TRIANGULACJA WIELOKĄTA PROSTEGO

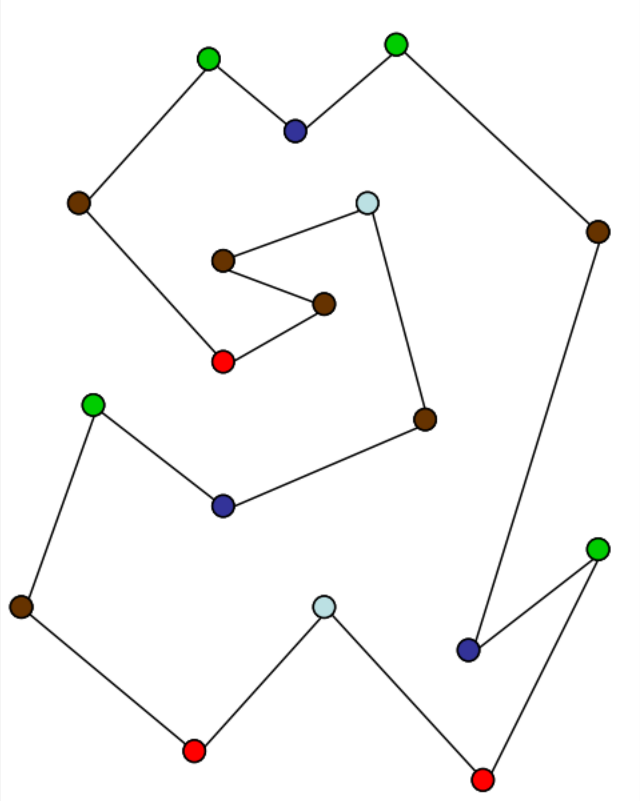
Bartłomiej Tempka

Radosław Kawa

ALGORYTM PODZIAŁU NA WIELOKĄTY MONOTONICZNE

RODZAJE WIERZCHOŁKÓW

początkowy, gdy obaj jego sąsiedzi leżą poniżej i kąt wewnętrzny < π
końcowy, gdy obaj jego sąsiedzi leżą powyżej i kąt wewnętrzny < π
łączący, gdy obaj jego sąsiedzi leżą powyżej i kąt wewnętrzny > π
dzielący, gdy obaj jego sąsiedzi leżą poniżej i kąt wewnętrzny > π
prawidłowy, w pozostałych przypadkach (ma jednego sąsiada powyżej, drugiego – poniżej).



ALGORYTM ZAMIATANIA

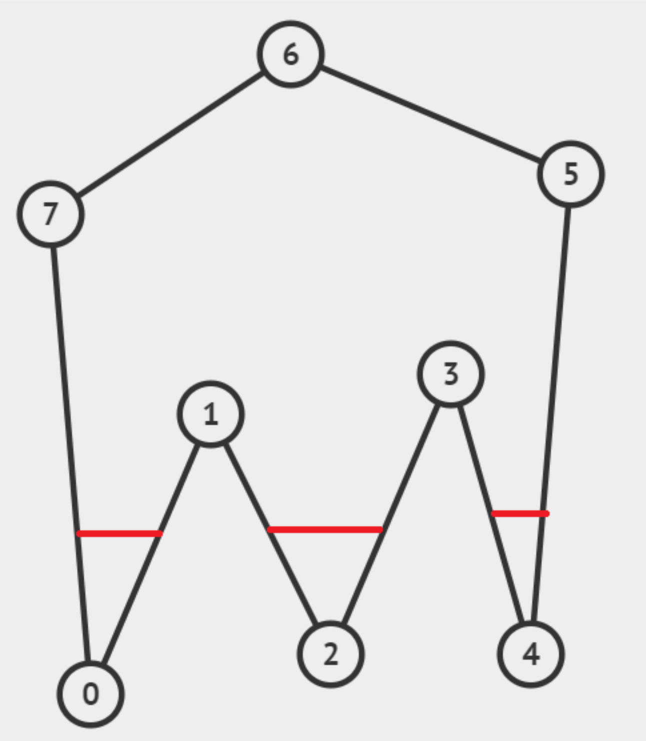
- Algorytm ma za zadanie przez dodawanie przekątnych wyeliminować wierzchołki dzielące i łączące
- Stan miotły- przedziały zawierające się wewnątrz wielokąta posortowane od lewej do prawej
- Pomocnik przedziału- najniższy wierzchołek powyżej miotły taki, że odcinek łączący ten wierzchołek z krawędzią ograniczającą przedział z lewej strony leży wewnątrz P

ZASTOSOWANE STRUKTURY

- Struktura zdarzeń- lista L uporządkowanych malejąco względem współrzędnej y wierzchołków P
- Struktura stanu- Drzewo Wyszukiwania Binarnego przechowujące aktualne przedziały zawierające się środku P

PRZYKŁADOWY STAN DRZEWA

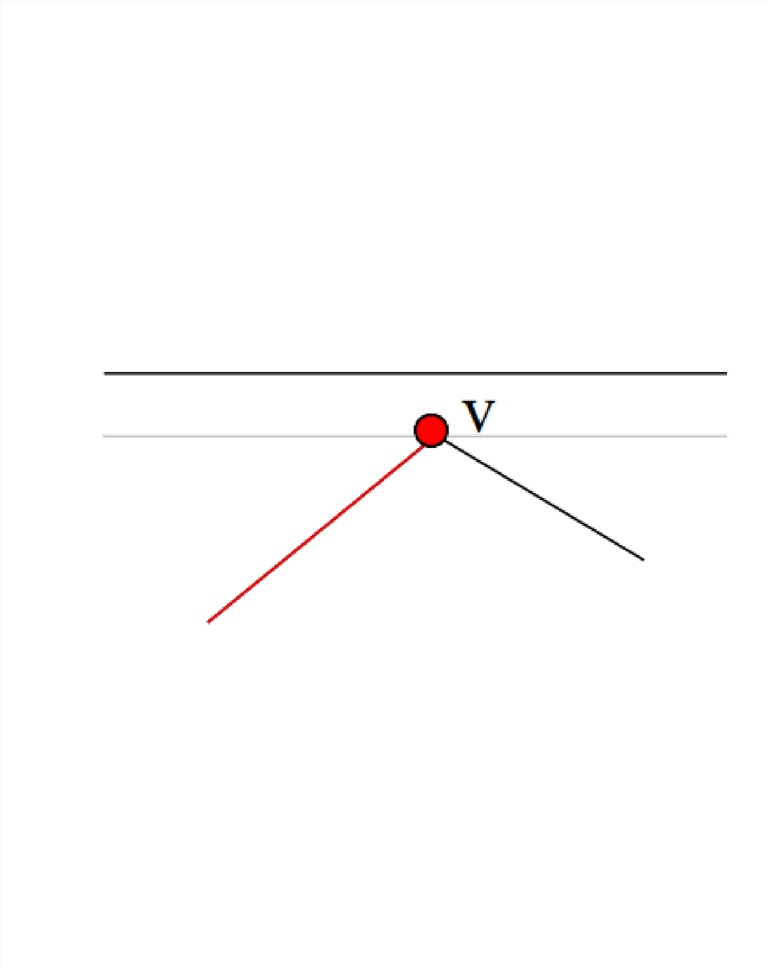
- Root = (1,2) – (2,3)
- Root.left = (7,0) – (1,0)
- Root.right =(3,4) – (4,5)



WIERZCHOŁEK POCZĄTKOWY

Wstaw przedział który się zaczyna do drzewa

Ustaw pomocnika drzewa na v

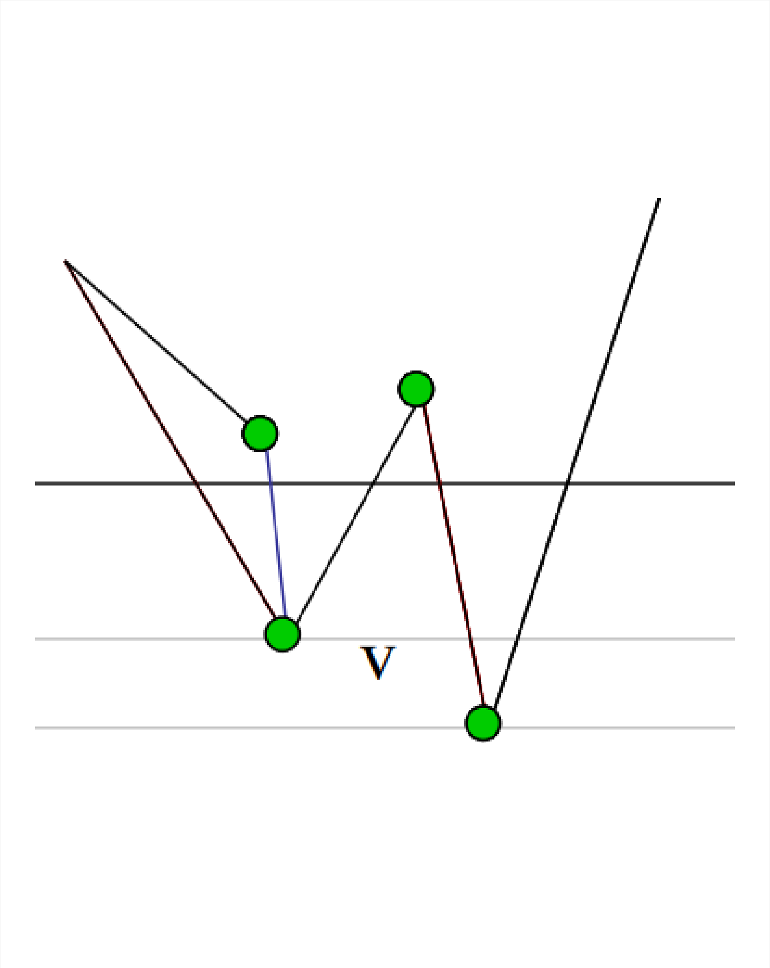


WIERZCHOŁEK KOŃCOWY

If pomocnik przedziału który się kończy jest wierzchołkiem łączący

Then wstaw przekątną między v i pomocnika

Usuń przedział z drzewa



WIERZCHOŁEK DZIELĄCY

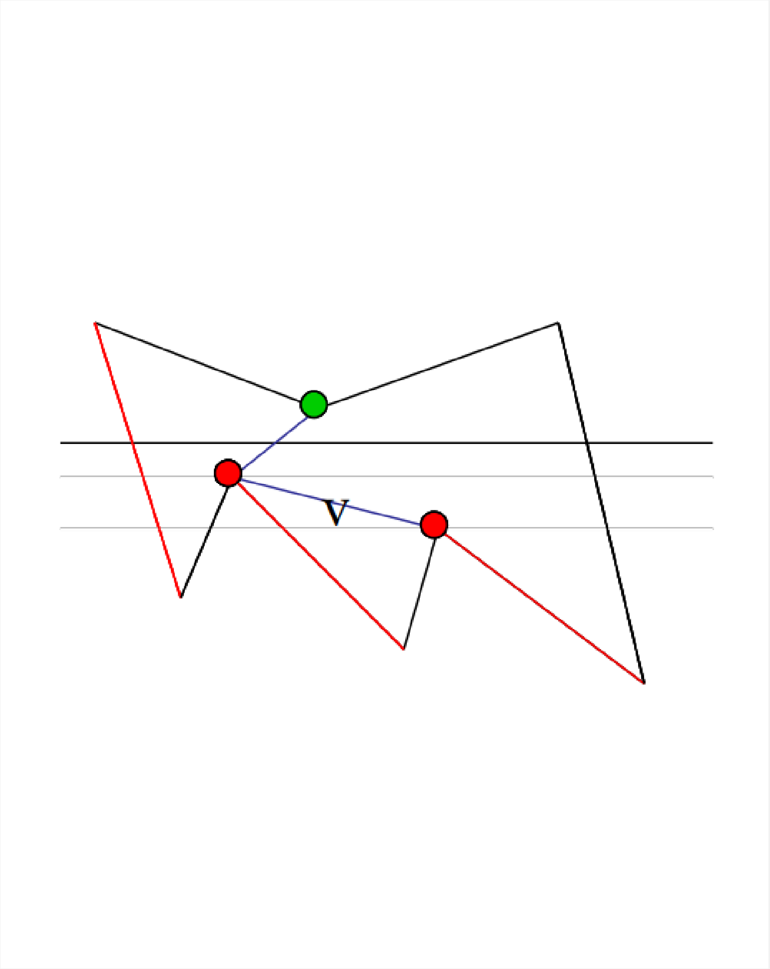
Znajdź w drzewie przedział który wierzchołek dzieli

Wstaw przekątną między wierzchołek v a pomocnik tego przedziału

Zmień bieżący przedział tak aby reprezentował przedział lewy

Wstaw przedział prawy do drzewa

Ustaw ich pomocników na v



WIERZCHOŁEK ŁĄCZĄCY

If pomocnik prawego przedziału jest wierzchołkiem łączącym

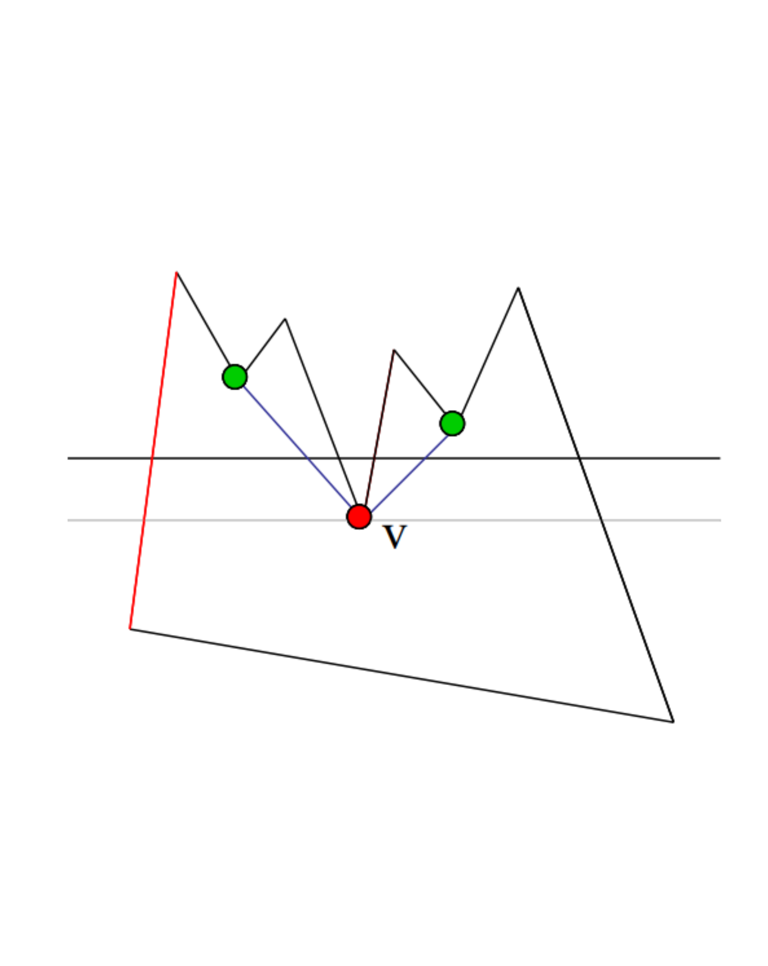
Then wstaw przekątną miedzy v i pomocnika

If pomocnik prawego lewego jest wierzchołkiem łączącym

Then wstaw przekątną miedzy v i pomocnika

Stwórz w drzewie jeden przedział z ich obu (usuń jeden zaktualizuj drugi)

Ustaw pomocnika na v



WIERZCHOŁEK PRAWIDŁOWY

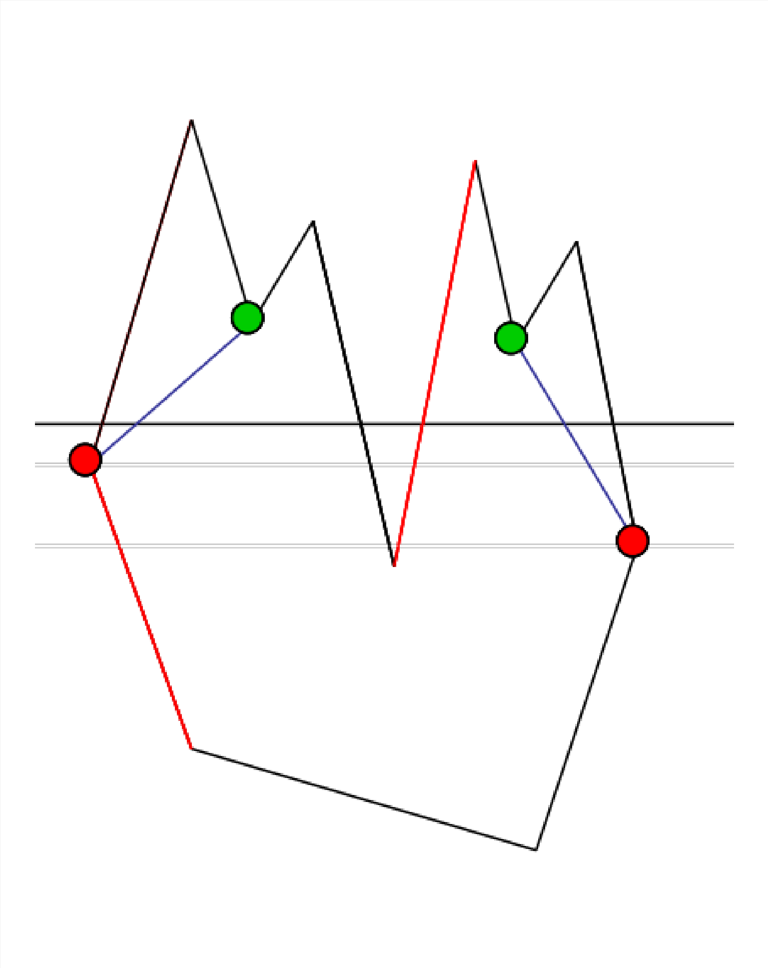
Znajdz przedział powyżej którego v jest końcem

If pomocnik przedziału jest wierzchołkiem łączącym

Then wstaw przekątną miedzy v i pomocnika

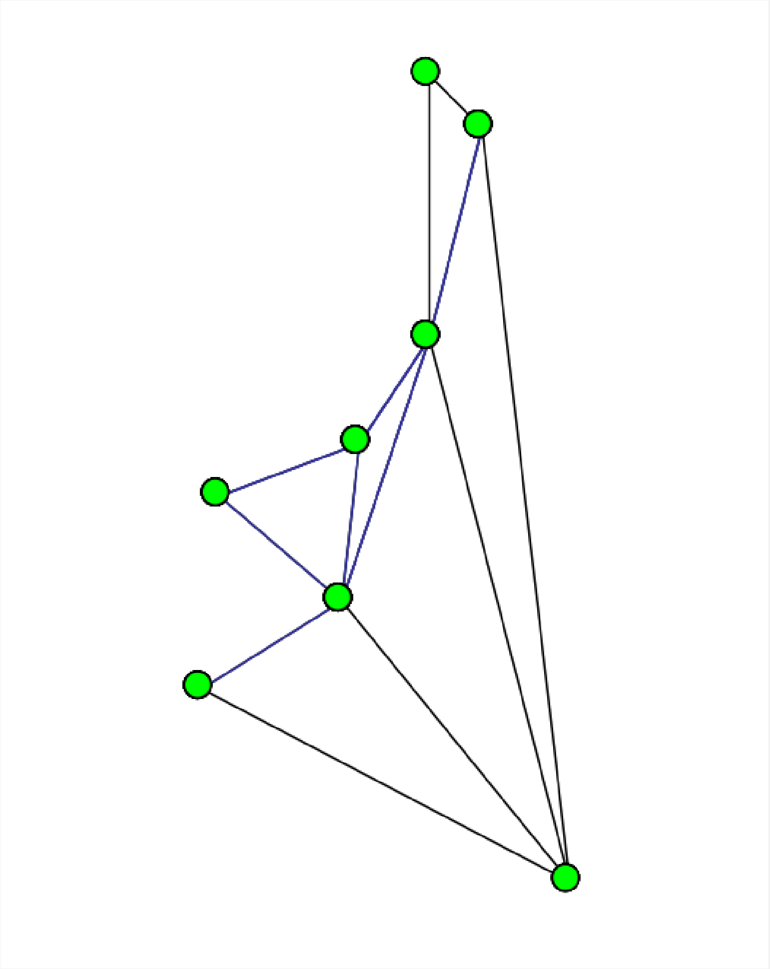
Ustaw pomocnika na v

Zaktualizuj przedział



TRIANGULOWANIE WIELOKĄTA MONOTONICZNEGO

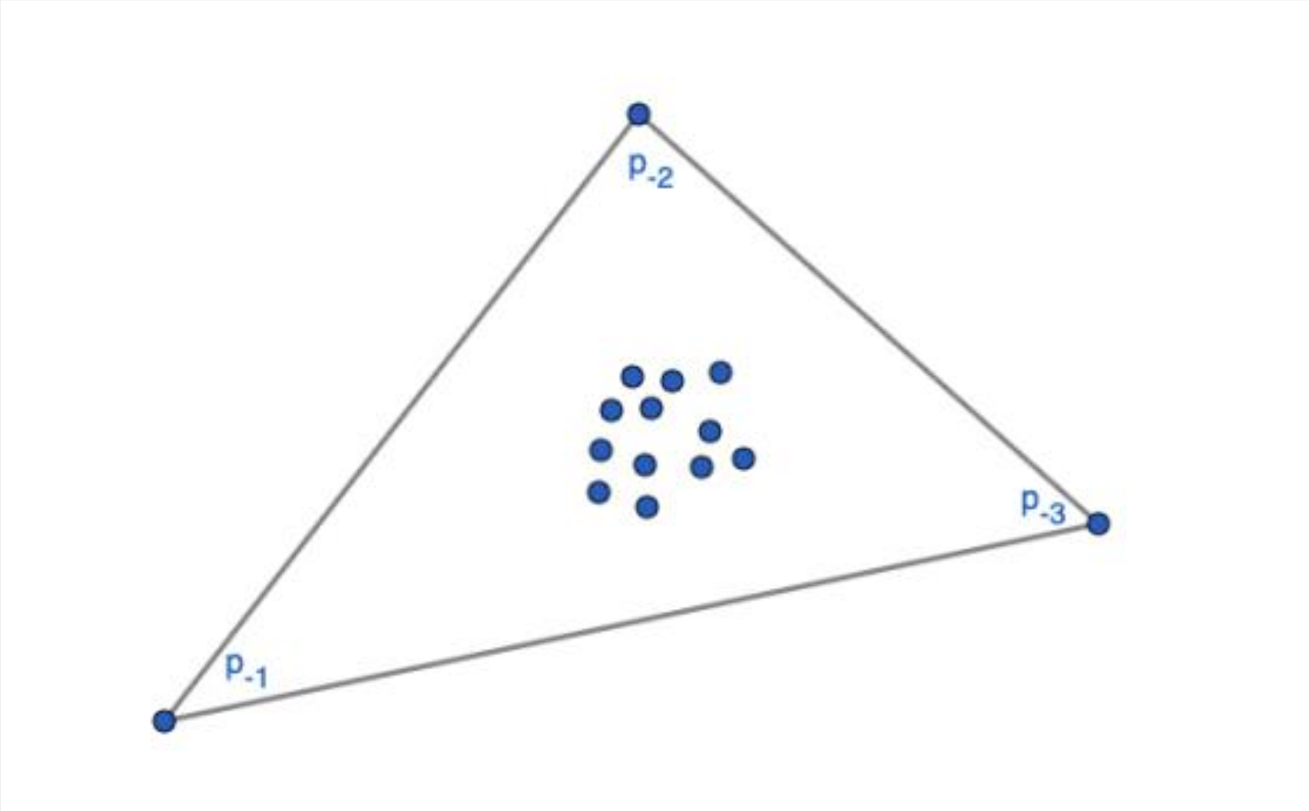
- Określamy lewy i prawy łańcuch wielokąta względem kierunku monotoniczności
- Porządkujemy wierzchołki wzdłuż kierunku monotoniczności
- Wkładamy dwa pierwsze wierzchołki na stos.
- Jeśli kolejny wierzchołek należy do innego łańcucha niż wierzchołek stanowiący szczyt stosu, to możemy go połączyć ze wszystkimi wierzchołkami na stosie. Na stosie zostają dwa wierzchołki, które były „zamiatane" ostatnie.
- Jeśli kolejny wierzchołek należy do tego samego łańcucha co wierzchołek ze szczytu stosu, to analizujemy kolejne trójkąty, jakie tworzy dany wierzchołek z wierzchołkami zdejmowanymi ze stosu.
  - Jeśli trójkąt należy do wielokąta, to usuwamy wierzchołek ze szczytu stosu
  - w przeciwnym przypadku umieszczamy badane wierzchołki na stosie.

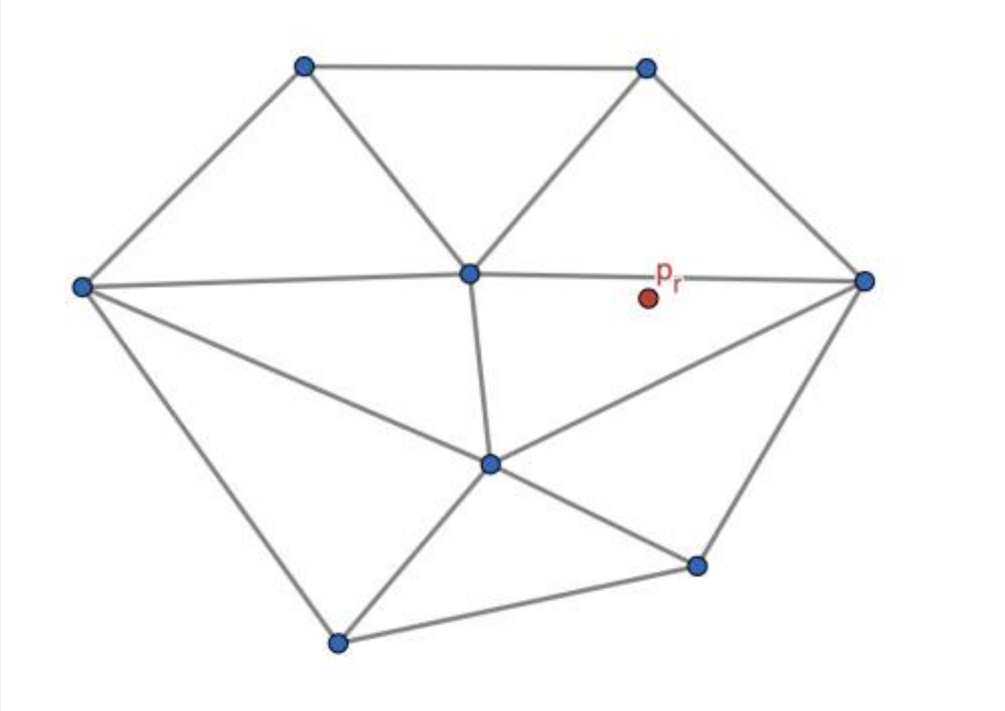


ALGORYTM DELAUNAYA

SKONSTRUOWANIE SUPERTRÓJKĄTA

- W pierwszym kroku algorytm tworzy supertrójkąt, który będzie zawierał wszystkie zadane punkty



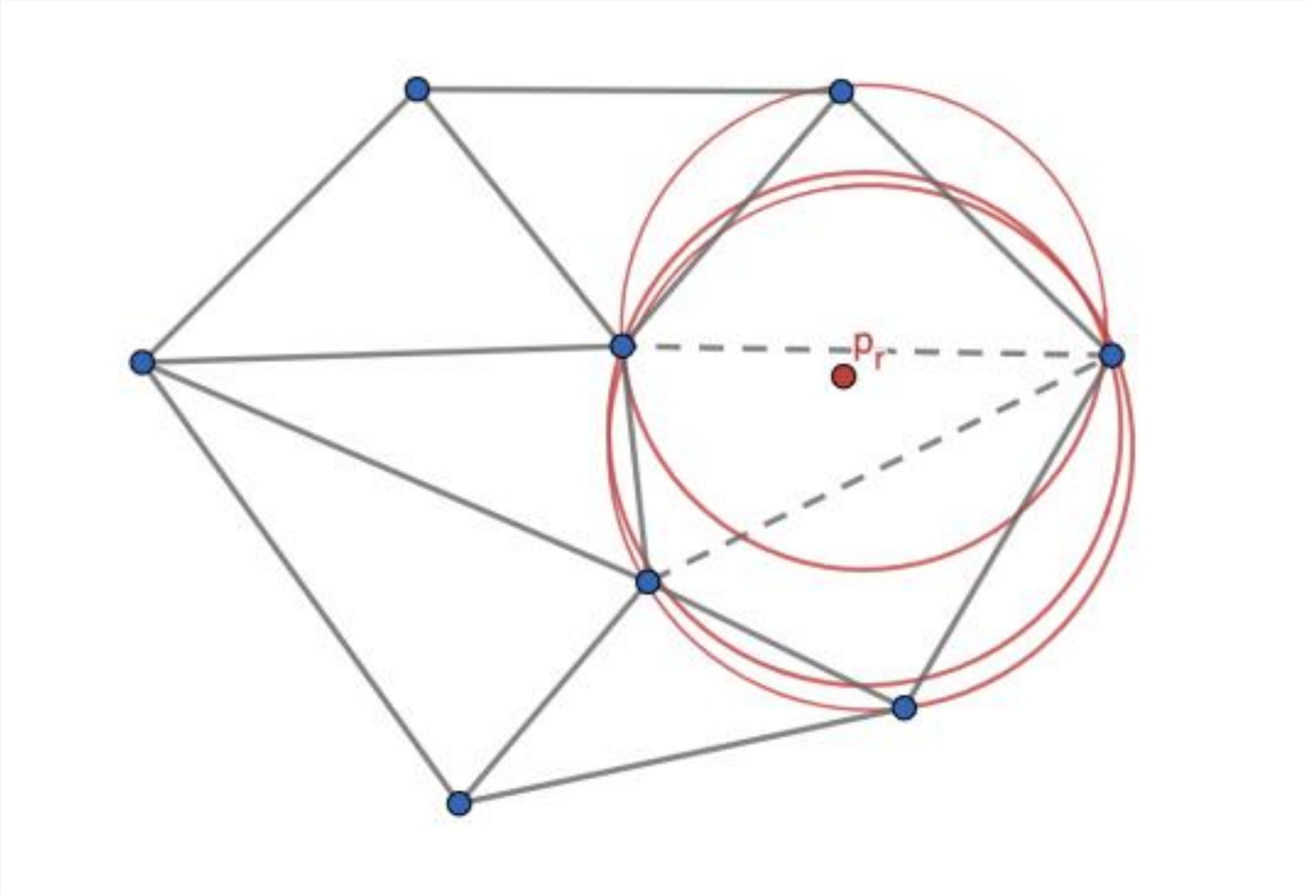


PODEJŚCIE ITERACYJNE

- Do algorytmu Delaunaya posłużyłem się metodą iteracyjną. Bowyera-Watsona, aby uzyskać lepszą złożoność
- Polega ono na retriangulacji otoczenia dodawanego punktu, czyli usuwaniu niepasujących trójkątów i tworzeniu nowych trójkątów przez dodawanie krawędzi z jednym końcem w dodanym punkcie. Najczęsciej algorytm będzie miał złożoność O(nlogn), bardzo rzadko O(n^2)

ZNAJDOWANIE TRÓJKĄTÓW DO USUNIĘCIA

- Z każdym kolejnym krokiem algorytmu sprawdzam czy dany punkt znajduje się w trójkątach trinagulacji, gdy tak się dzieje są one wrzucane na stos i przechowywane, aż do momentu usunięcia



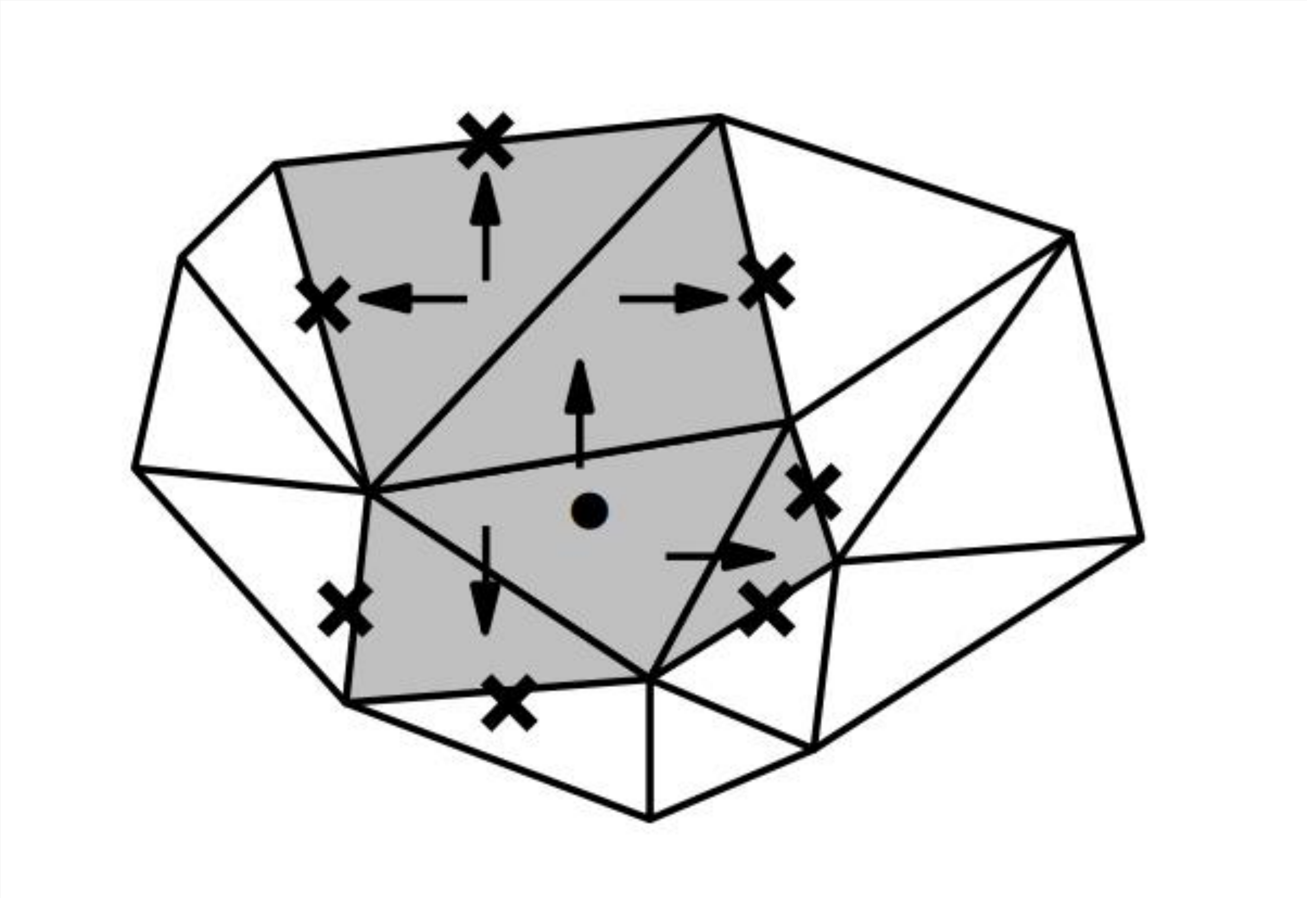
$$\begin{vmatrix} A_x - D_x & A_y - D_y & (A_x^2 - D_x^2) + (A_y^2 - D_y^2) \\ B_x - D_x & B_y - D_y & (B_x^2 - D_x^2) + (B_y^2 - D_y^2) \\ C_x - D_x & C_y - D_y & (C_x^2 - D_x^2) + (C_y^2 - D_y^2) \end{vmatrix} > 0$$

ZAWIERANIE SIĘ PUNKTU W OKRĘGU OPISANYM NA TRÓJKĄCIE

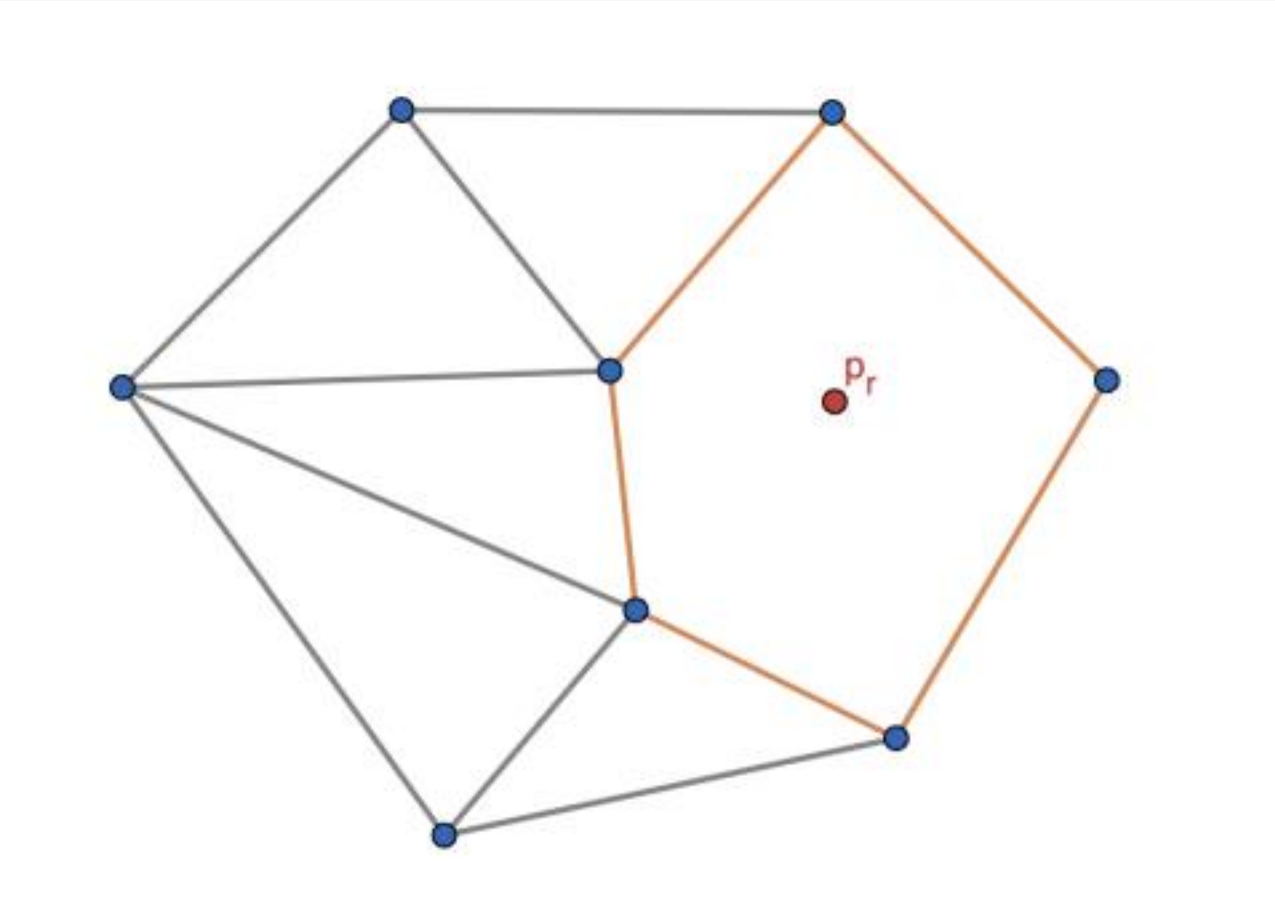
- Jest to sprawdzane za pomocą wyznaczników i orientacji względem jakiej są zadane dane trójkąty, gdy wyznacznik jest większy od 0, oznacza to, że dany punkt znalazł się w okręgu opisanym na tym trójkącie

POSZUKIWANIE SĄSIADÓW TRÓJKĄTÓW

- Polega to na zwykłym przeszukiwaniu trójkątów, które, dzielą ze sobą daną krawędź, gdy tak się dzieje to je pomijamy, dlatego interesują nas tylko krawędzie, które nie będą w żaden sposób dzielone, gdy znajdziemy wrzucane są na jakiś stos
- Te krawędzie będą potrzebne do utworzenia nowych trójkątów triangulacji

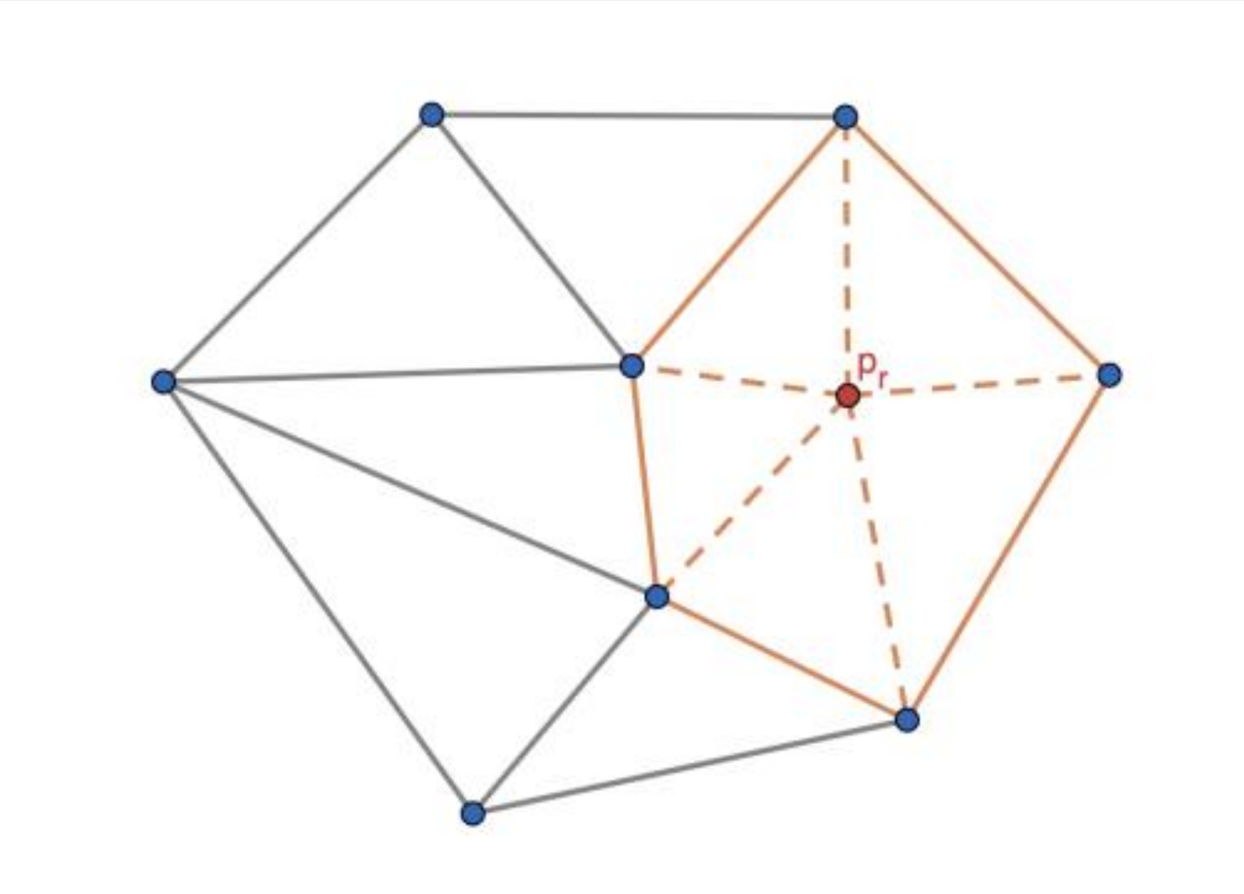


USUNIĘCIE TRÓJKĄTÓW



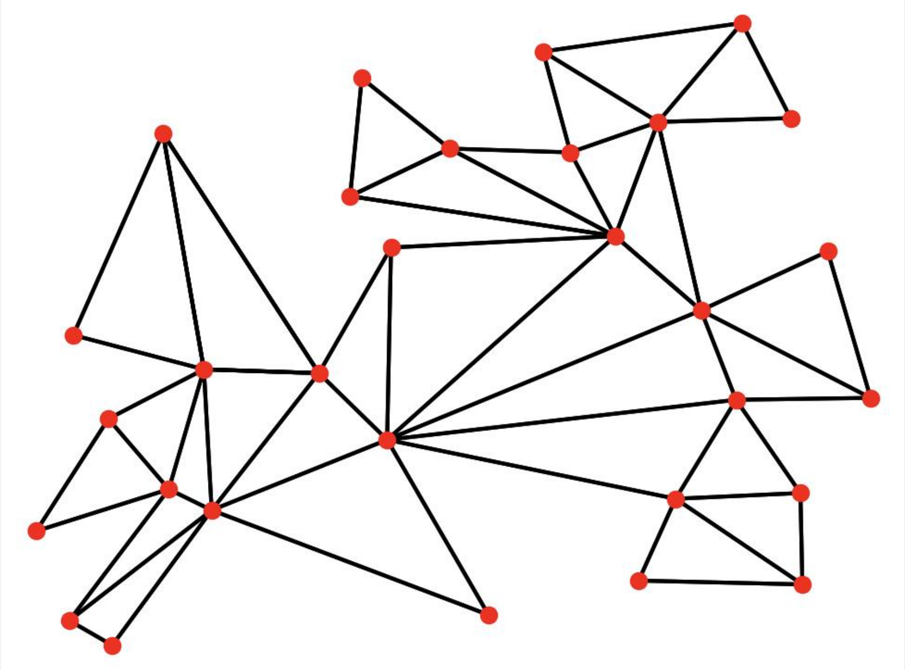
DODANIE NOWYCH TRÓJKĄTÓW

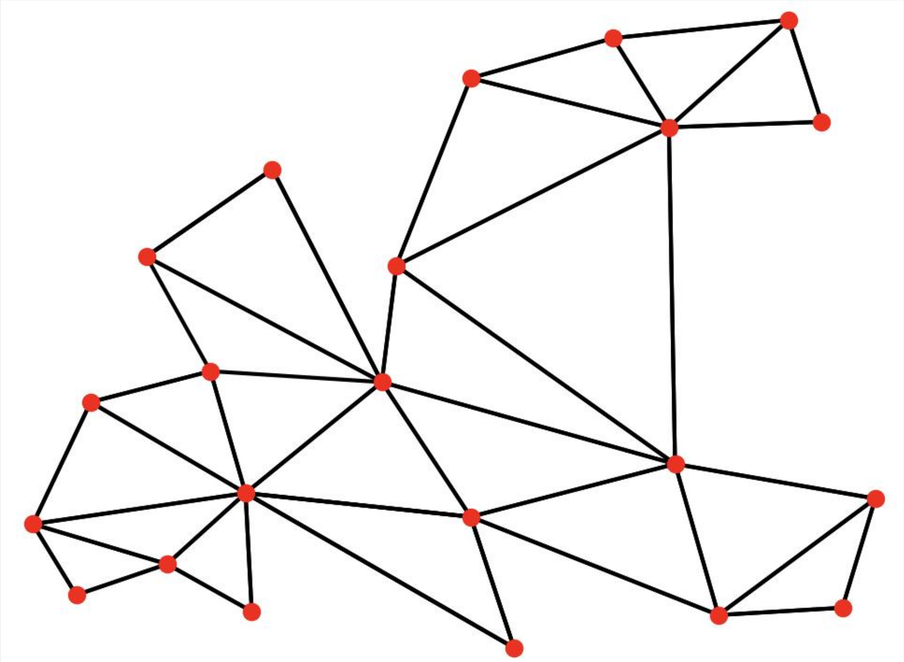
Nowe trójkąty są utworzone z zapisanych wcześniej krawędzi i aktualnie przeglądanego punktu.



USUNIĘCIE SUPER-TRÓJKĄTA ORAZ ZEWNĘTRZNYCH TRÓJKĄTÓW

- Po znalezieniu triangulacji, należy usunąć wszystkie trójkąty, które dzielą wierzchołki z supertrójkątem. Zwyczajnie sprawdza czy dany wierzchołek znajduje się w wierzchołkach badanego trójkąta
- Do usuwania zewnętrznych trójkątów wykorzystałem orientacje w jaki sposób te trójkąty zostały zadane (względem przeciwnym do wskazówek zegara) , gdy jest inaczej to je zwyczajnie usuwam
- Dodatkowo sprawdzam przecięcia trójkątów z krawędziami wielokąta, gdy tak się dzieje to je usuwam, na ich miejsce daje poprawione wersje trójkątów, z zmienionymi współrzędnymi, oczywiście gdy mogę je dodać, czyli gdy nie kolidują z innymi trójkątami





PRZYKŁADOWE WYNIKI ALGORYTMU