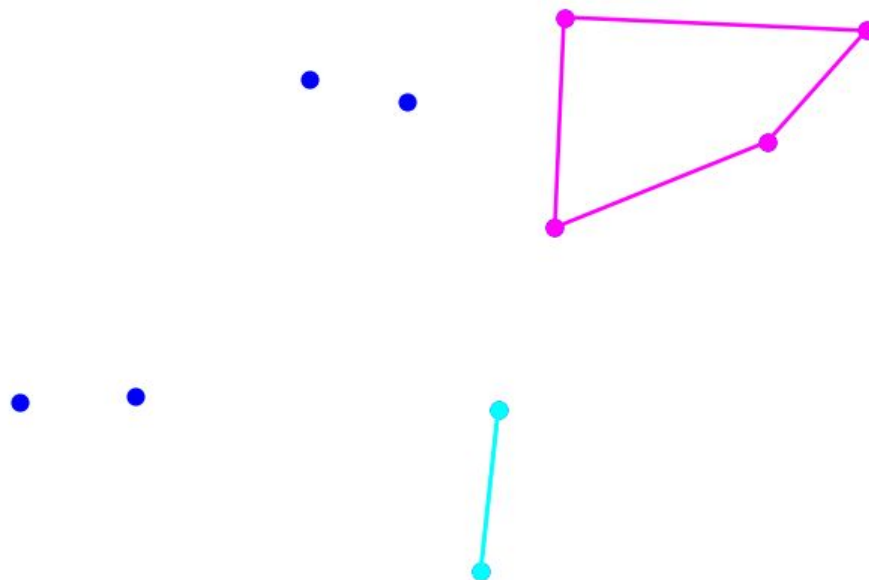


- Złącz oba łańcuchy

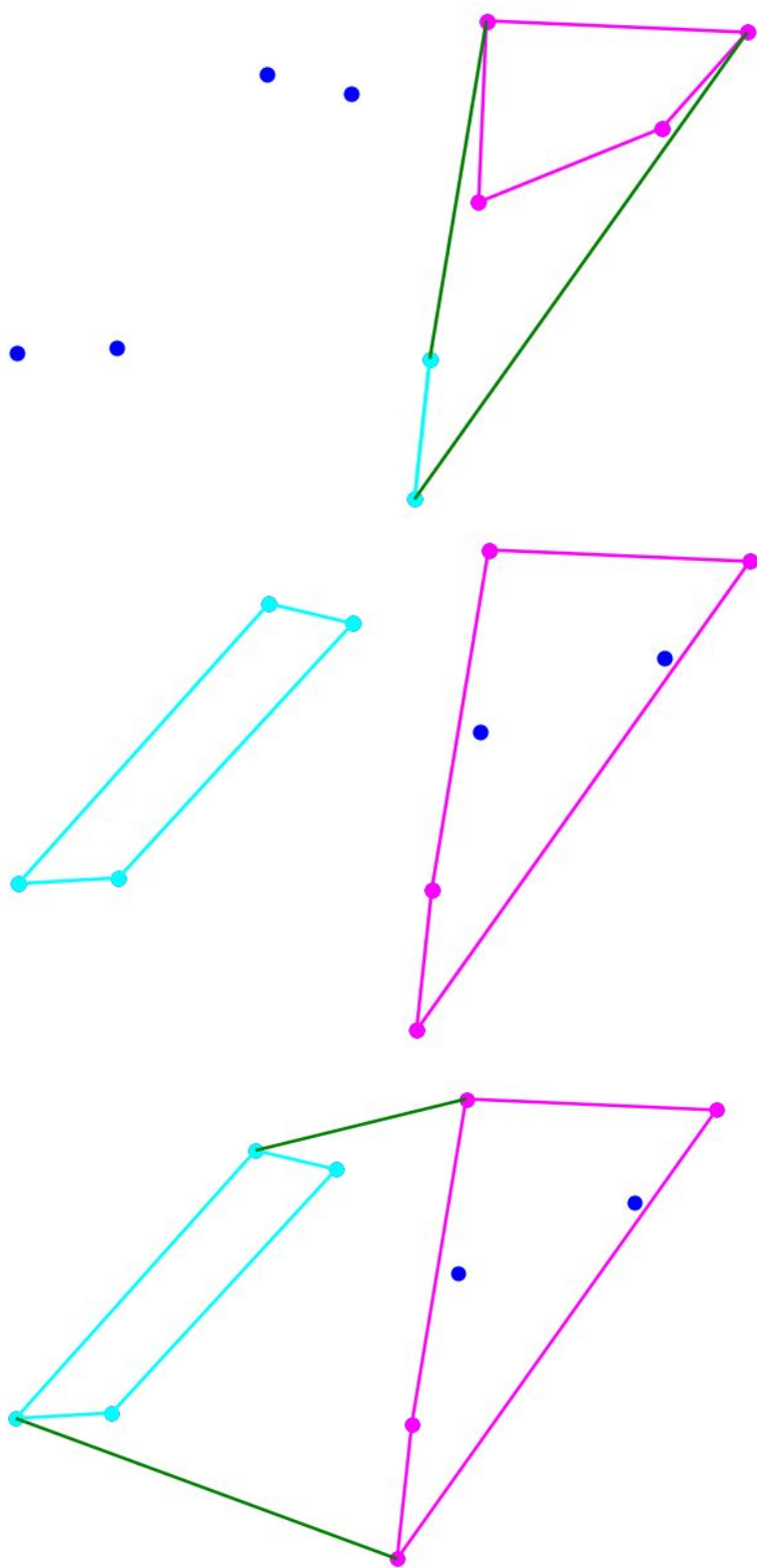


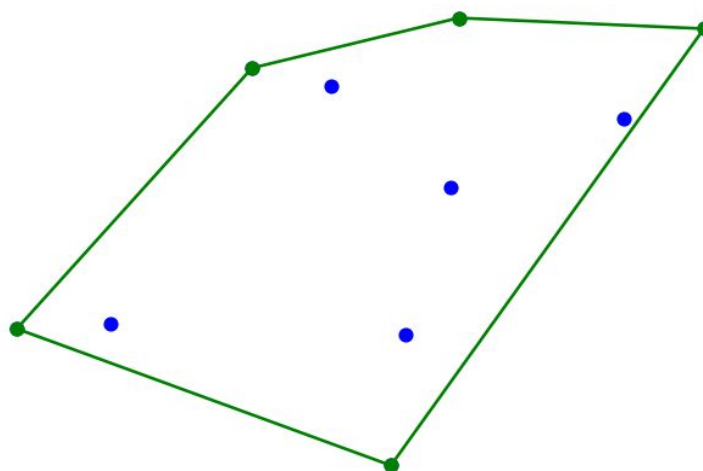
4. Algorytm Dziel i Rządź

- Dopóki rozmiar dowolnego zbioru F przekracza daną stałą K
- Dziel zbiory względem mediany współrzędnych x -owych punktów
- Znajdź otoczki wypukłe zbiorów o rozmiarze mniejszym niż K
- Dopóki otoczka zbioru wejściowego S nie została znaleziona



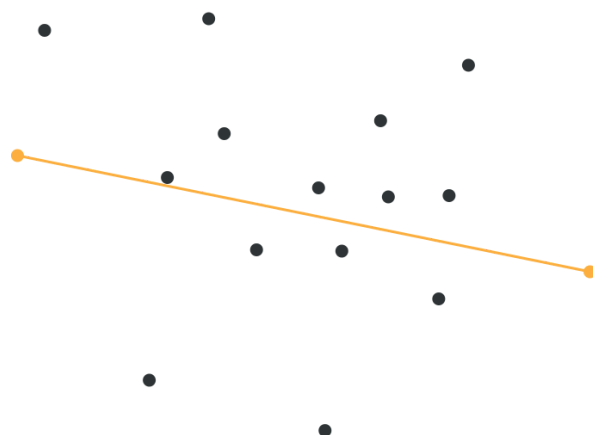
- Sklejaj otoczki sąsiadujących zbiorów znajdując górną i dolną styczną



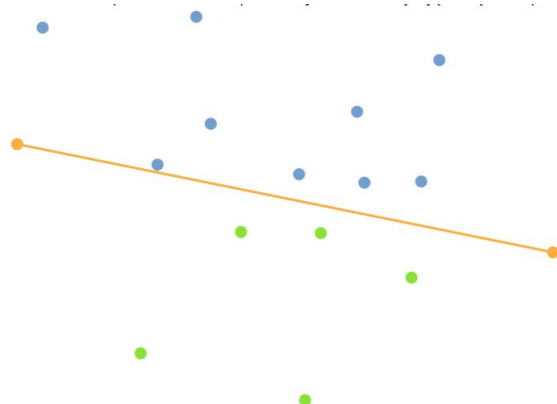


5. Algorytm QuickHull

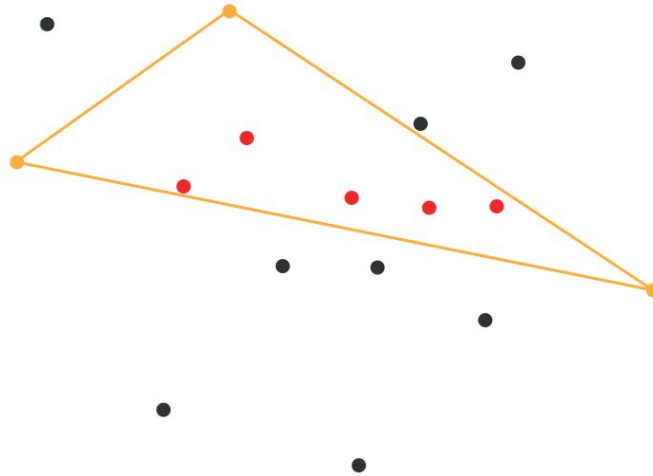
- Znajdź w zbiorze punktów dwa skrajne punkty o minimalnej i maksymalnej współrzędnej x (A i B)



- Podziel zbiór punktów na dwa podzbiory S1 i S2 znajdujące się nad i pod prostą AB



- Wywołaj rekurencyjnie QuickHull(S1,A,B) i QuickHull(S2,B,A), następnie
 - ❑ W funkcji QuickHull(S,A,B):
 - ❑ jeśli S jest puste -> return
 - ❑ jeśli S ma jeden element -> dodaj element do otoczki -> return
 - ❑ w innym wypadku: znajdź punkt C najbardziej oddalony od prostej AB, dodaj go do otoczki, odrzuć punkty z wnętrza trójkąta ABC

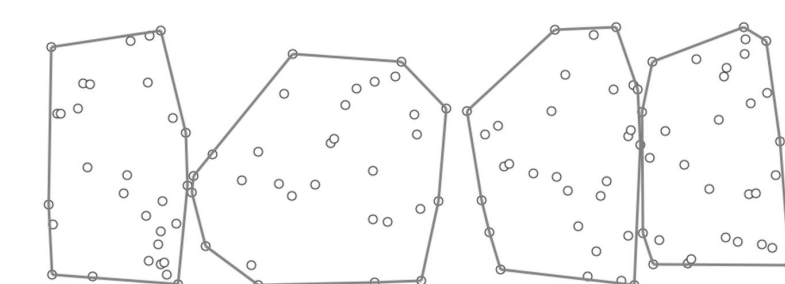


- ❑ znajdź zbiór punktów S1 znajdujący się po prawej stronie prostej AC oraz zbiór punktów S2 znajdujący się po prawej stronie BC
- ❑ wywołaj rekurencyjnie dla QuickHull(A,C,S1) i QuickHull(B,C,S2)

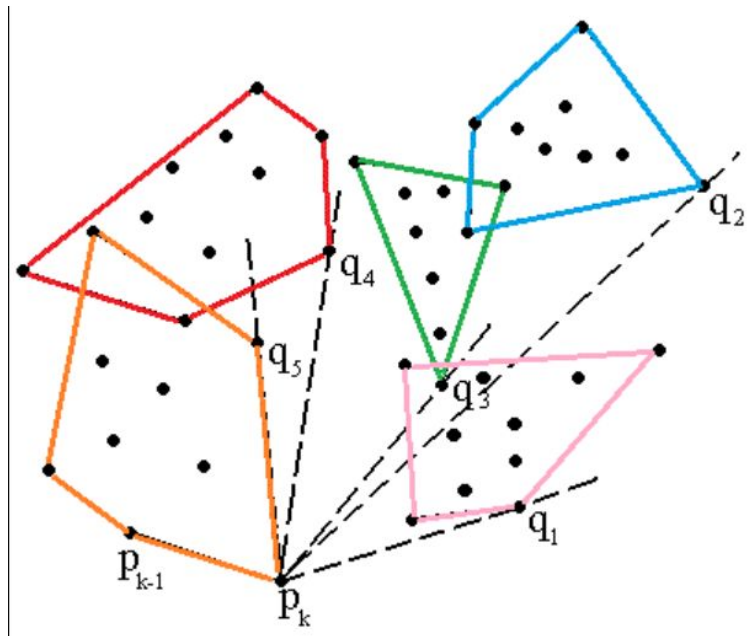
6. Algorytm Chana

Jedno przejście algorytmu wymaga znajomości wartości parametru m który oznacza ilość punktów otoczki. W naszym algorytmie sprawdzamy wartość tego parametru wykonując algorytm dla wartości od 0 do n gdzie n to ilość punktów w zbiorze.

- wyznaczamy punkt p który na pewno należy do otoczki, np. punkt o najmniejszej współrzędnej x .
- dzielimy n punktów zbioru wejściowego na $\frac{n}{m}$ grup o rozmiarze m
- dla każdej z tych grup wyznaczamy otoczkę za pomocą algorytmu grahama



- następnie dla każdej z tych otoczek wyznaczamy za pomocą wyszukiwania binarnego styczną z punktu do otoczki



- za pomocą algorytmu jarvisa znajdujemy styczną wyznaczającą największy kąt
- jeżeli znaleziony punkt nie jest punktem p, dodajemy go do otoczki w przeciwnym razie kończymy algorytm