



# ALGORYTM KIRKPATRICK'A

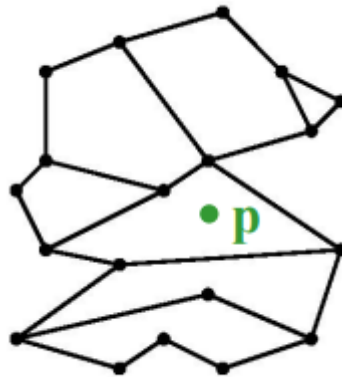
---

Mateusz Bartnicki

Jan Kalęba

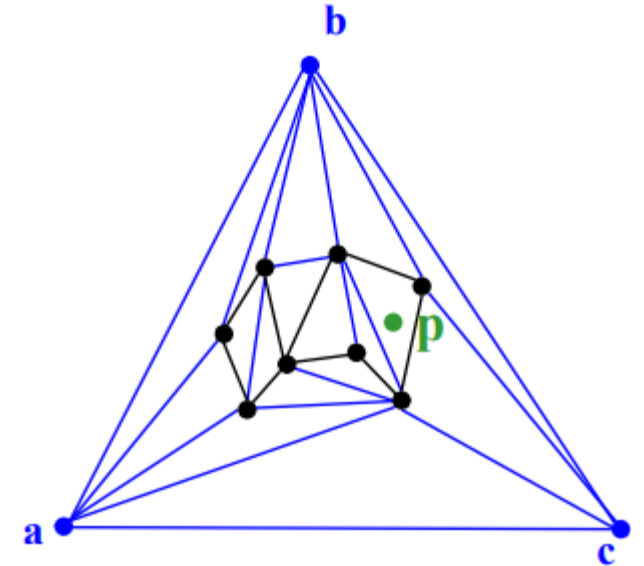
# DO CZEGO SŁUŻY ALGORYTM KIRKPATRICK'A?

Zadany jest obszar z podziałem poligonowym oraz współrzędne punktu  $p$ . Zadaniem algorytmu jest zlokalizowanie, w którym elemencie znajduje się szukany punkt



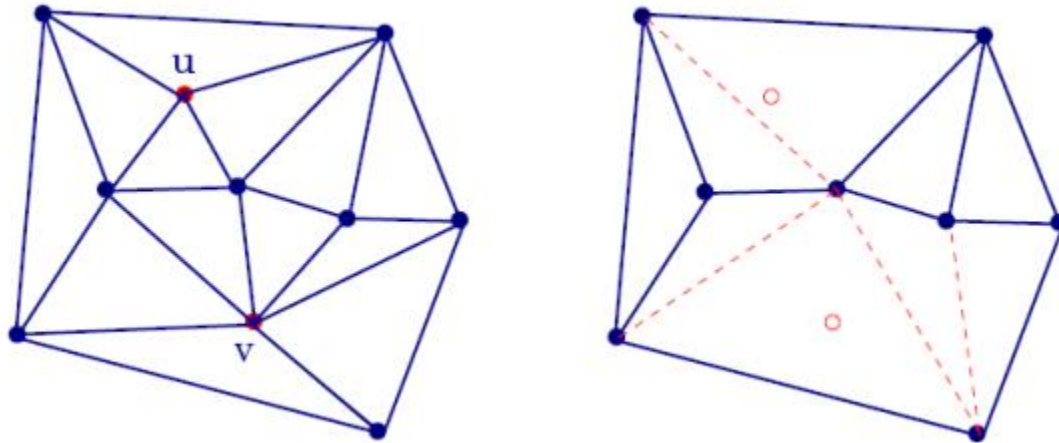
# PRZEBIEG ALGORYTMU – PRZYGOTOWANIE

- Sprawdzenie poprawności podanego podziału poligonowego. Jeśli nie jest on triangulacją, trzeba ją wykonać.
- Sprawdzenie otoczki wypukłej punktów. Jeśli nie tworzą one trójkąta, trzeba go utworzyć oraz połączyć z figurą. (wierzchołki a, b i c na rysunku zostały dodatkowo utworzone i połączone z otoczką czarnej figury).

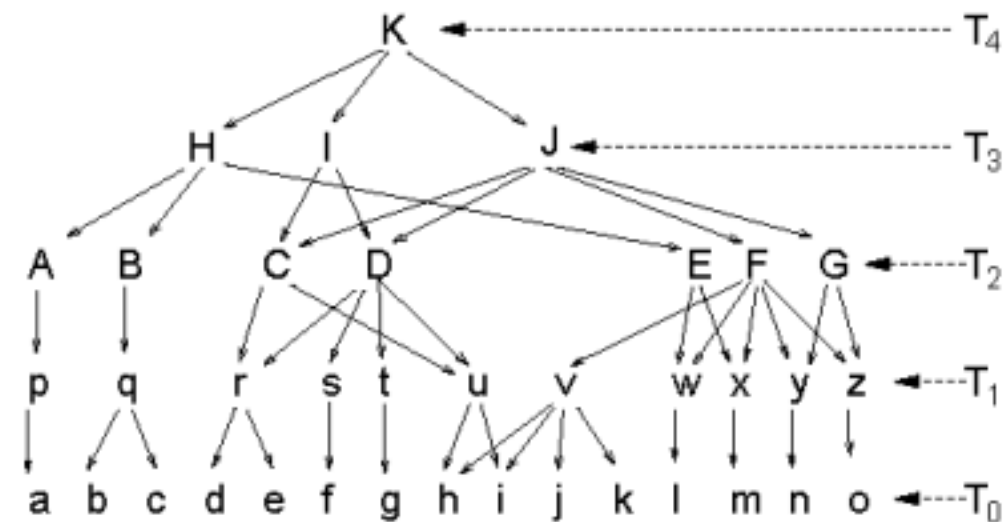
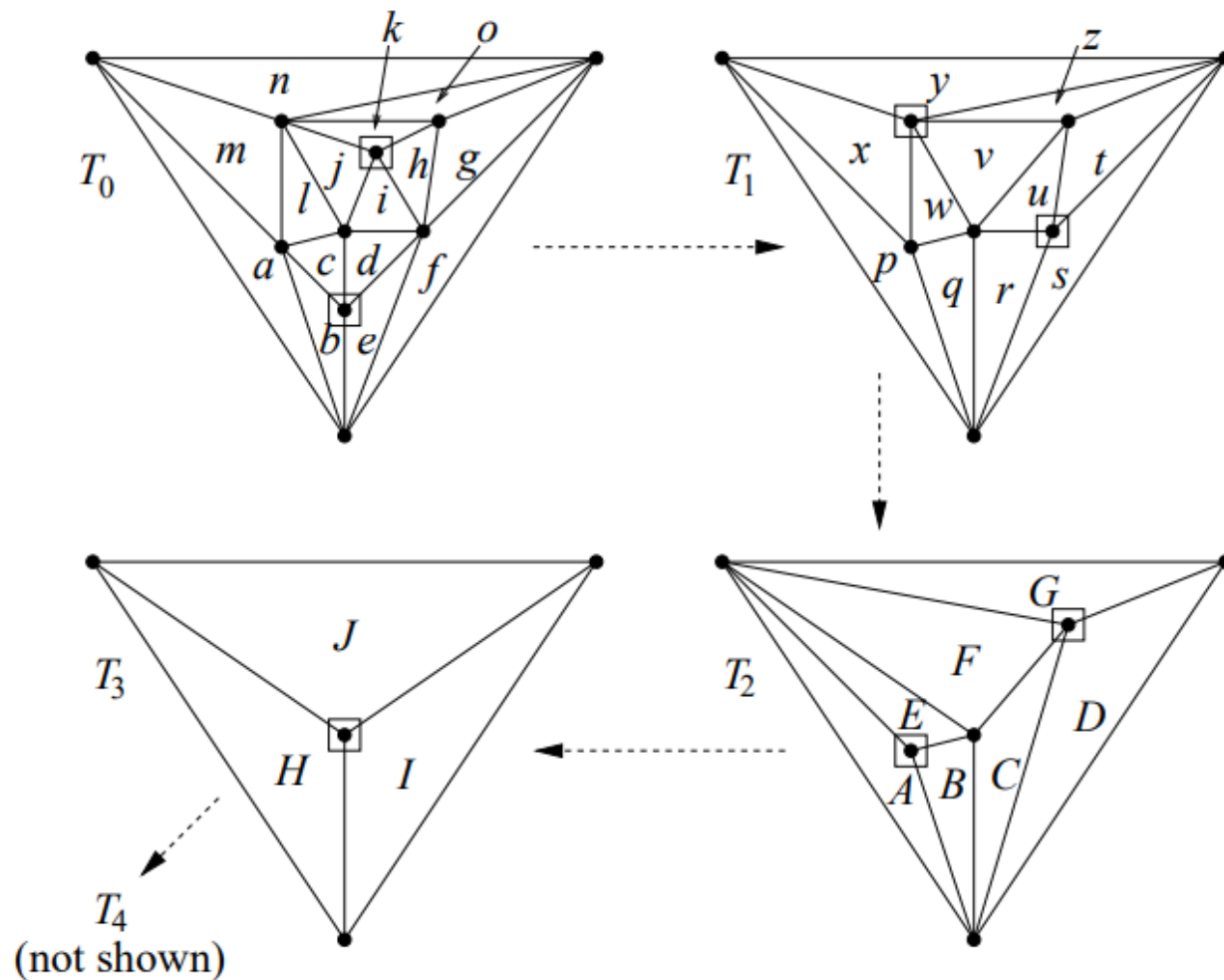


# PRZEBIEG ALGORYTMU

W kolejnych krokach będziemy poszukiwać zbioru niezależnych wierzchołków wewnętrznych, które następnie będą usuwane wraz z krawędziami. Po tym kroku, figura wymaga ponownej triangulacji. Będziemy to czynić, aż zostanie nam trójkąt bez punktów w środku.



# UZYSKANIE ODPOWIEDZI



# PSEUDOKOD

**Kirkpatrick(T, point):**

T – zbiór trójkątów

jeśli otoczka wypukła figury nie jest trójkątem:

utwórz trójkąt obejmujący wszystkie punkty

połącz trójkąt z otoczką wypukłą

utwórz drzewo poprzez preprocessing

**Preprocess(T, S):**

S – zbiór wszystkich punktów

dopóki  $|T| > 1$ :

znajdź zbiór niezależnych wewnętrznych wierzchołków

usuń trójkąty zawierające te wierzchołki

usuń wierzchołki z S wraz z krawędziami

ponownie trianguluj S i dodaj nowe trójkąty do T

do drzewa dodaj nowe trójkąty jako dzieci obszaru, z którego powstały

odszukaj trójkąt zawierający punkt w drzewie

# ANALIZA ZŁOŻONOŚCI

Preprocessing:

$n$  – liczba wierzchołków

Przetwarzanie wstępne polega na budowaniu hierarchii trójkątów. Hierarchia ma  $(\log n)$  poziomów. Na każdym poziomie konstrukcja tych trójkątów może być wykonana w czasie  $O(n)$ . Ogólna złożoność przetwarzania wstępnego wynosi  $O(n * \log n)$ .

Odszukiwanie wyniku:

Na każdym poziomie sprawdzenie, czy punkt zapytania znajduje się w trójkącie, może być wykonane w czasie  $O(1)$ , a poziomów jest  $(\log n)$ . Odszukiwanie wyniku jest zatem w czasie  $O(\log n)$ .

Całkowita złożoność algorytmu wynosi  $O(n * \log n)$ .

# BIBLIOGRAFIA

<https://sites.cs.ucsb.edu/~suri/cs235/Location.pdf>

<https://ics.uci.edu/~goodrich/teach/geom/notes/Kirkpatrick.pdf>

Prezentacja z wykładu