Geometria Obliczeniowa

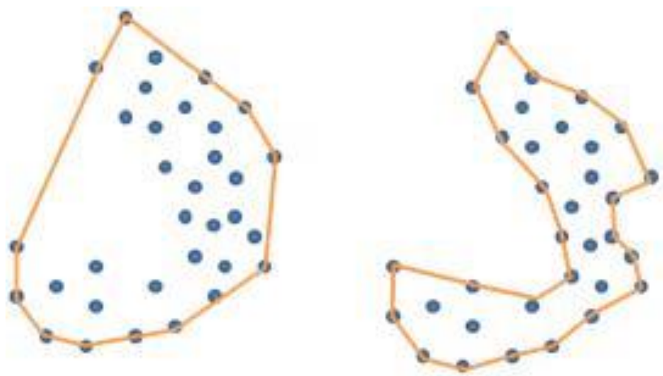
Otoczka wypukła

Prof. dr hab. inż. **Łukasz Madej**
Mgr inż. **Daniel Bachniak,** Mgr inż. **Mateusz Mojżeszko**
Katedra Informatyki Stosowanej i Modelowania
Wydział Inżynierii Metali i Informatyki Przemysłowej

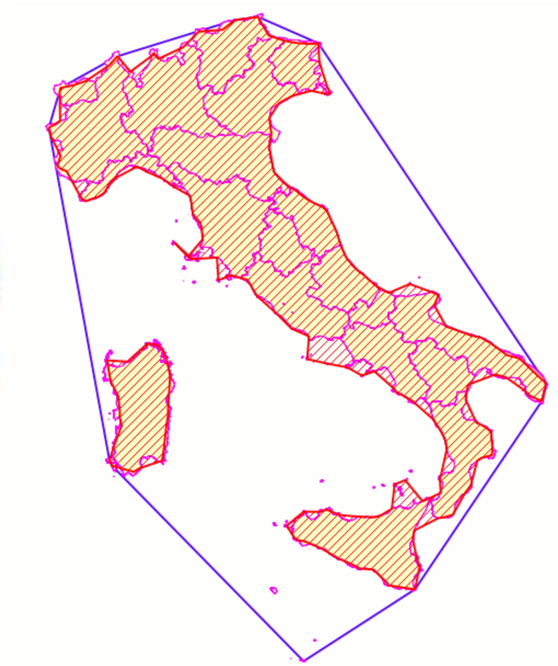
Budynek B5
p. 716
lmadej@agh.edu.pl
home.agh.edu.pl/lmadej

Otoczka wypukła

Wielokąt wypukły



Convex hull   Concave hull



Niech **A** będzie skończonym zbiorem punktów na płaszczyźnie.

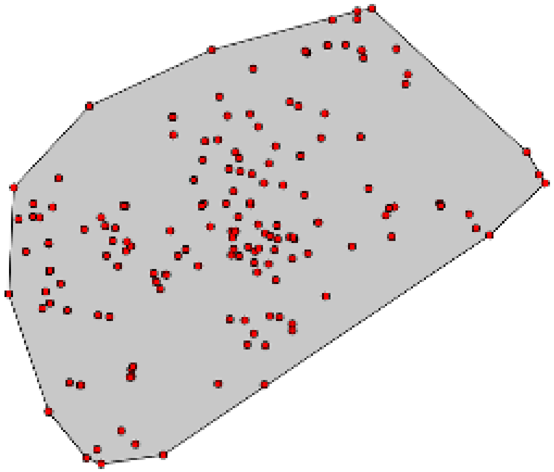
Otoczka wypukła: najmniejszy zbiór wypukły taki, że każdy punkt zbioru **A** znajduje się albo w jego wnętrzu albo na brzegu (convex hull).

1. **Algorytm Jarvisa - dwa warianty**

2. **Algorytm Grahama**

3. **Metoda trójkątów**

4. **Algorytm Quickhull**



Algorytmy wykorzystywane w wymienionych metodach:

- obliczanie kąta między wektorami (iloczyn skalarny),
- położenie punktu względem wektora,
- test przynależności punktu do odcinka,
- odległość punktu od prostej.

https://grass.osgeo.org/grass76/manuals/addons/v.concave.hull.html

Algorytm Jarvisa

Otoczka wypukła

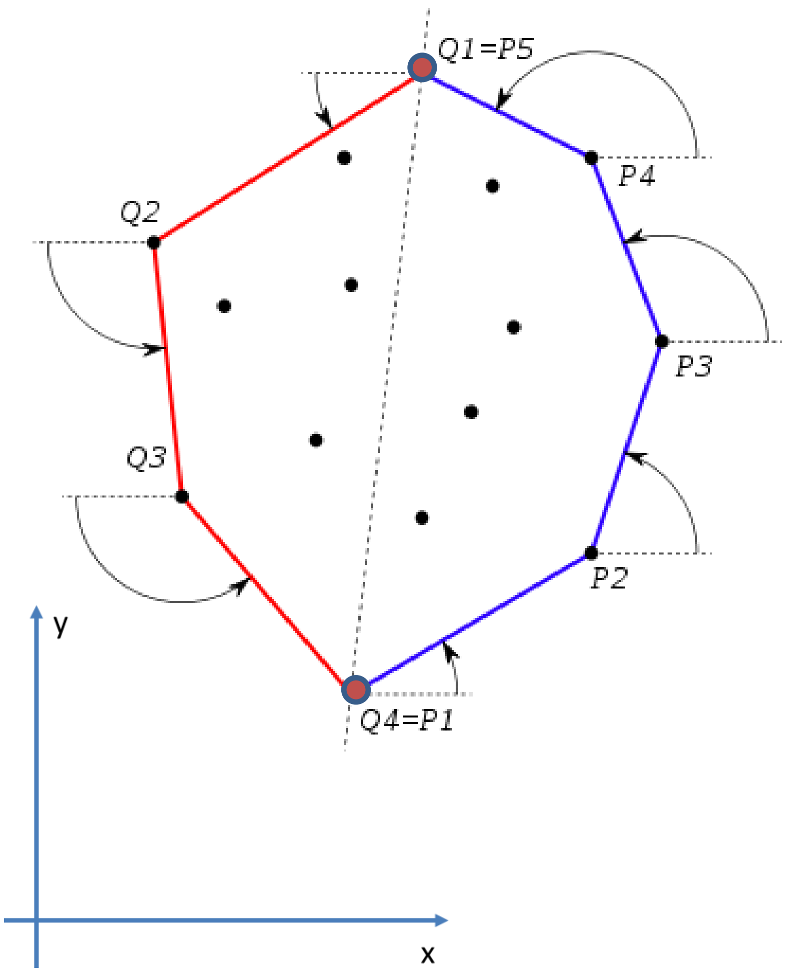
Wariant 1 algorytmu Jarvisa (1)

Krok 1:

P1 – wyznaczenie punktu o **najmniejszej współrzędnej y** (jeśli jest więcej niż jeden wybierany jest punkt o najmniejszej współrzędnej x).

Krok 2:

Q1 – wyznaczenie punktu o **największej współrzędnej y** (jeśli jest więcej niż jeden wybierany jest punkt o największej współrzędnej x).

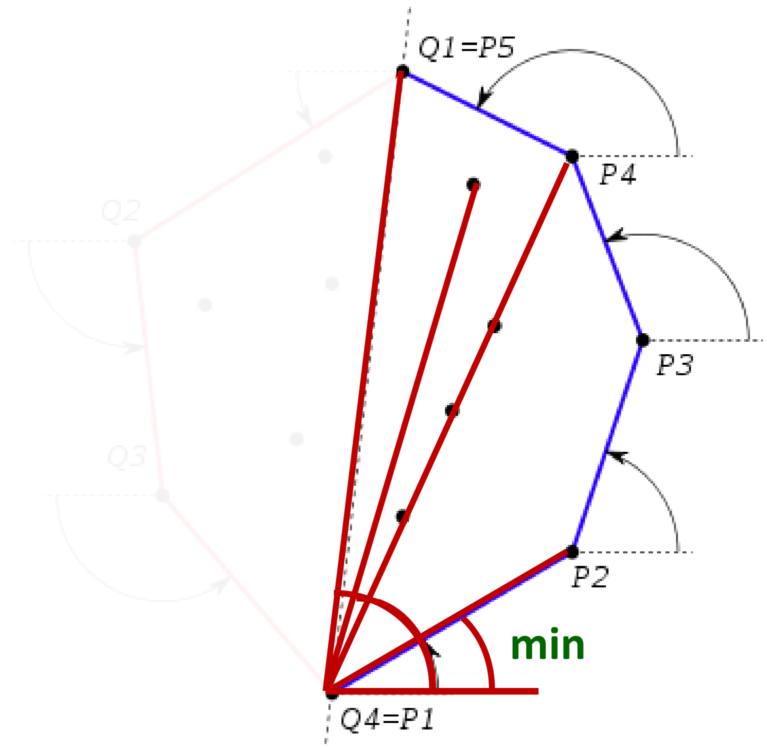


Otoczka wypukła

Wariant 1 algorytmu Jarvisa (2)

Krok 3 (wyznaczenie prawego łańcucha):

i = 1
powtarzaj:
    S = Pi <- „najnowszy” punkt otoczki
    N = **znajdź punkt**, który tworzy
        najmniejszy kąt między wektorem [1, 0]
        a wektorem SN
    Pi+1 = N
    jeśli N = Q1: koniec iterowania
    i = i + 1



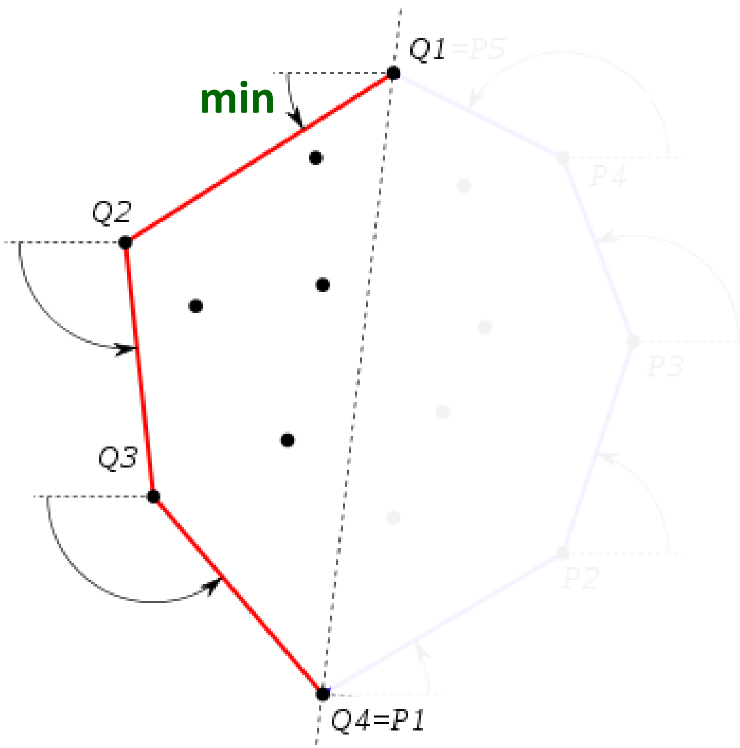
https://pl.wikipedia.org

Otoczka wypukła

Wariant 1 algorytmu Jarvisa (3)

Krok 4 (wyznaczenie lewego łańcucha):

i = 1
powtarzaj:
  S = Qi   <- „najnowszy" punkt otoczki
  N = **znajdź punkt**, który tworzy
    najmniejszy kąt między wektorem [-1, 0]
    a wektorem SN
  Qi+1 = N
  jeśli N = Q4: koniec iterowania
  i = i + 1



Otoczka wypukła

Wariant 2 algorytmu Jarvisa (1)

Tworzenie całej otoczki, brak podziału na 2 etapy.

Krok 1:

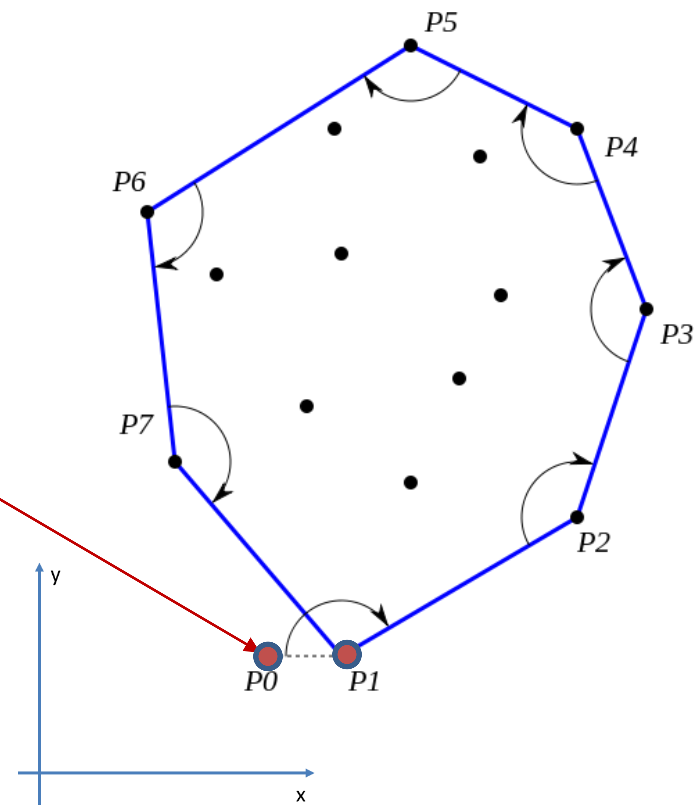
P1 – wyznaczenie punktu o **najmniejszej współrzędnej y** (jeśli jest więcej niż jeden wybierany jest punkt o najmniejszej współrzędnej x).

Zdefiniuj punkt: $P0 = (-\infty, P1.y)$

Krok 2:

i = 1
powtarzaj:
  Pi+1 <- nowy punkt otoczki taki, że **kąt** pomiędzy punktami Pi-1,Pi,Pi+1 **jest największy**

  jeśli Pi+1 = P1: koniec iterowania
  i = i + 1

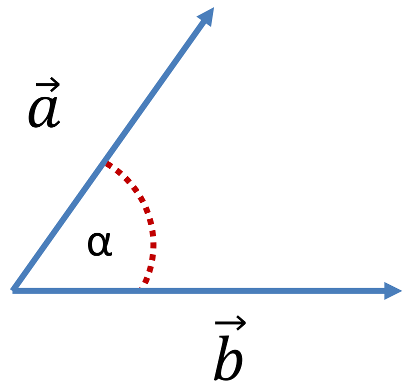


https://pl.wikipedia.org

Kąt między wektorami

Iloczyn skalarny

$$\vec{a}\circ\vec{b} = a_1b_1 + a_2b_2 + \cdots + a_nb_n$$



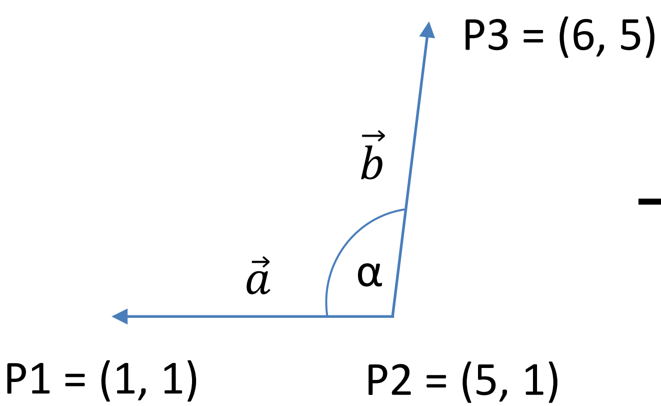
$$\vec{a}\circ\vec{b} = |\vec{a}| \cdot |\vec{b}| \cdot \cos(\alpha)$$

$\vec{a}\circ\vec{b}$ – iloczyn sklarny wektorów $\vec{a}$ i $\vec{b}$

$|\vec{a}|, |\vec{b}|$ – długość wektora $\vec{a}$ i $\vec{b}$

$$\cos(\alpha) = \frac{\vec{a}\circ\vec{b}}{|\vec{a}| \cdot |\vec{b}|}$$

Zadanie



$$\vec{a} = [1-5, 1-1] = [-4, 0]$$

$$\vec{b} = [6-5, 5-1] = [1, 4]$$

$$|\vec{a}| = \sqrt{(-4)^2+0^2} = 4$$

$$|\vec{b}| = \sqrt{1^2+5^2} \approx 5.1$$

$$|\vec{a}|\circ|\vec{b}| = (-4) * 1 + 0 * 4 = -4$$

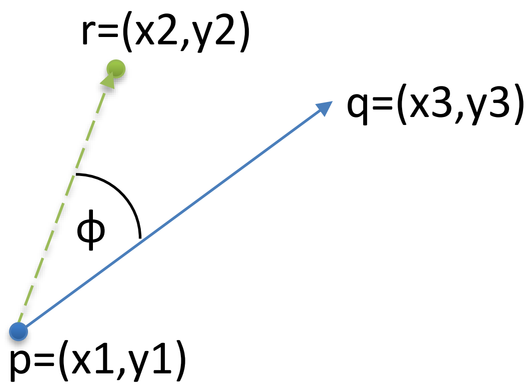
$$\cos(\alpha) = \frac{-4}{20.4} \approx -0.196$$

$$\alpha = \arccos(-0.196) \approx 101$$

Położenie punktu względem wektora (1)

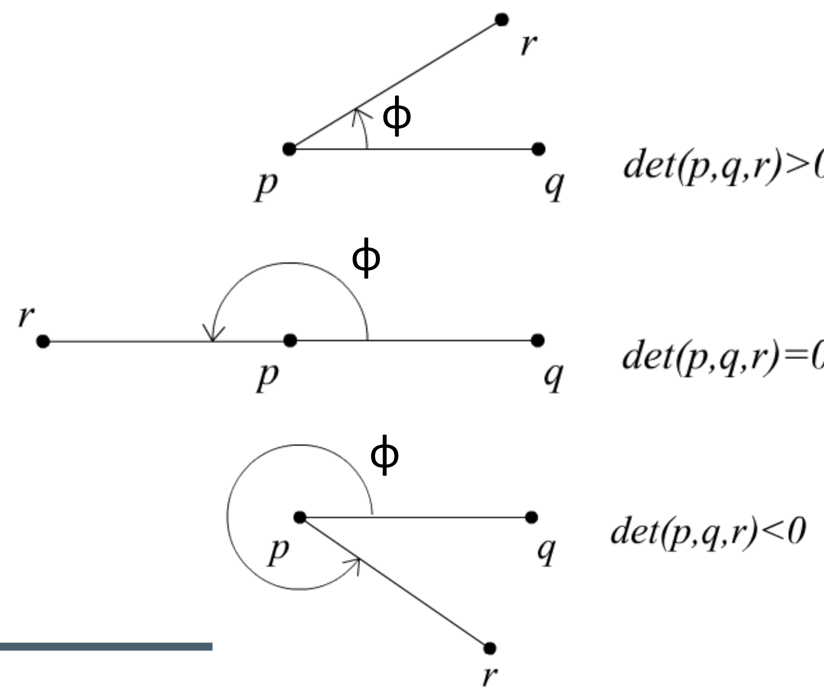
Po której stronie wektora pq znajdujesię punkt r?

Przypomnienie:
Wykład 1: z równania ogólnego prostej



Z wyznaczników

$$\det(p,q,r) = \begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix}$$



- Jeżeli **det(p,q,r) > 0** to sinϕ > 0 i punkt r leży **po lewej** stronie odcinka pq.
- Jeżeli **det(p,q,r) = 0** to sinϕ = 0 i punkty p, q i r są **współliniowe**.
- Jeżeli **det(p,q,r) < 0** to sinϕ < 0 i punkt r leży **po prawej** stronie odcinka pq.

Przypomnienie

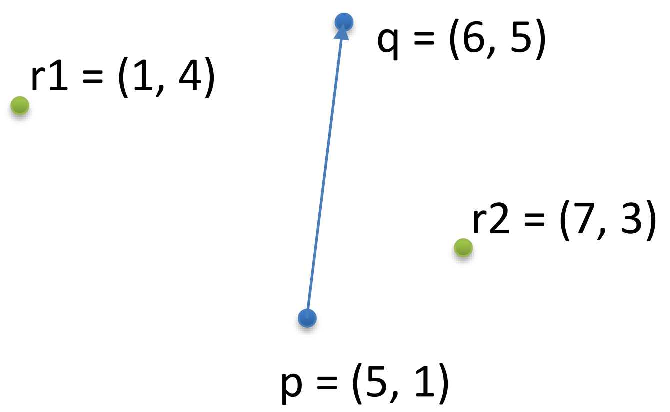
$$A=\begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{bmatrix}$$

$$Det\left(A\right)=x_{11}x_{22}x_{33}+x_{12}x_{23}x_{31}+x_{21}x_{32}x_{13}-\left(x_{13}x_{22}x_{31}+x_{32}x_{23}x_{11}+x_{21}x_{12}x_{33}\right)$$

$$A=\begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{bmatrix} \qquad A=\begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{bmatrix}$$

Położenie punktu względem wektora (2)

Przykład – zadanie….



det(p,q,r1) = 19 > 0: po lewej stronie

det(p,q,r2) = -6 < 0: po prawej stronie

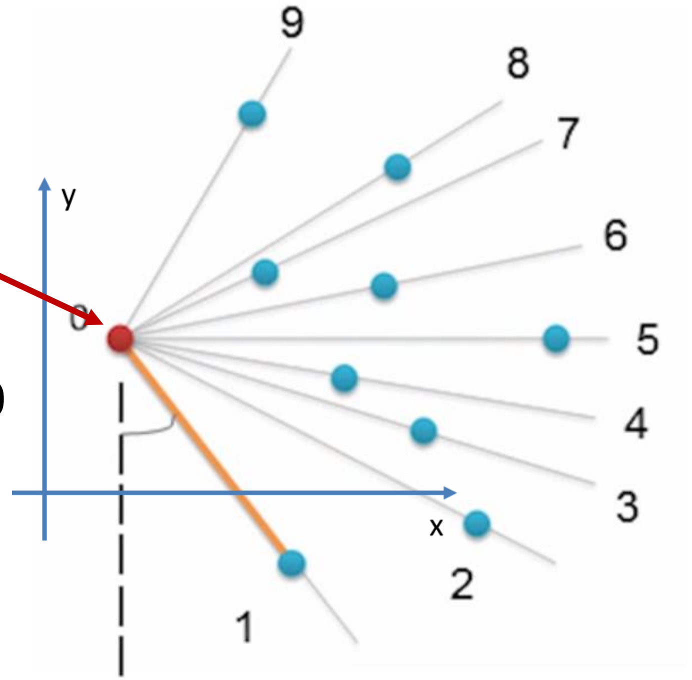
Algorytm Grahama

Otoczka wypukła

Algorytm Grahama (1)

Krok 1:

Punkt 0 – wyznaczenie punktu o **najmniejszej współrzędnej x** (jeśli jest więcej niż jeden wybierany jest punkt o najmniejszej współrzędnej y).



Convex Hull Algorithm Presentation for CSC 335 (Analysis of Algorithms) at TCNJ.

Krok 2:

Przesunięcie wszystkich punktów tak aby punkt 0 pokrył się z początkiem układu współrzędnych.

Krok 3:

Posortowanie punktów względem kąta pomiędzy punktem 0 a pionowym wektorem skierowanym w dół. **Kierunek** sortowania – **przeciwny do kierunku ruchu wskazówek zegara**. Jeśli jest kilka punktów o tym samym kącie to uwzględniana jest również odległość od punktu 0.

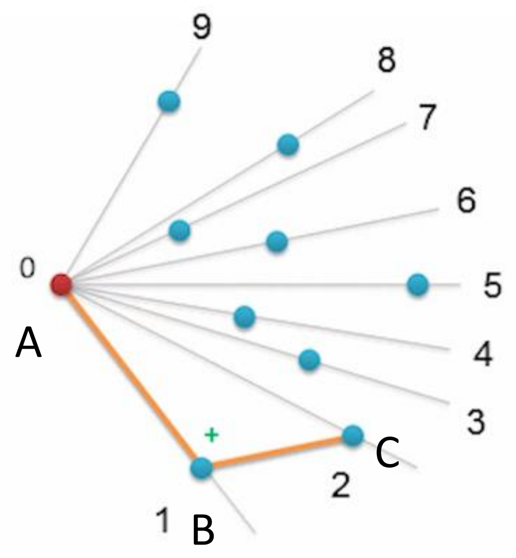
Otoczka wypukła

Algorytm Grahama (2)

Krok 4 (powtarzany do końca algorytmu):

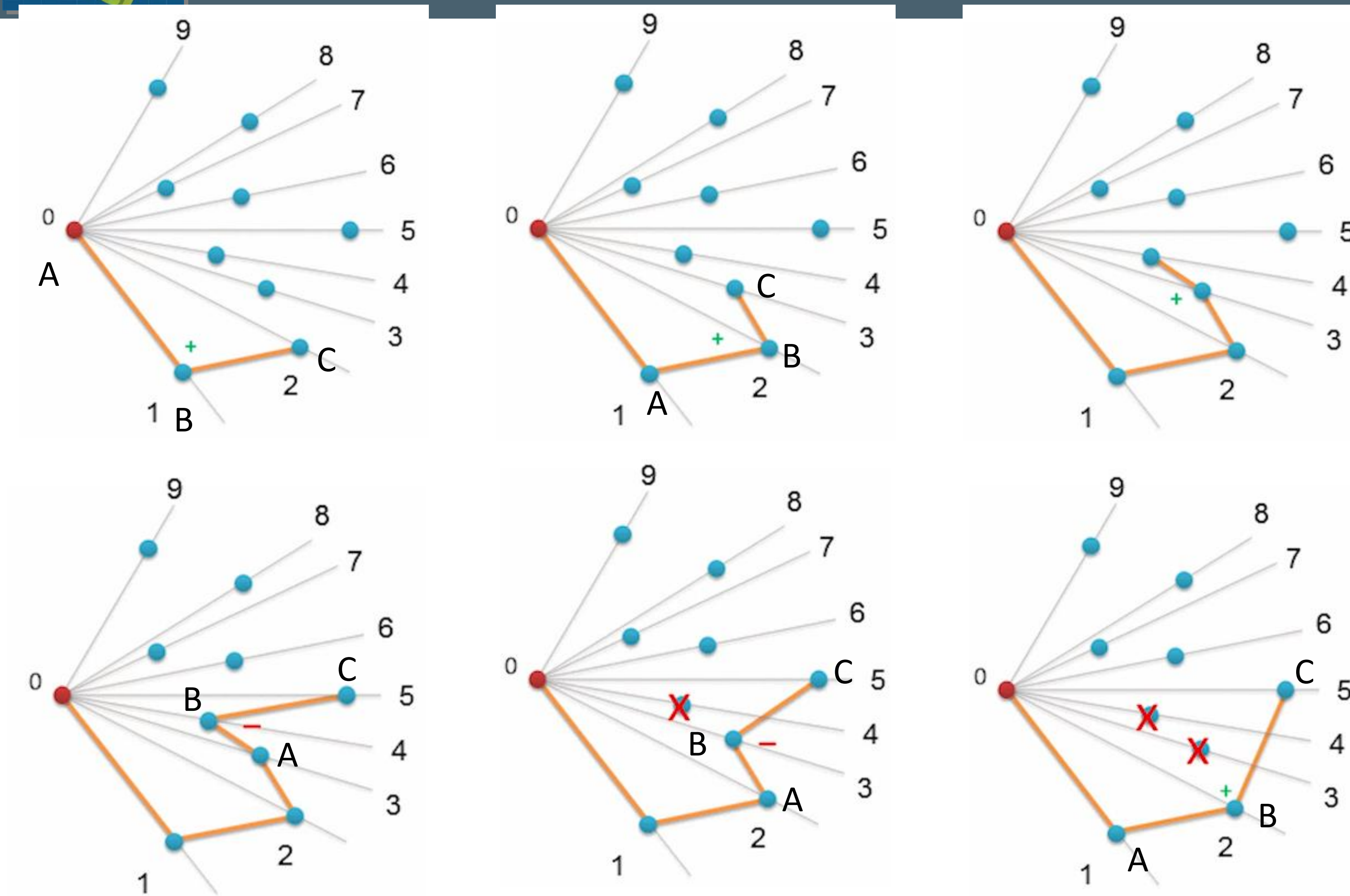
Dobieranie kolejnych punktów z posortowanej listy.
Jeśli oznaczyć 3 ostatnio dobrane punkty jako A,B,C to wykonujemy test względnego położenia tych 3 punktów.

- Jeśli punkt B leży po lewej stronie wektora AC:
  to punkt B nie należy do otoczki wypukłej,
- Jeśli punkt B leży po prawej stronie wektora AC:
  to punkt B należy do otoczki wypukłej.



Otoczka wypukła

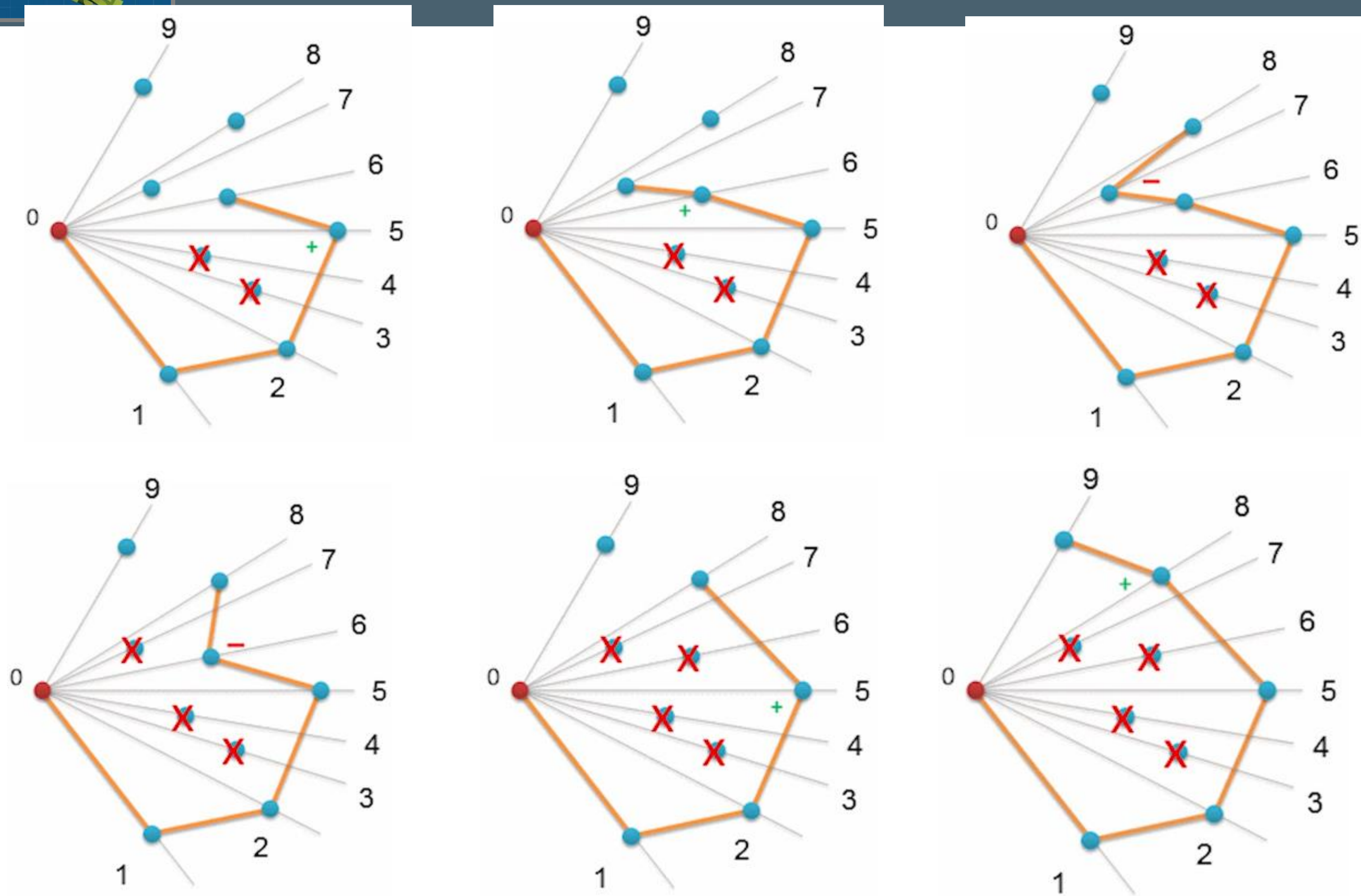
Algorytm Grahama (3)



Convex Hull Algorithm Presentation for CSC 335 (Analysis of Algorithms) at TCNJ.

Otoczka wypukła

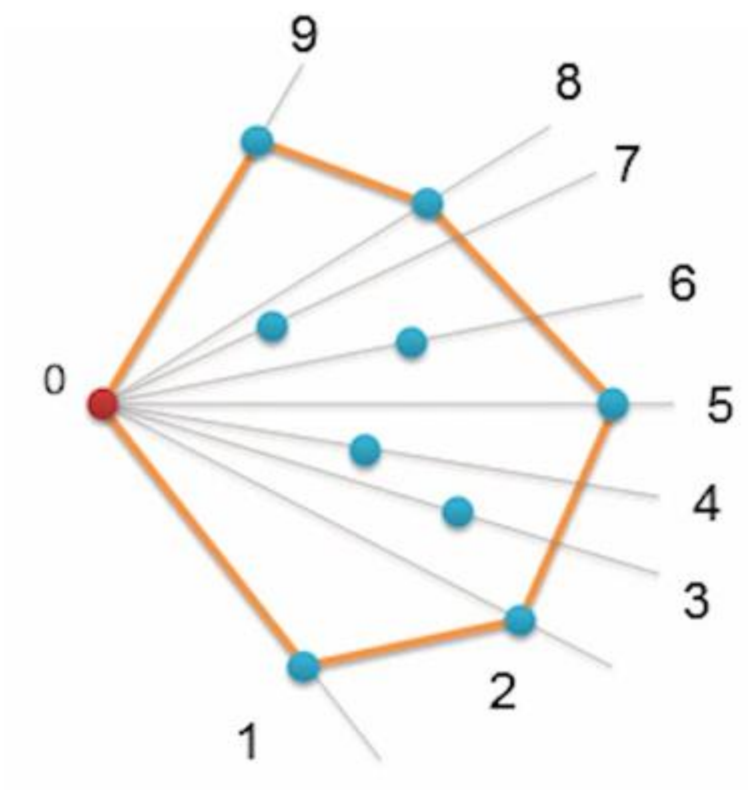
Algorytm Grahama (4)



Convex Hull Algorithm Presentation for CSC 335 (Analysis of Algorithms) at TCNJ.

Otoczka wypukła

Algorytm Grahama (5)



Metoda trójkątów

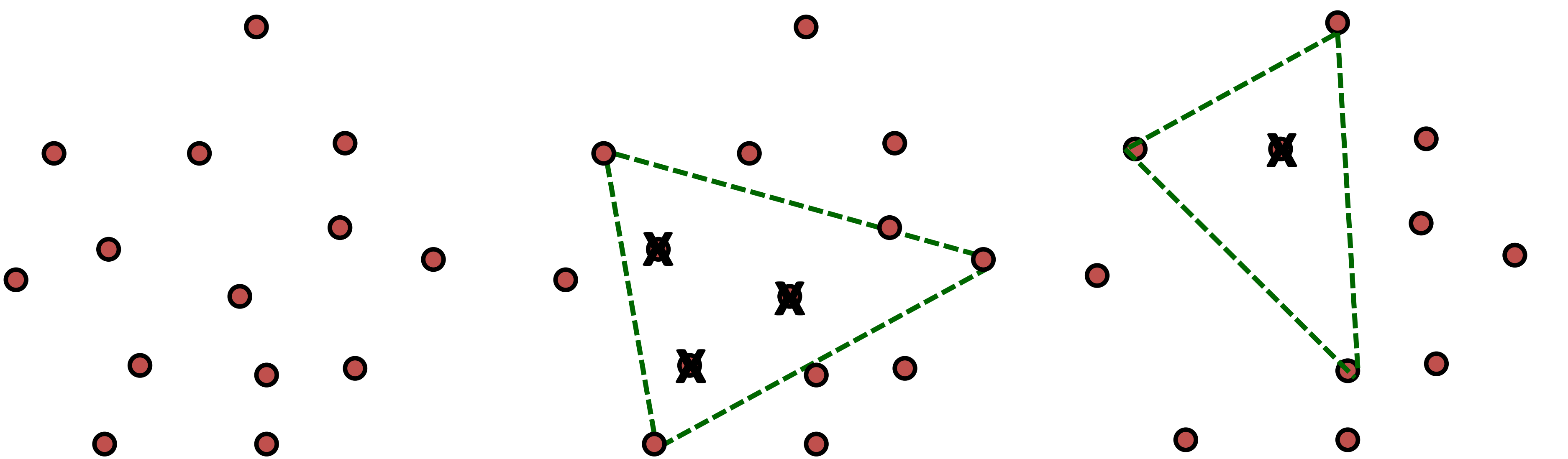
Otoczka wypukła

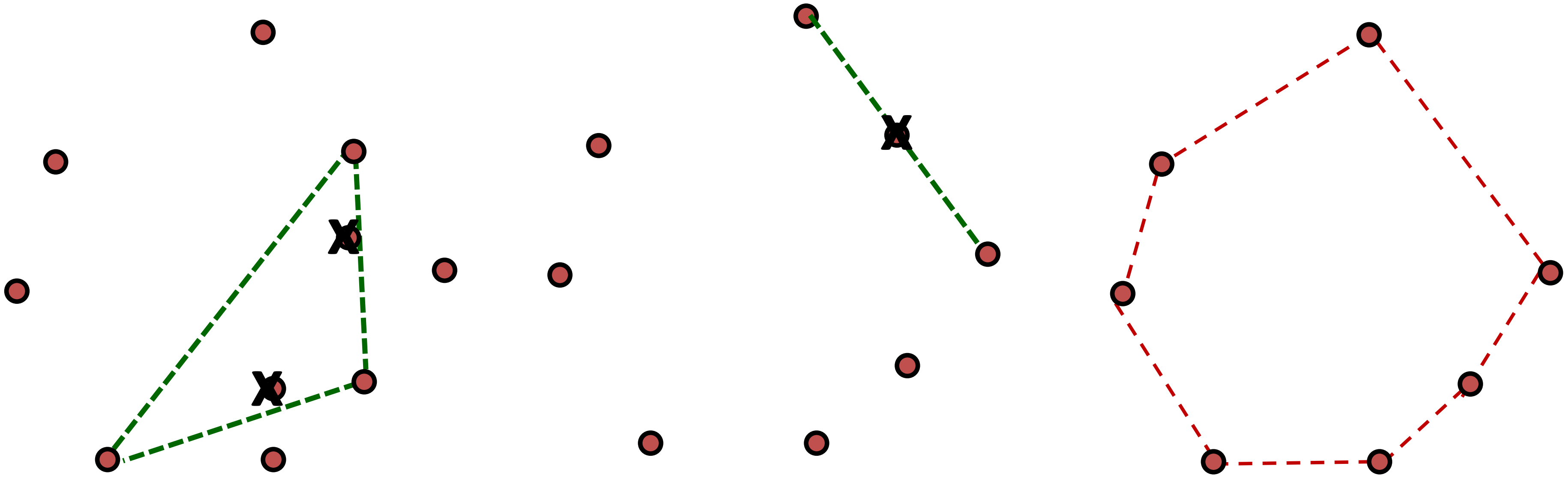
Metoda trójkątów

X – zbiór wszystkich punktów

Dla każdej trójki punktów p_1, p_2, p_3 ze zbioru X:
 - jeżeli nie są one współliniowe (**wykład 2**), to **usuń wszystkie punkty** zbioru X leżące **wewnątrz trójkąta** (p_1,p_2,p_3) (**wykład 3**),
 - jeżeli są **współliniowe**, to usuń ze zbioru X punkt środkowy (z tej trójki).

Punkty, które pozostały tworzą zbiór wierzchołków otoczki wypukłej zbioru X.



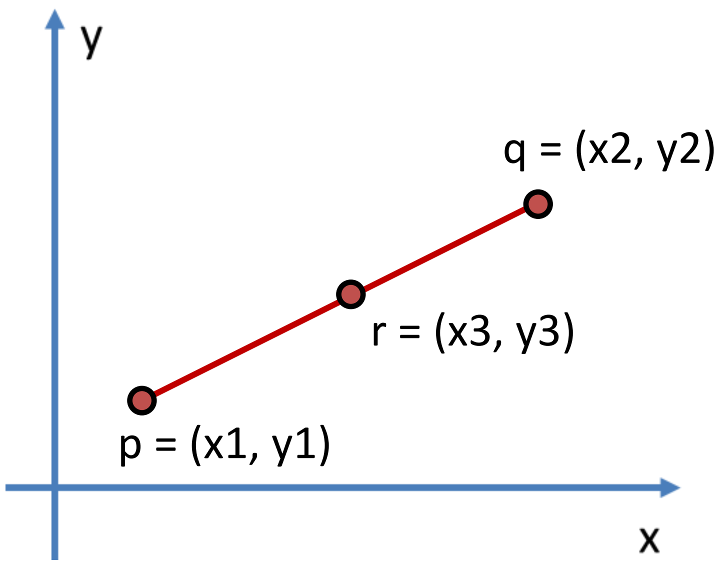


Test przynależności punktu do odcinka

Z wyznacznika

Dane są trzy punkty: p = (x1, y1), q = (x2, y2), r = (x3, y3).
Problem: **czy punkt r należy do odcinka pq?**

$$\det(p,q,r) = \begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix}$$



- Jeśli punkty p, q i r **są współliniowe**:

det (p,q,r) = 0

oraz

- Jeśli r znajduje się „**pomiędzy końcami odcinka**” pq:

(p.x <= **r.x** <= q.x lub q.x <= **r.x** <= p.x)

i

(p.y <= **r.y** <= q.y lub q.y <= **r.y** <= p.y)

- To punkt r należy do odcinka pq.

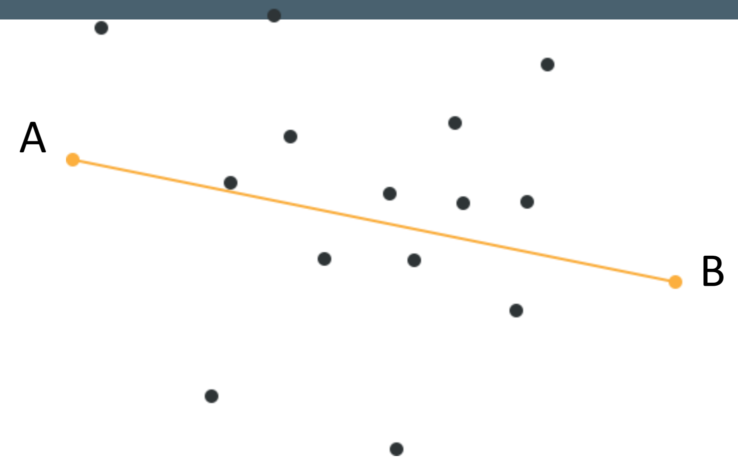
Algorytm Quickhull

Otoczka wypukła

Quickhull – algorytm typu dziel i zwyciężaj (1)

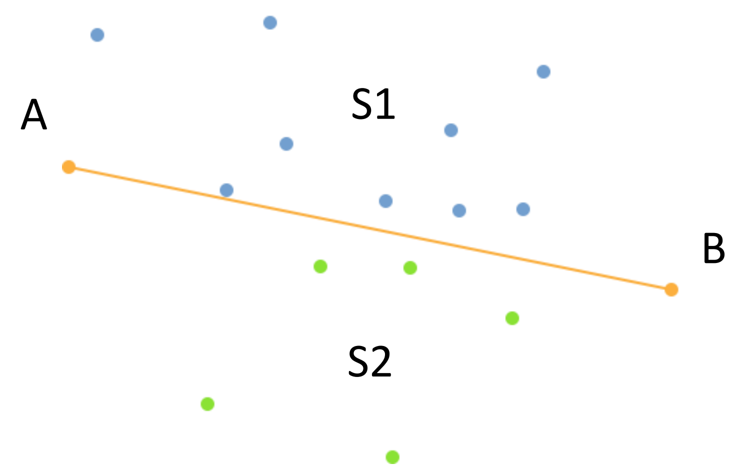
Krok 1:

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



Krok 2:

Podziel zbiór punktów na **dwa podzbiory** S1 i S2 znajdujące się nad i pod prostą A i B



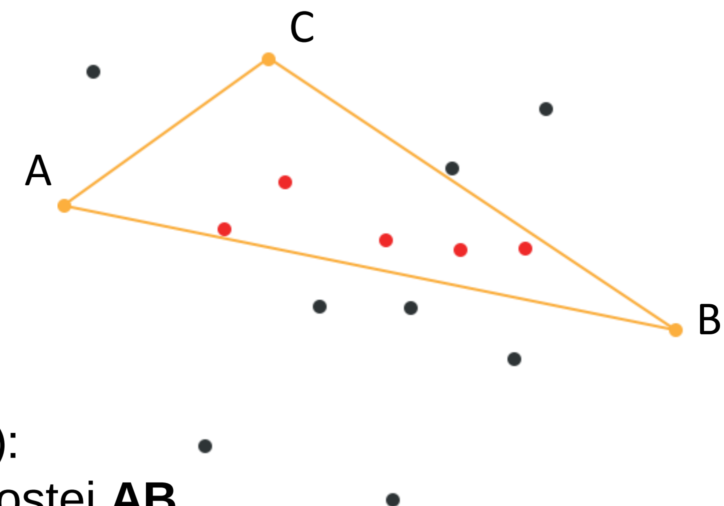
Krok 3:

Wywołaj rekurencyjnie **QuickHull(A, B, S1)** i **QuickHull(B, A, S2)**

Otoczka wypukła

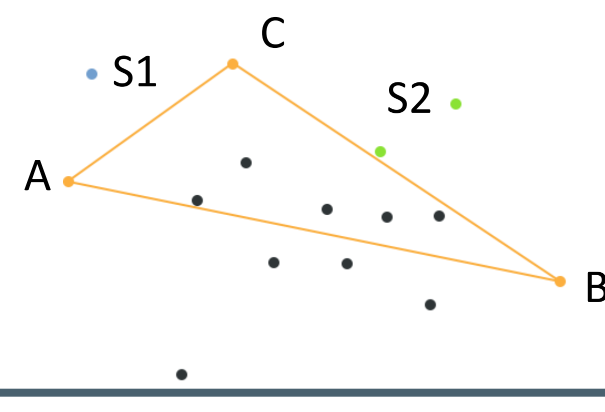
Quickhull – algorytm typu dziel i zwyciężaj (2)

Zamiast wyznaczać dwa skrajne punkty, można uwzględnić np. trzy skrajne, otrzymując trójkąt i od razu odrzucić wszystkie punkty należące do wnętrza figur. Wówczas procedurę QuickHull należy wywołać dla każdego boku figury, uprzednio dzieląc odpowiednio zbiór punktów.:

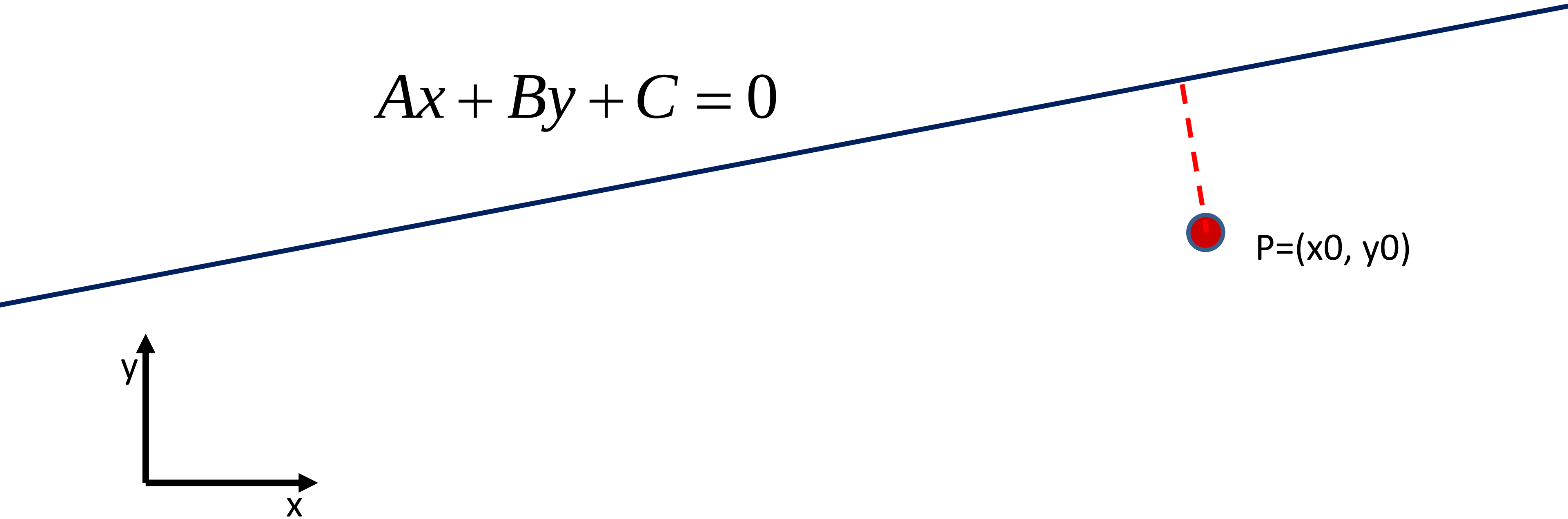


QuickHull(A, B, S1)

- Jeśli S1 (S2) **jest pusty**: **koniec,**
- Jeśli S1 (S2) ma **jeden punkt**:
  należy on do otoczki wypukłej,
- W przeciwnym przypadku (punktów jest więcej):
  - znajdź **punkt najbardziej oddalony** od prostej **AB**
  - **odrzuć** wszystkie **punkty z wnętrza trójkąta** ABC (nie należą do otoczki)
  - Znajdź zbiór S1 punktów znajdujących się **po prawej stronie prostej AC** oraz zbiór S2 punktów **po prawej stronie prostej BC**.
  - Wywołaj rekurencyjnie **QuickHull(A, C, S1)** oraz **QuickHull(B, C, S2)**.



Odległość punktu od prostej



Odległość punktu P=(x0;y0) od prostej o równaniu Ax+By+C=0 jest dana wzorem:

$$d = \frac{|Ax_0 + By_0 + C|}{\sqrt{A^2 + B^2}}$$

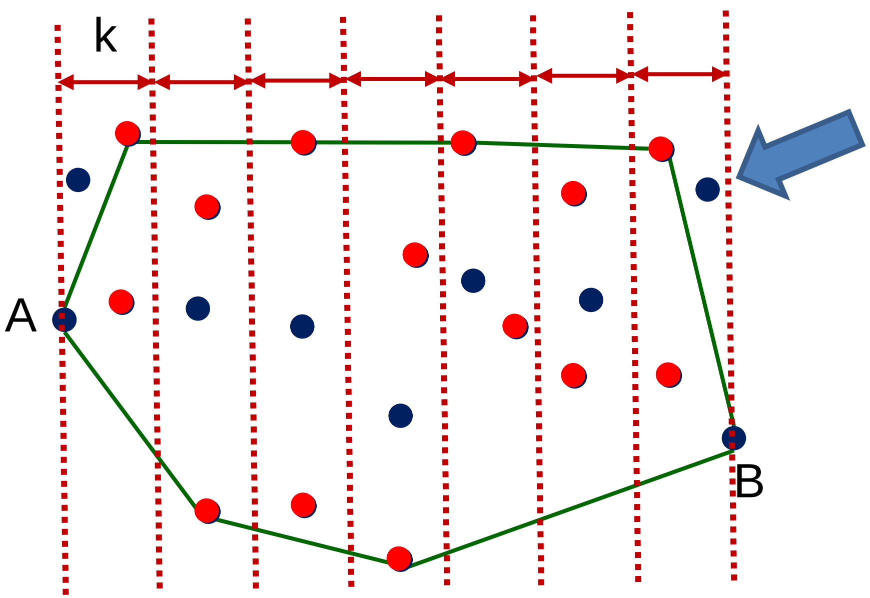
http://www.naukowiec.org/wiedza/matematyka/odleglosc-punktu-od-prostej_1930.html

Rozwiązania przybliżone

Istnieje wiele algorytmów określających przybliżoną otoczkę wypukłą. Są one **szybkie**, **proste** jednak występują **błędy rozwiązania**!

Krok 1:

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



Krok 2:

Podziel obszar pomiędzy A i B na równe podobszary o danej szerokości k

Krok 3:

Znajdź w każdym pasie **punkty skrajne** względem współrzędnej y

Krok 4:

Znajdź otoczkę wypukłą dla tak wyselekcjonowanych punktów.