Przetwarzanie i przechowywanie opisu siatki trójkątnej na płaszczyźnie

Paweł Surdyka

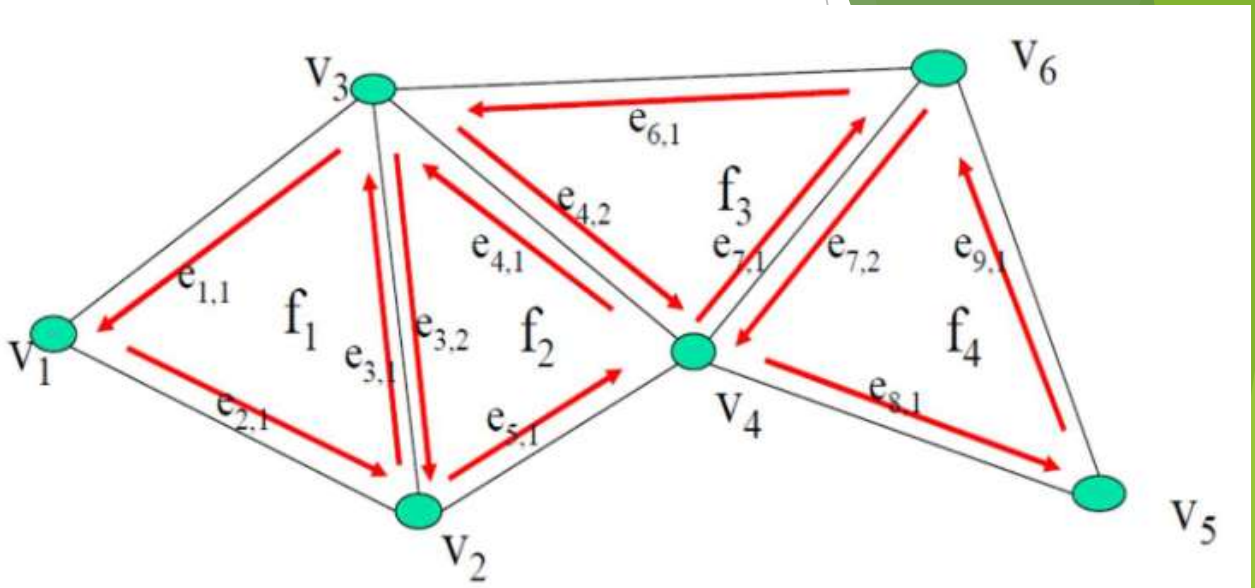
Paweł Derbisz

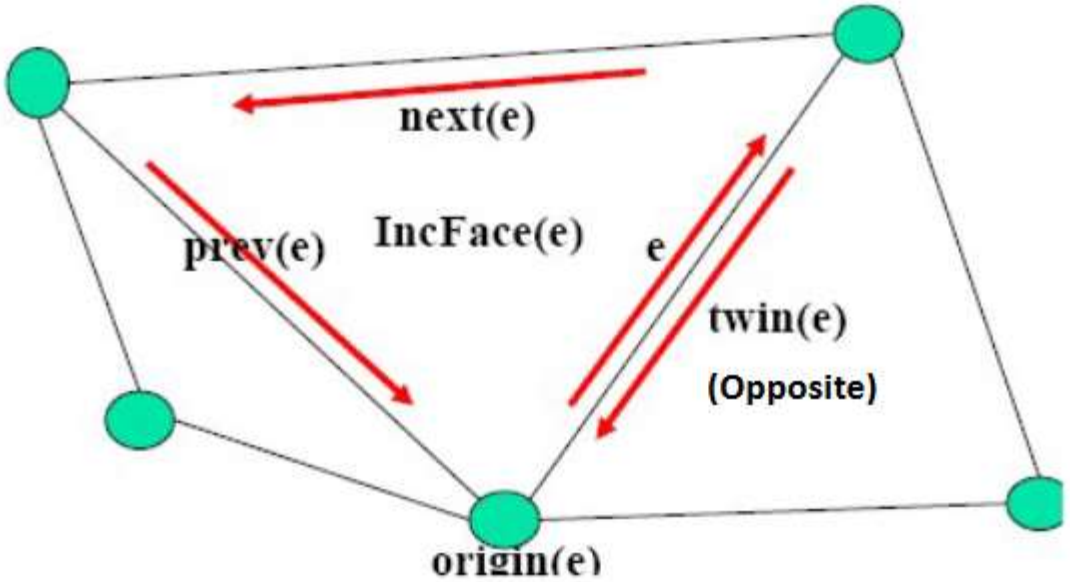
Opis struktury Half-Edge

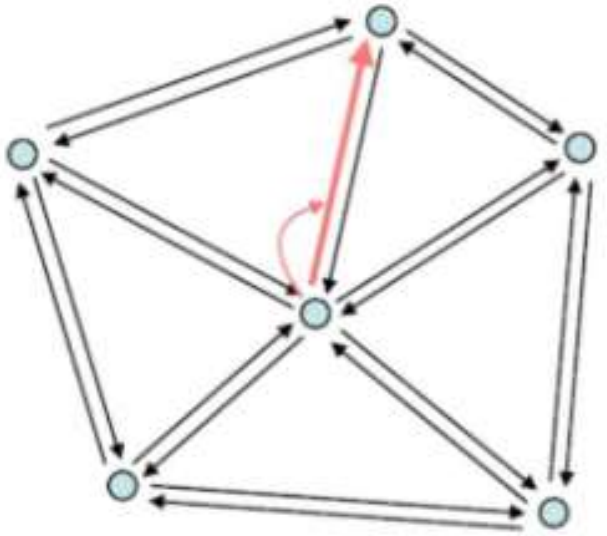
A dokładniej Doubly connected edge list (DCEL) jest to struktura danych, która jest używana do reprezentacji siatki w postaci half-edge. Struktura ta składa się z trzech głównych elementów:

1. Krawędzie (edges): Każda krawędź jest reprezentowana przez pojedynczy obiekt, który zawiera informacje o punkcie początkowym, trójkącie, który ją zawiera oraz o kolejnej, poprzedniej i odwrotnej krawędzi.
2. Wierzchołki (vertices): Każdy wierzchołek jest reprezentowany przez pojedynczy obiekt, który zawiera informacje o położeniu wierzchołka oraz o half-edge, który jest przypisany do danego wierzchołka.
3. Trójkąty (faces): Każdy trójkąt jest reprezentowany przez pojedynczy obiekt, który zawiera informacje o half-edge, który jest skojarzony z danym trójkątem.

Przykładowy wygląd







Wykorzystanie reprezentacji DCEL

Reprezentacja DCEL jest często używana w algorytmach graficznych i geometrii komputerowej. Oto kilka przykładów zastosowań:

1. Triangulacja Delaunay'a: Algorytm ten jest używany do generowania triangulacji punktów na płaszczyźnie. Reprezentacja DCEL jest używana do przechowywania i przetwarzania informacji o krawędziach i wierzchołkach w celu zbudowania triangulacji.
2. Operacje Boolean: Algorytm ten jest używany do łączenia lub dzielenia dwóch lub więcej obiektów. Reprezentacja DCEL jest używana do przechowywania i przetwarzania informacji o krawędziach i wierzchołkach, co umożliwia wykonywanie operacji Boolean na tych obiektach.
3. Detekcja kolizji: Reprezentacja DCEL jest używana do przechowywania i przetwarzania informacji o krawędziach i wierzchołkach w celu określenia, czy dwie figury przestrzenne nachodzą na siebie.
4. Przetwarzanie obrazów: Algorytmy wykorzystujące reprezentację DCEL mogą być używane do segmentacji obrazów, rozpoznawania kształtów czy też analizy obrazów 3D.
5. Analiza topologiczna: DCEL jest używane do reprezentowania topologii sieci danych, gdzie pojedyncze krawędzie reprezentują połączenie między dwoma wierzchołkami.

Zalety reprezentacji half-Edge nad reprezentacją tablicową

- Łatwiejszy dostęp do informacji o krawędziach: W reprezentacji half-edge, w każdej krawędzi jest informacja o sąsiadujących krawędziach, co umożliwia łatwiejszy do nich dostęp w porównaniu z reprezentacją tablicową, gdzie informacje o krawędziach są rozproszone po tablicy trójkątów.
- Łatwiejszy dostęp do informacji o wierzchołkach: W reprezentacji half-edge, można łatwo uzyskać informacje o incydentnych wierzchołkach wykorzystując informacje o początkach sąsiadujących krawędzi
- Łatwiejsze operacje na siatce: Reprezentacja half-edge umożliwia łatwiejsze wykonywanie operacji na siatce, takich jak przeszukiwanie sąsiednich krawędzi i trójkątów, przekształcanie siatki, czy też modyfikowanie jej.

Porównanie operacji z użyciem dwóch reprezentacji pod kątem kosztu pamięciowego i czasowego

- Reprezentacja składająca się z listy punktów i listy trójkątów (każdy trójkąt opisany przez numery trzech wierzchołków, punkty numerowane są od zera)
- Reprezentacja składająca się ze struktur danych:

  -Half-Edge

  -Vertex

  -Face

OPERACJA 1: wyznaczanie otoczenia dla wybranego wierzchołka (kolejne warstwy incydentnych wierzchołków – należy rozpatrzyć otoczenia składające się z jednej warstwy oraz dwóch warstw)

Reprezentacja pierwsza

Algorytm polega na:

- przeglądanie tablicy z trójkątami w celu znalezienia trójkątów zawierających dany punkt i utworzeniu pierwszej warstwy wierzchołków

- powtórzenie podpunktu powyżej dla każdego punktu z pierwszej warstwy

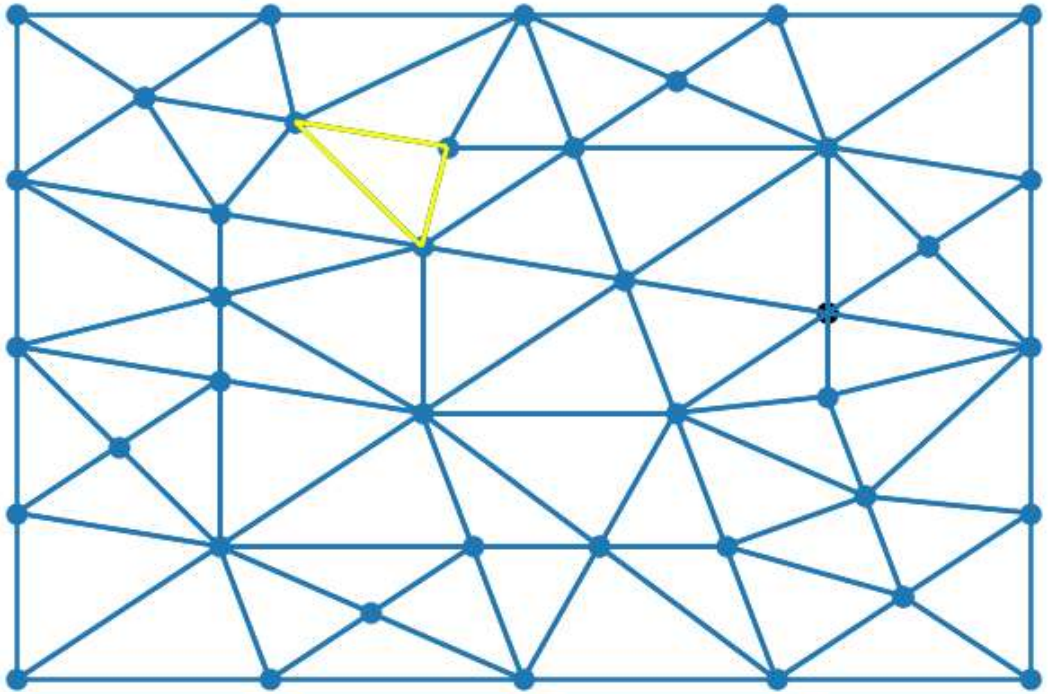
w celu utworzenia drugiej warstwy

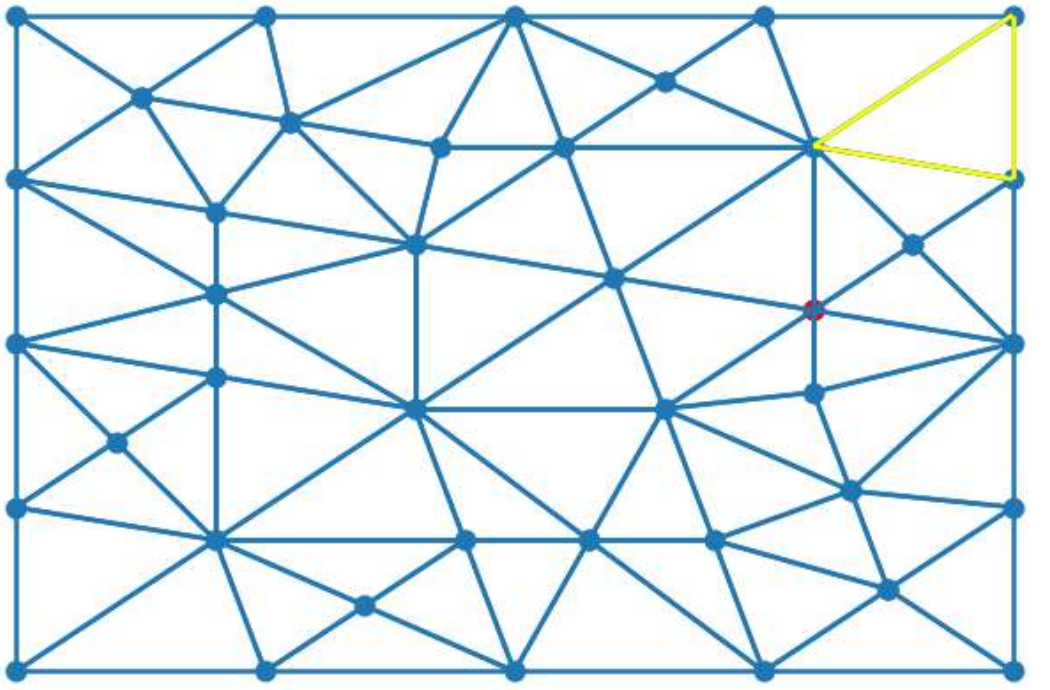
-usunięcie postarzających się punktów

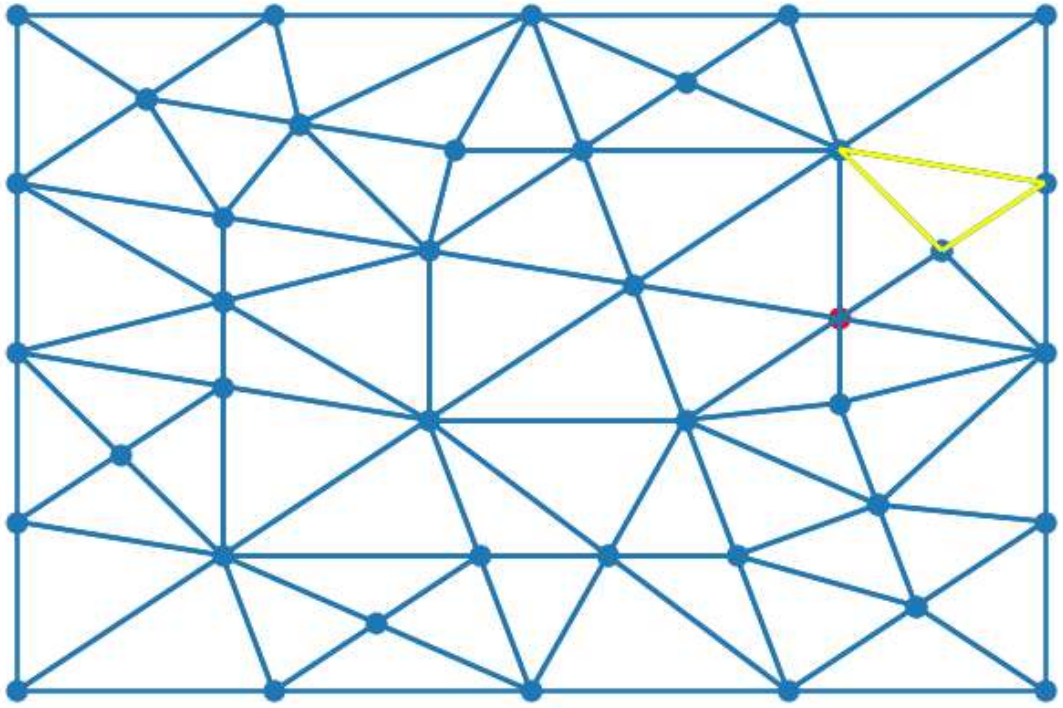
Złożoność czasowa : O(n*d)

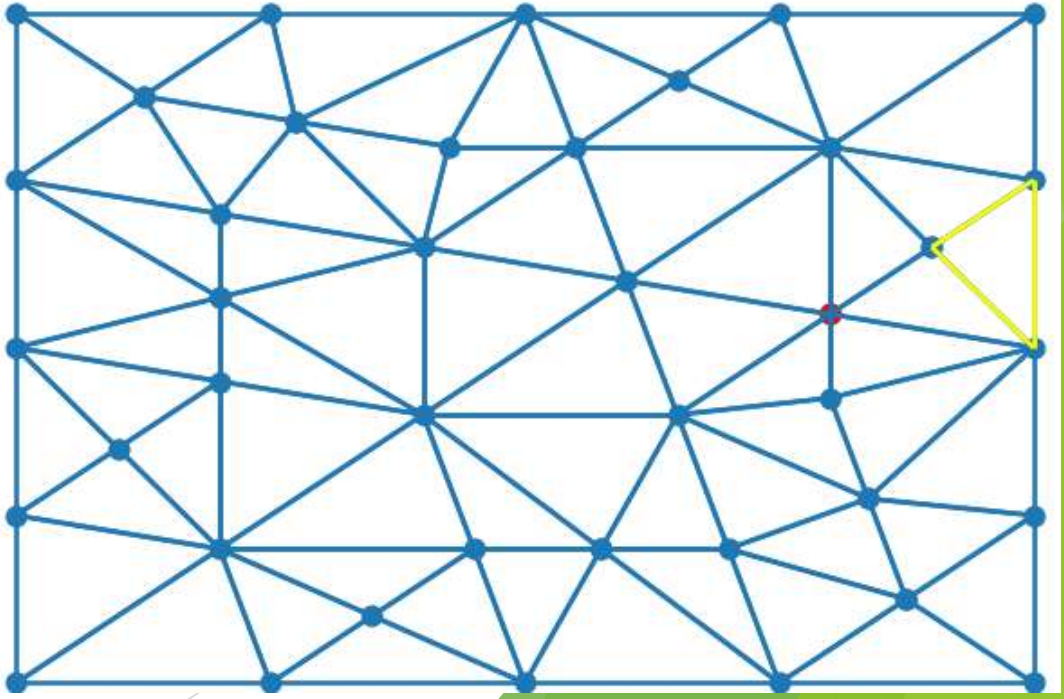
Złożoność pamięciowa : O(d^2) gdzie

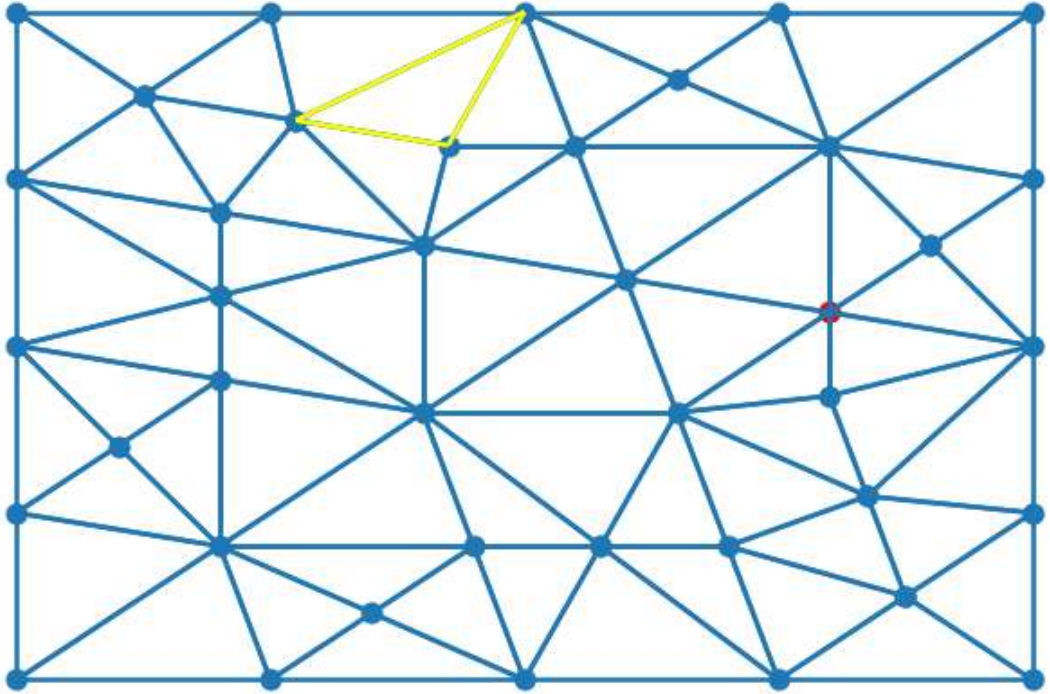
n - liczba punktów, d – maksymalny stopień wierzchołka w siatce

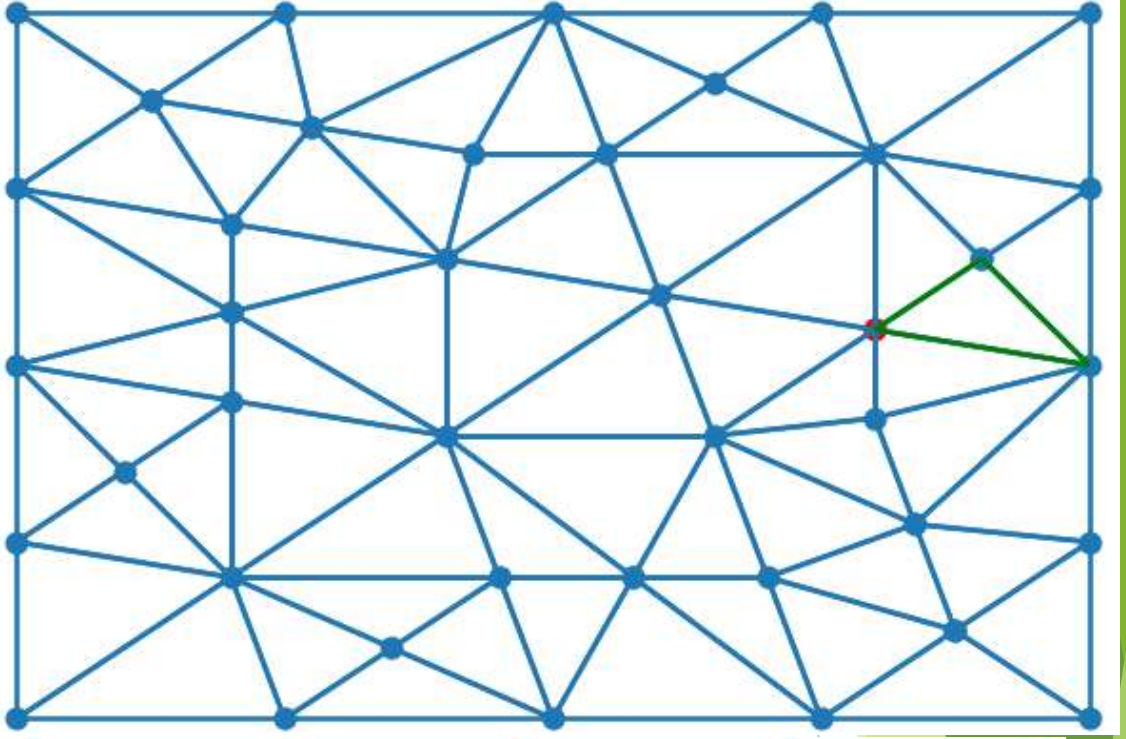


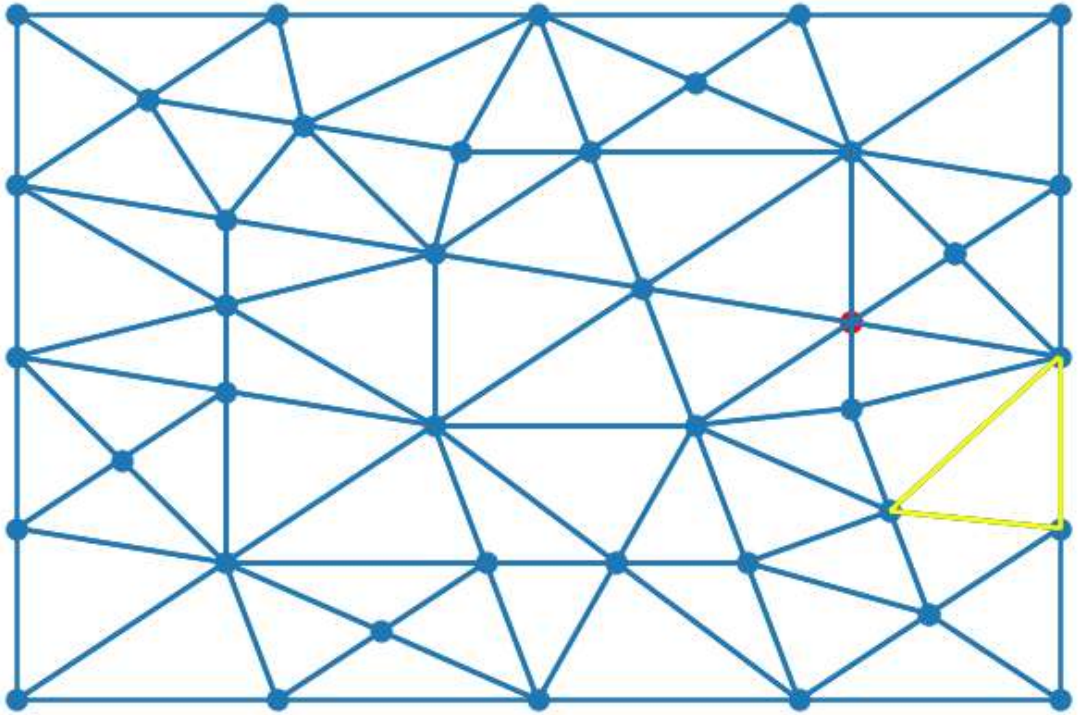


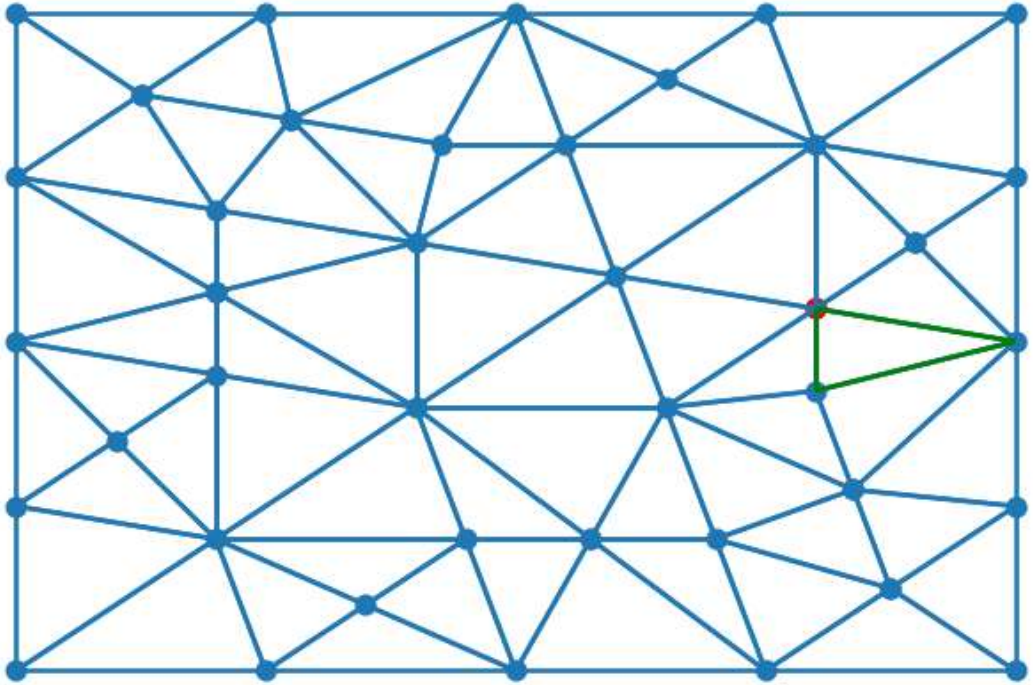












Reprezentacja druga

Algorytm polega na:

- sprawdzenie czy można ,,okrążyć'' dany wierzchołek

- wyznaczeniu wszystkich wychodzących pół-krawędzi z danego wierzchołka

oraz jednej krawędzi wchodzącej do danego wierzchołka jeśli nie da się okrążyć danego wierzchołka

- pozyskanie informacji o sąsiadujących wierzchołkach dzięki operacji edge.twin.vertex na wychodzących wierzchołkach

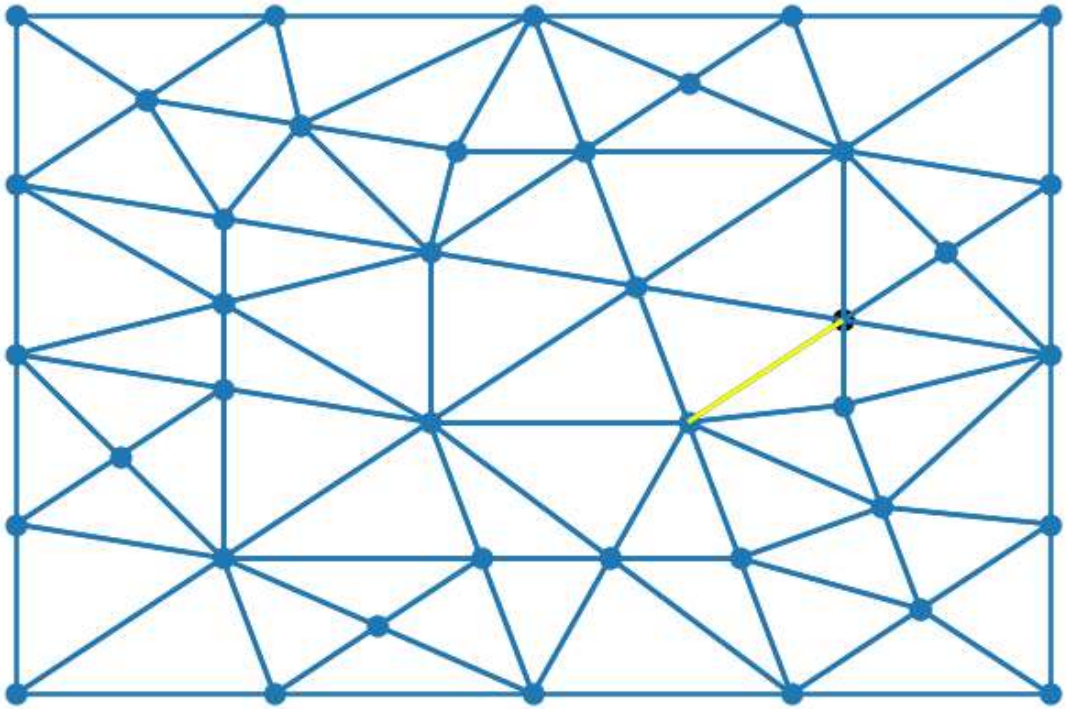
-powtórzeniu powyższych operacji dla wszystkich wierzchołków z wyznaczonej pierwszej warstwy

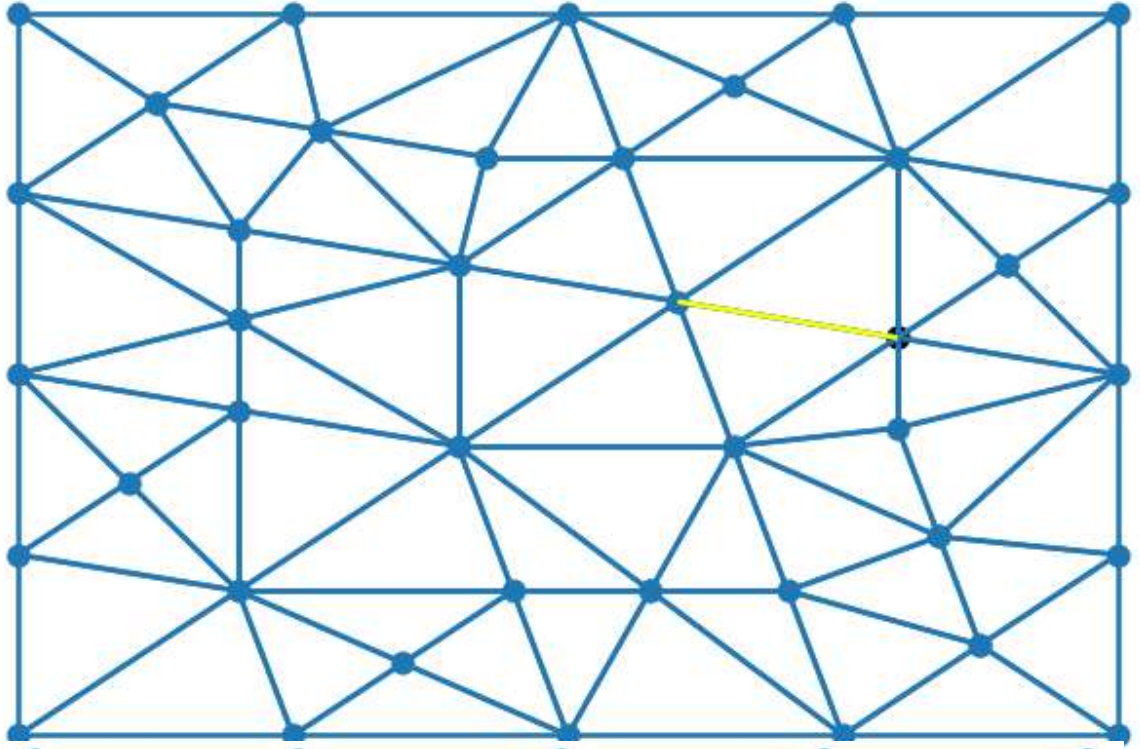
- usunięciu duplikujących się wierzchołków

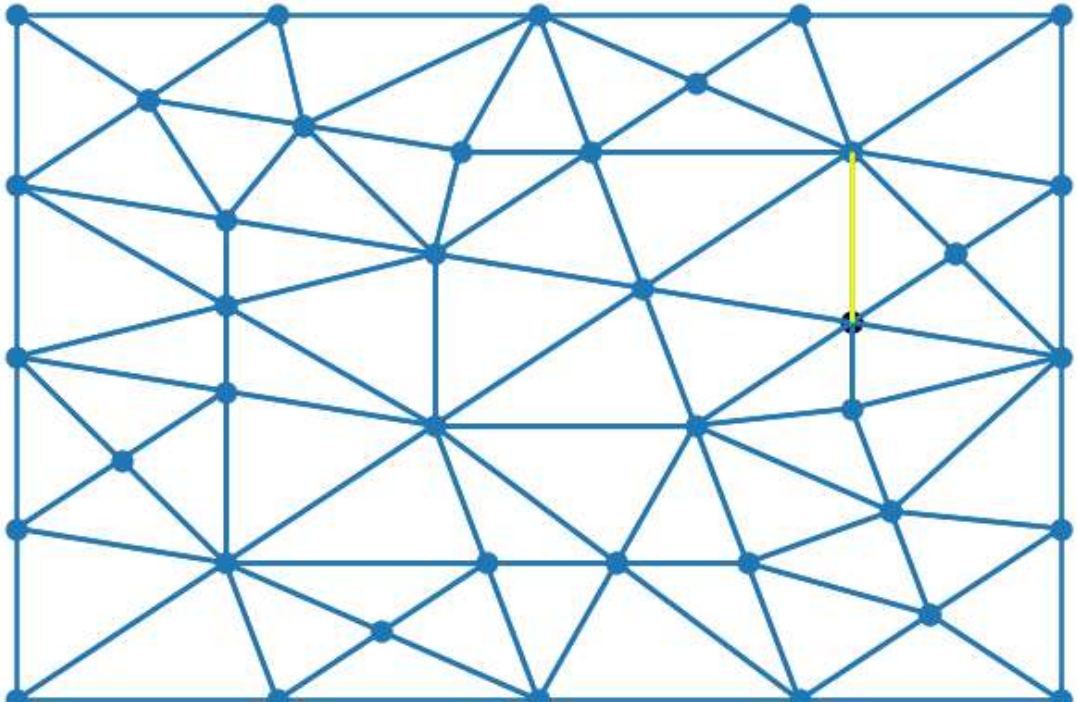
Złożoność czasowa : O(d^2)

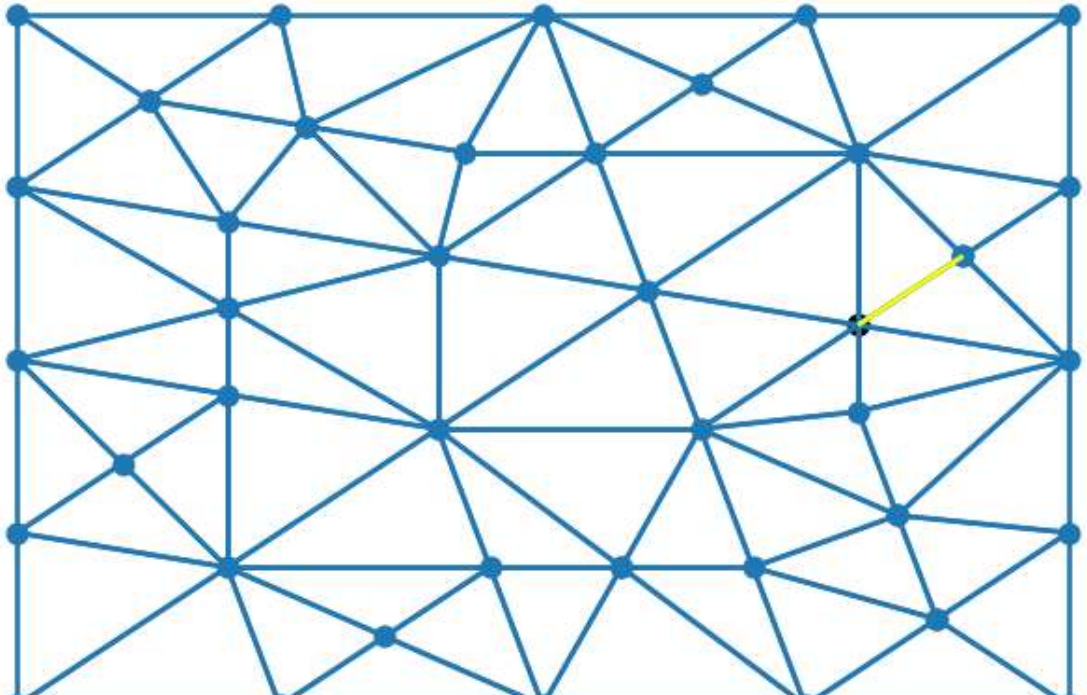
Złożoność pamięciowa : O(d^2) gdzie

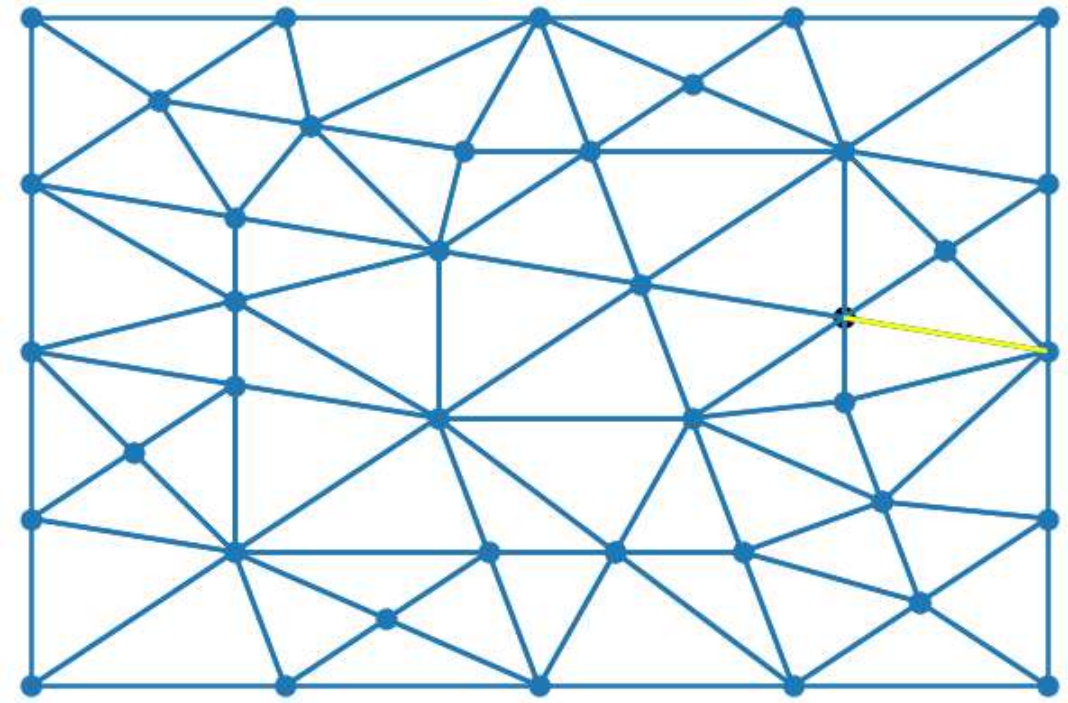
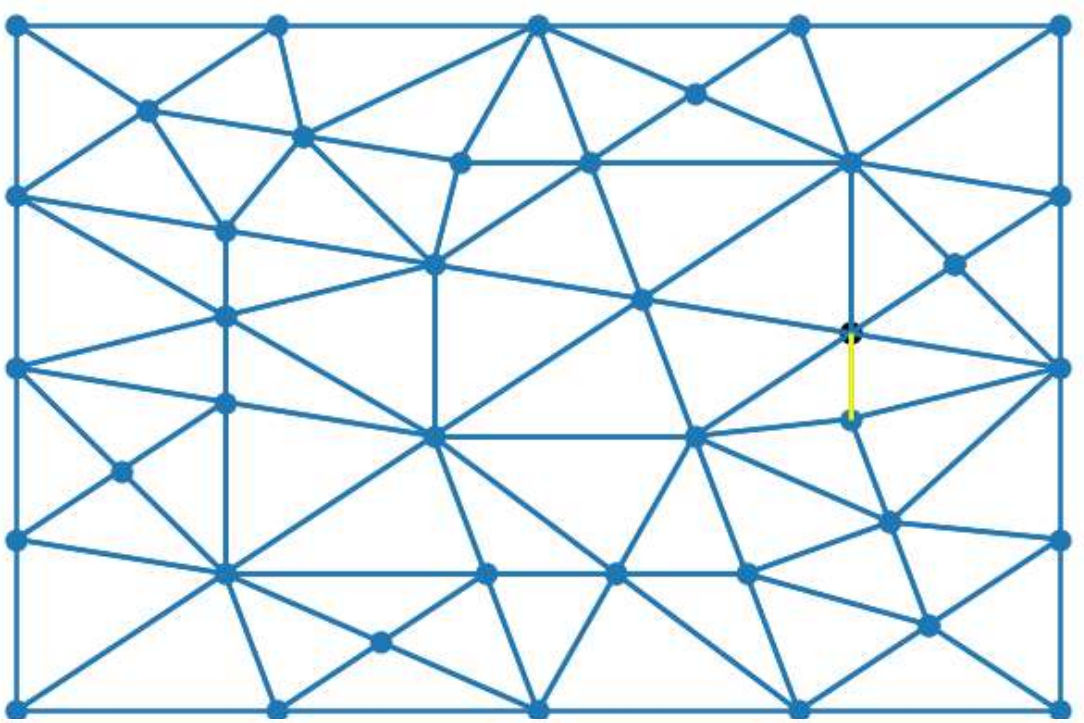
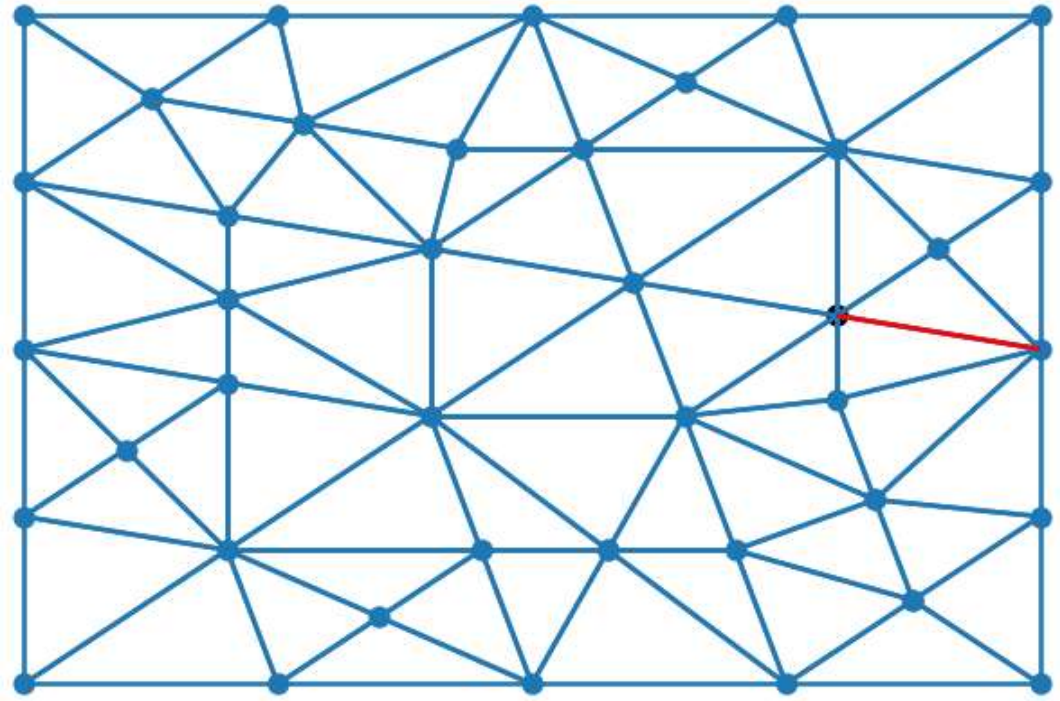
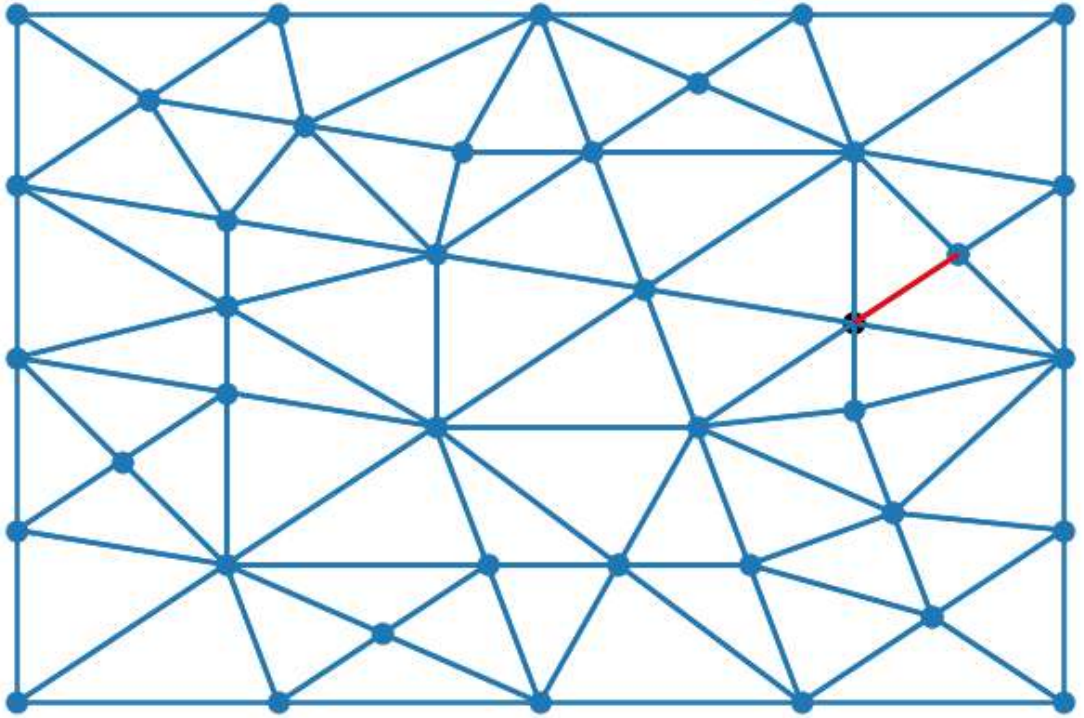
d – maksymalny stopień wierzchołka w siatce









OP2: wyznaczanie otoczenia dla wybranego trójkąta (kolejne warstwy incydentnych trójkątów – należy rozpatrzyć otoczenia składające się z jednej warstwy oraz dwóch warstw)

Reprezentacja pierwsza

Algorytm polega na:

- przeglądanie tablicy z trójkątami w celu znalezienia trójkątów zawierających dokładnie dwa takie same punkty jak wybrany trójkąt

- powtórzenie podpunktu powyżej dla każdego trójkąta z pierwszej warstwy

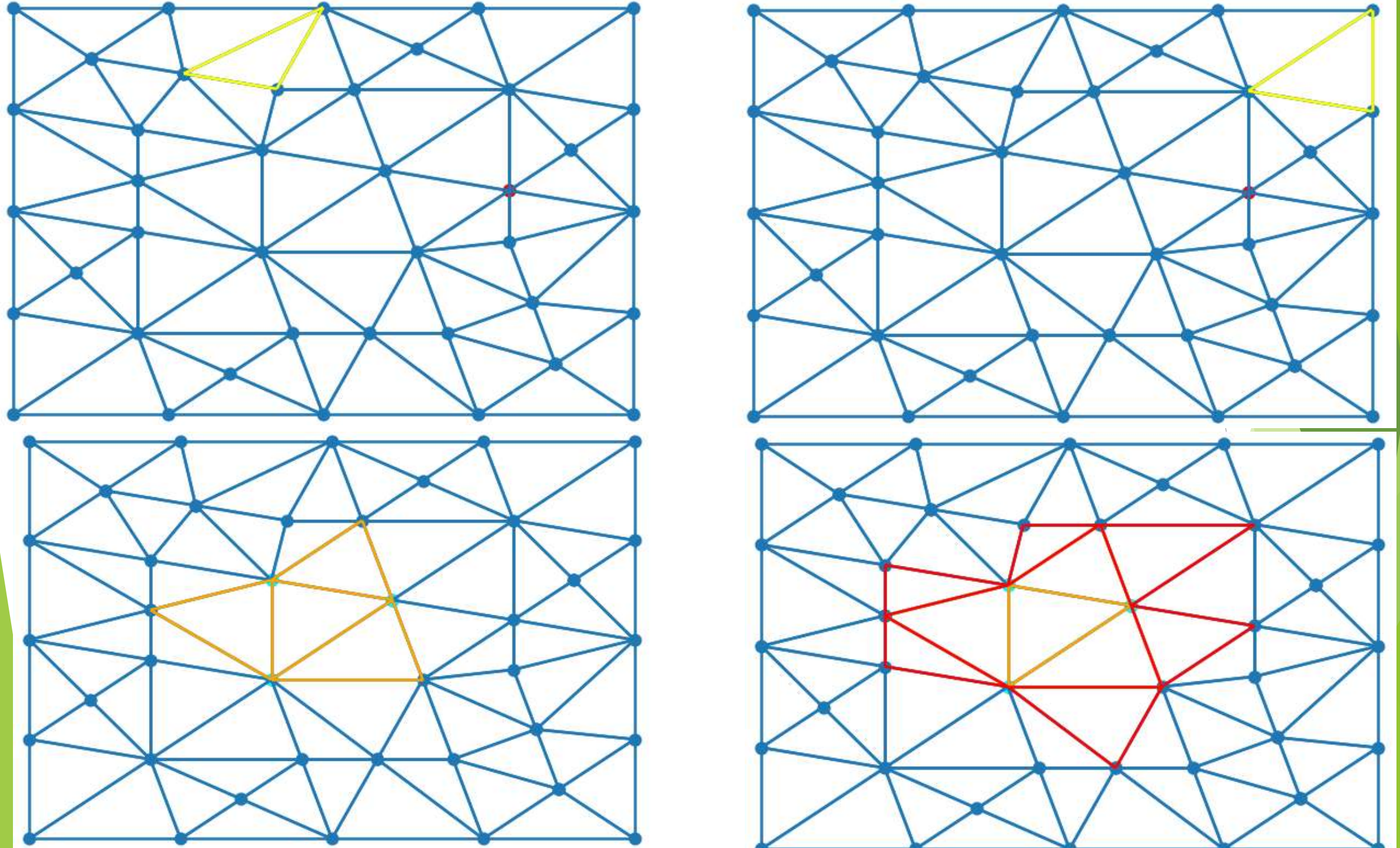
w celu utworzenia drugiej warstwy

-usunięcie postarzających się trójkątów

Złożoność czasowa : O(n)

Złożoność pamięciowa : O(1)

n - liczba punktów



Reprezentacja druga

Algorytm polega na:

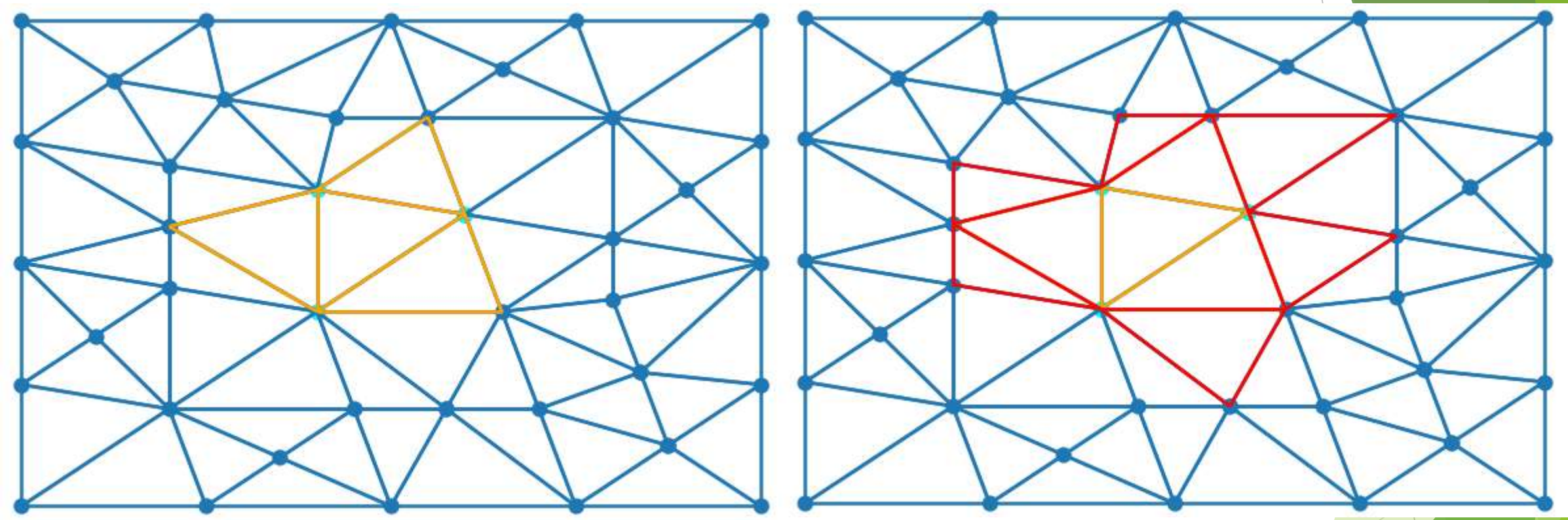
- utworzeniu pierwszej warstwy incydentnych trójkątów z trójkątów które zawierają krawędzie przeciwne do krawędzi wybranego trójkąta

-powtórzeniu powyższych operacji dla wszystkich wierzchołków z wyznaczonej pierwszej warstwy

- usunięciu duplikujących się wierzchołków

Złożoność czasowa : O(1)

Złożoność pamięciowa : O(1)



OP3: przeglądanie incydentnych trójkątów od wybranego trójkąta w kierunku wybranego punktu (dla odszukania trójkąta zawierającego dany punkt)

Reprezentacja pierwsza

Algorytm polega na:

- przeglądanie tablicy z trójkątami w celu znalezienia nie odwiedzonych jeszcze trójkątów zawierających dokładnie dwa takie same punkty jak wybrany trójkąt i wsadzenia je do kolejki, wraz z zapisaniem jego odległości od trójkąta startowego, jego dzieci oraz rodzica

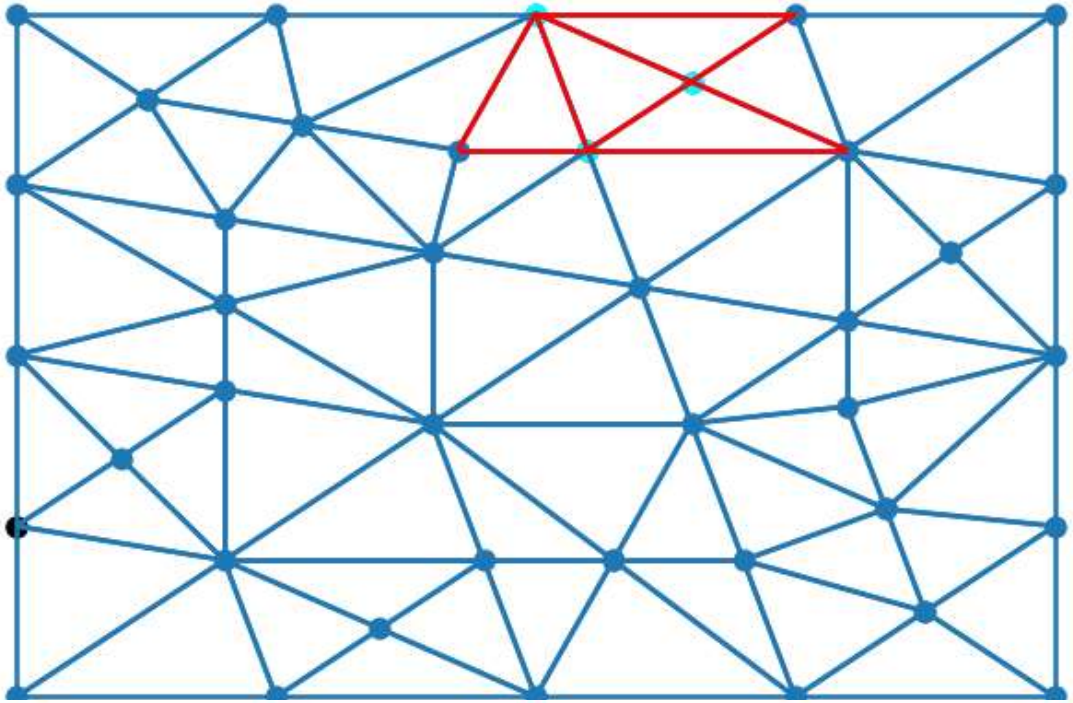
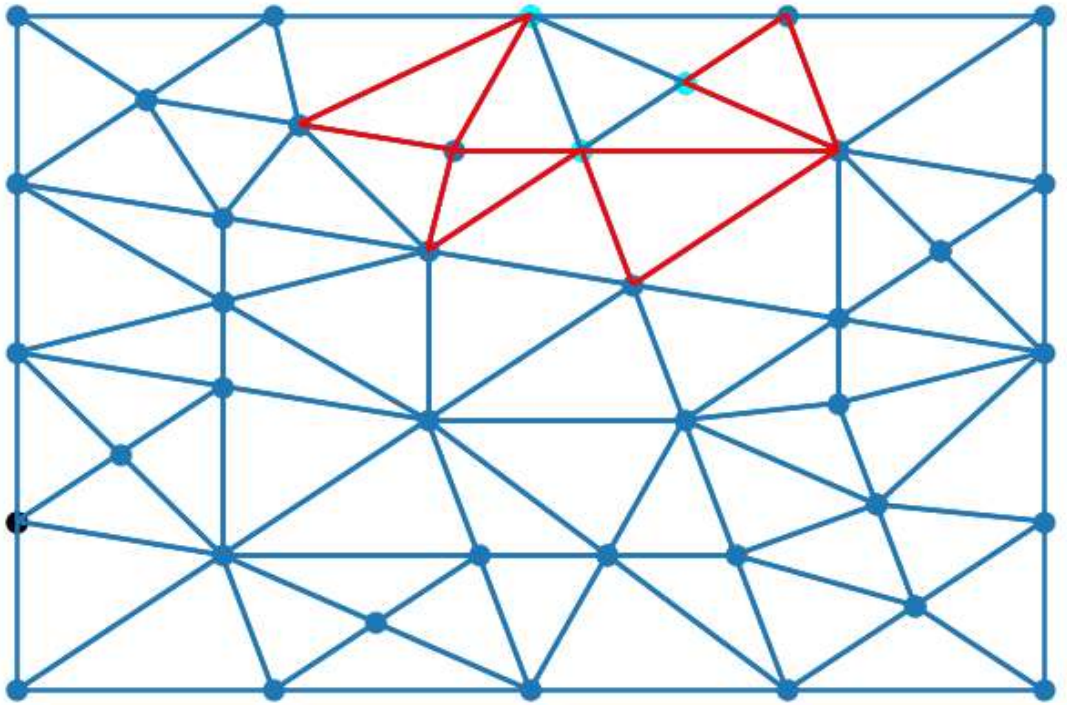
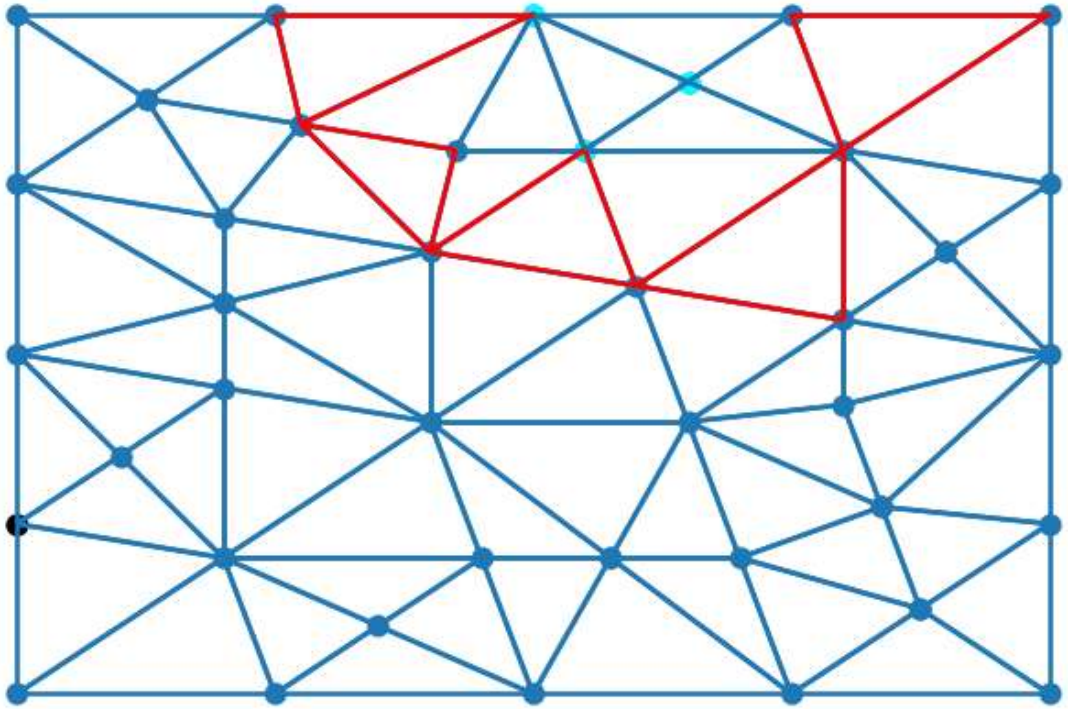
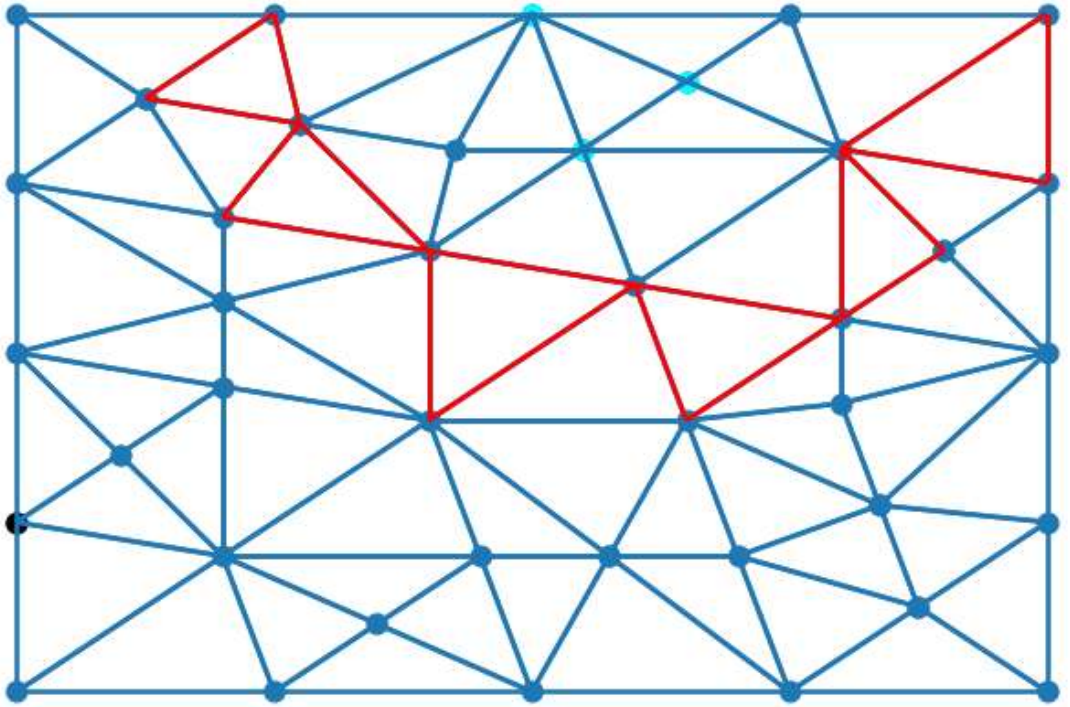
- powtórzenie podpunktu powyżej do momentu aż kolejka się opróżni

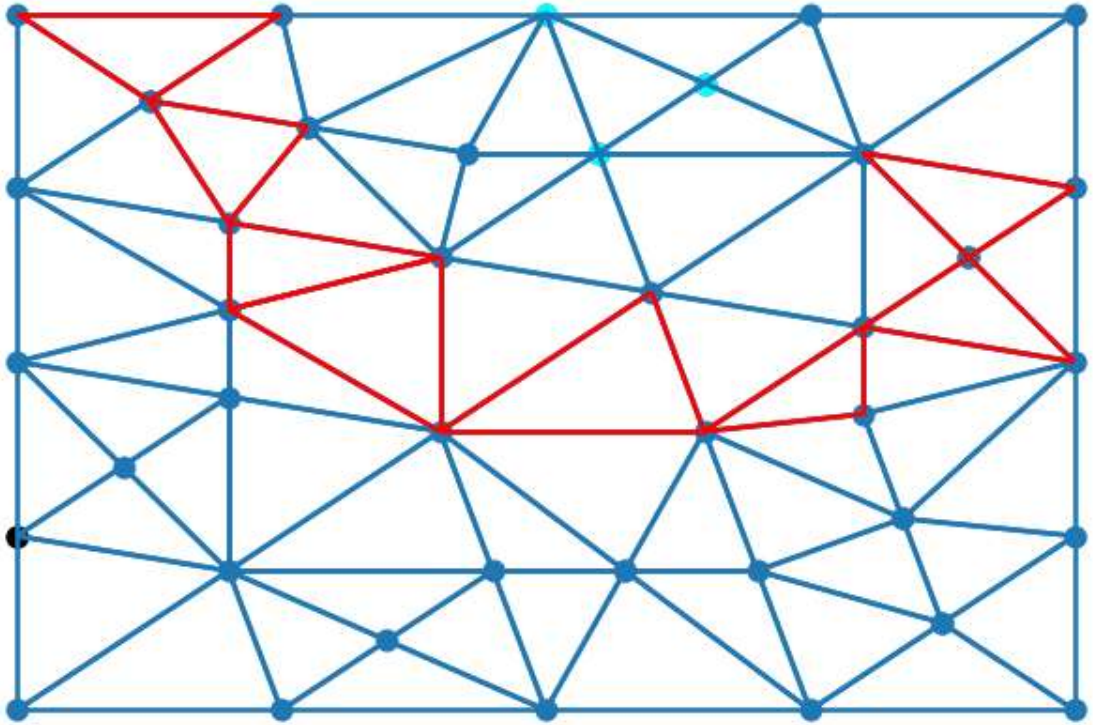
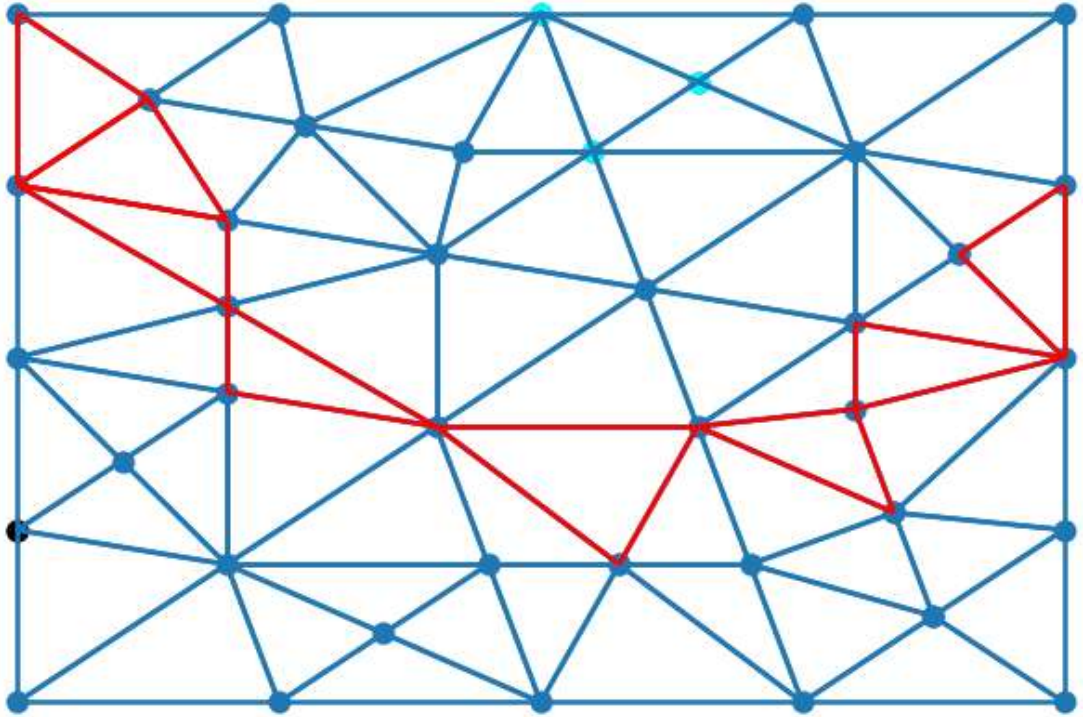
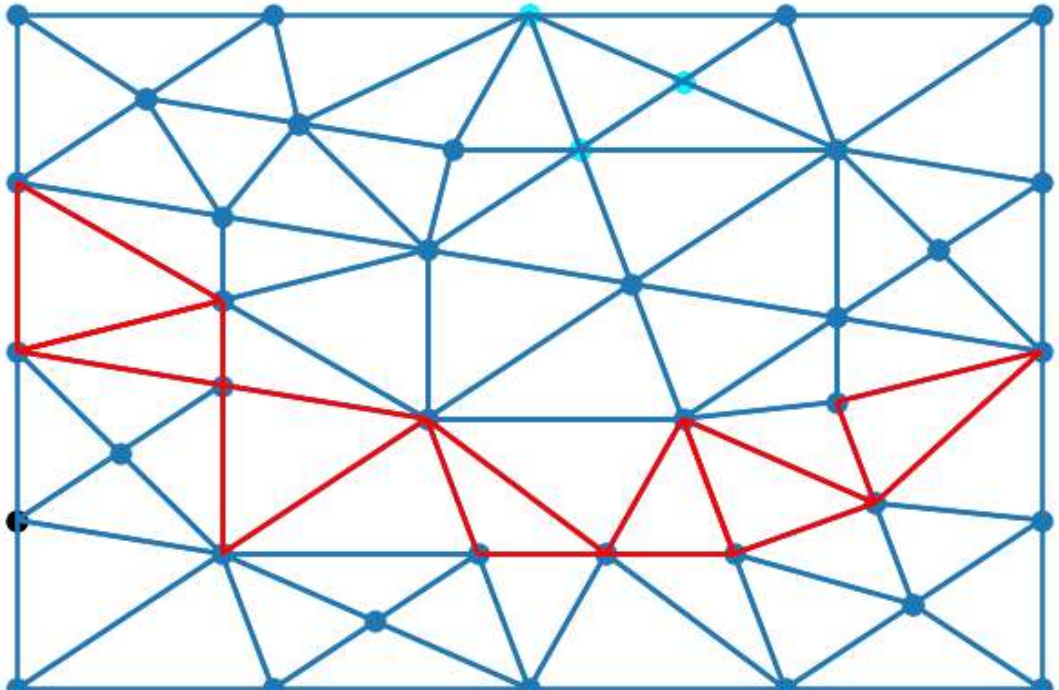
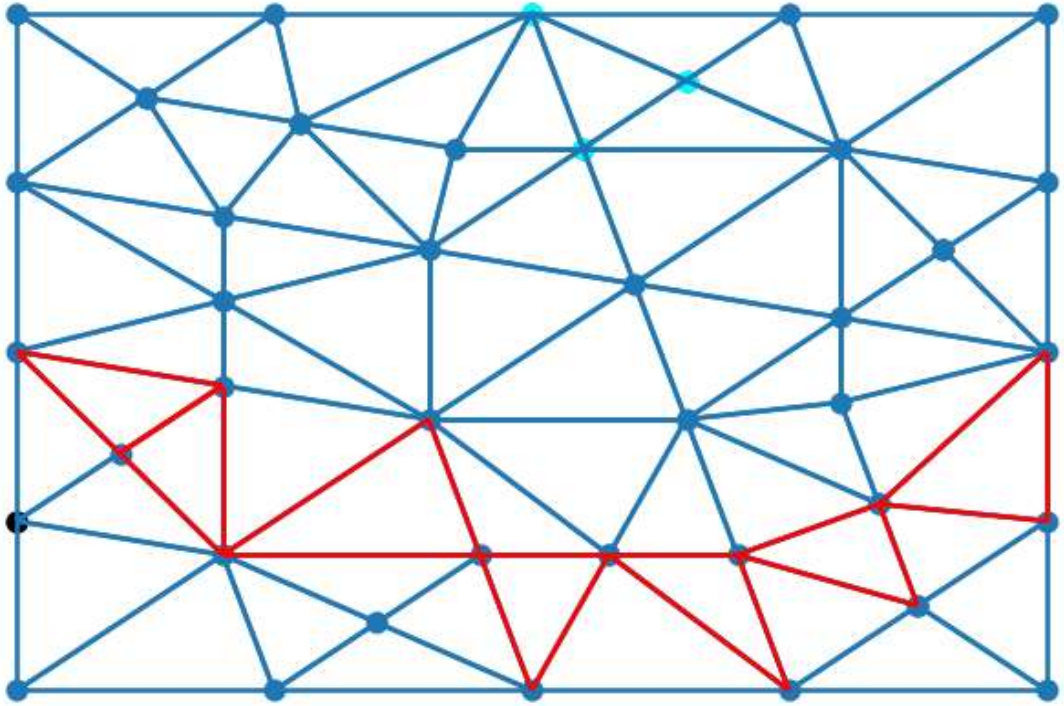
- znalezienie najbliższego trójkąta zawierającego wskazany punkt

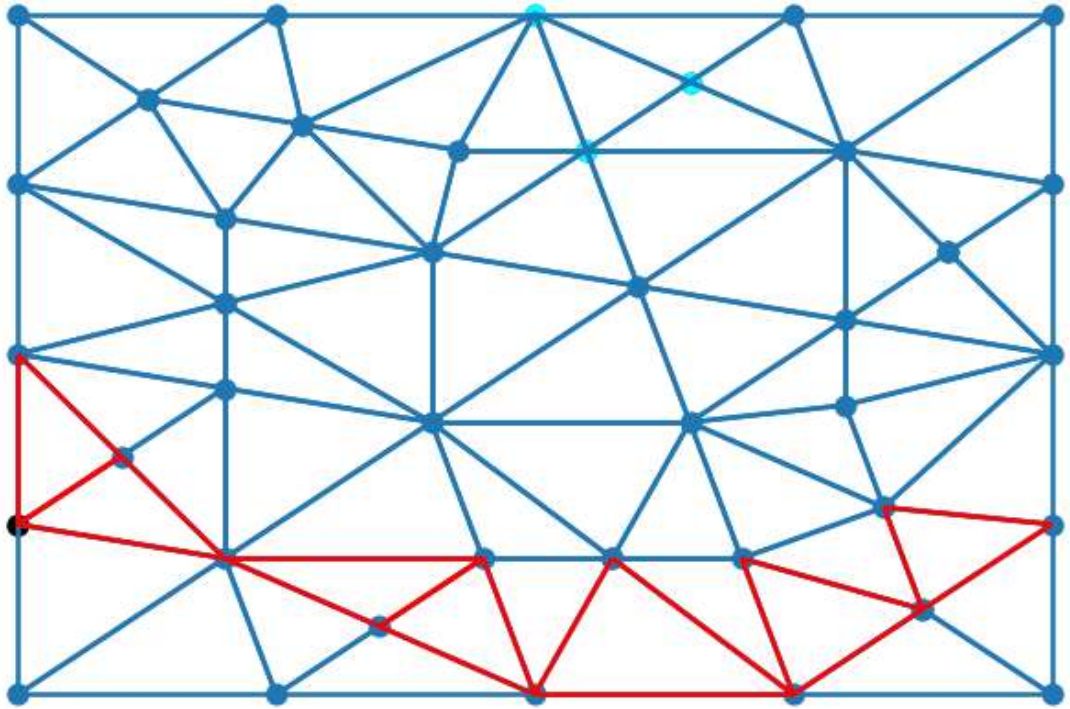
Złożoność czasowa : O(n^2)

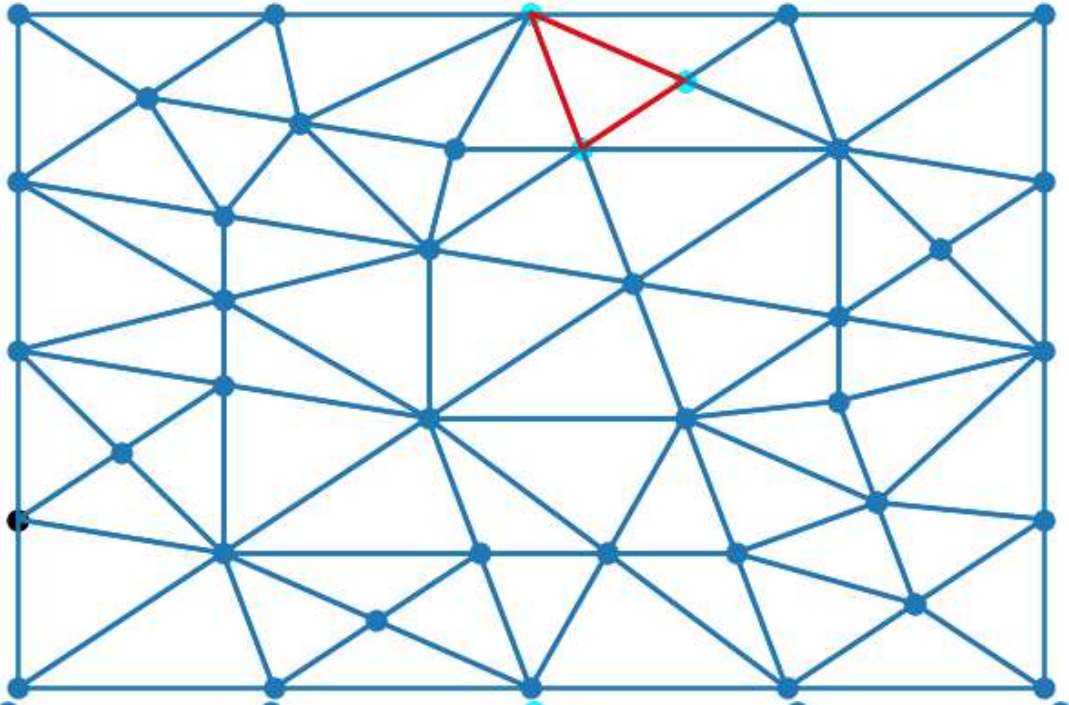
Złożoność pamięciowa : O(n)

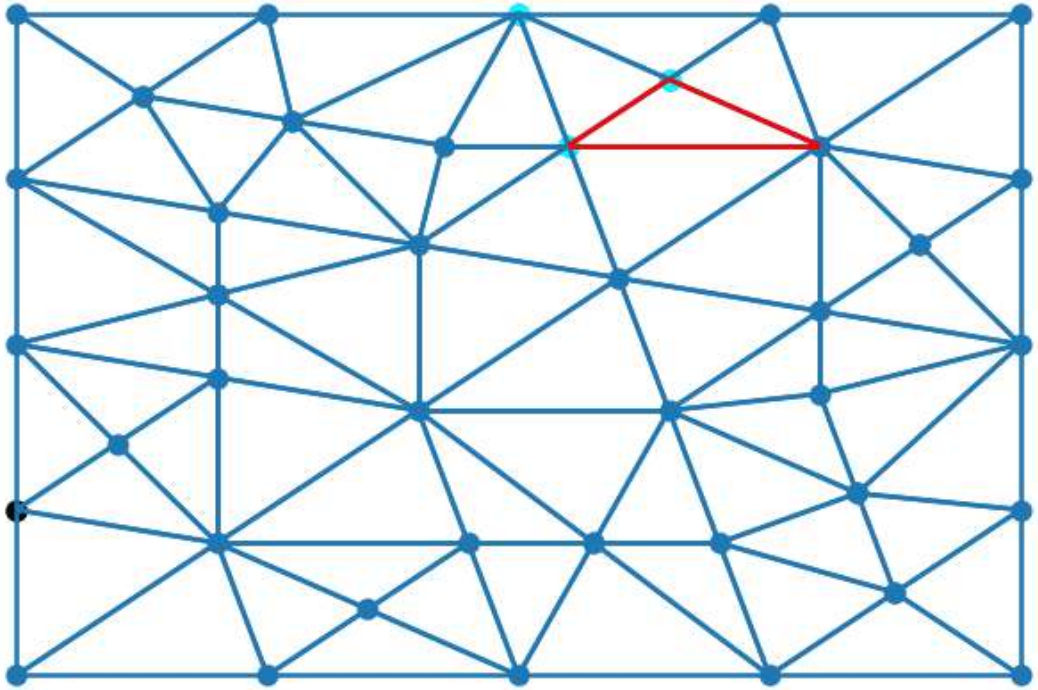
n - liczba punktów

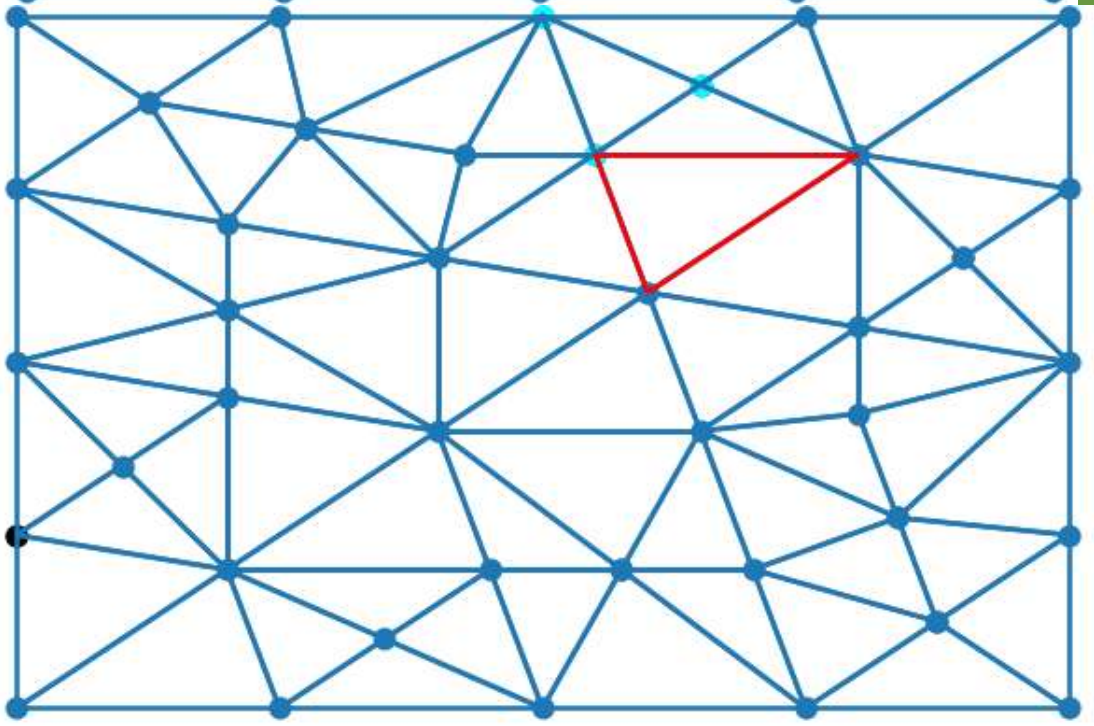





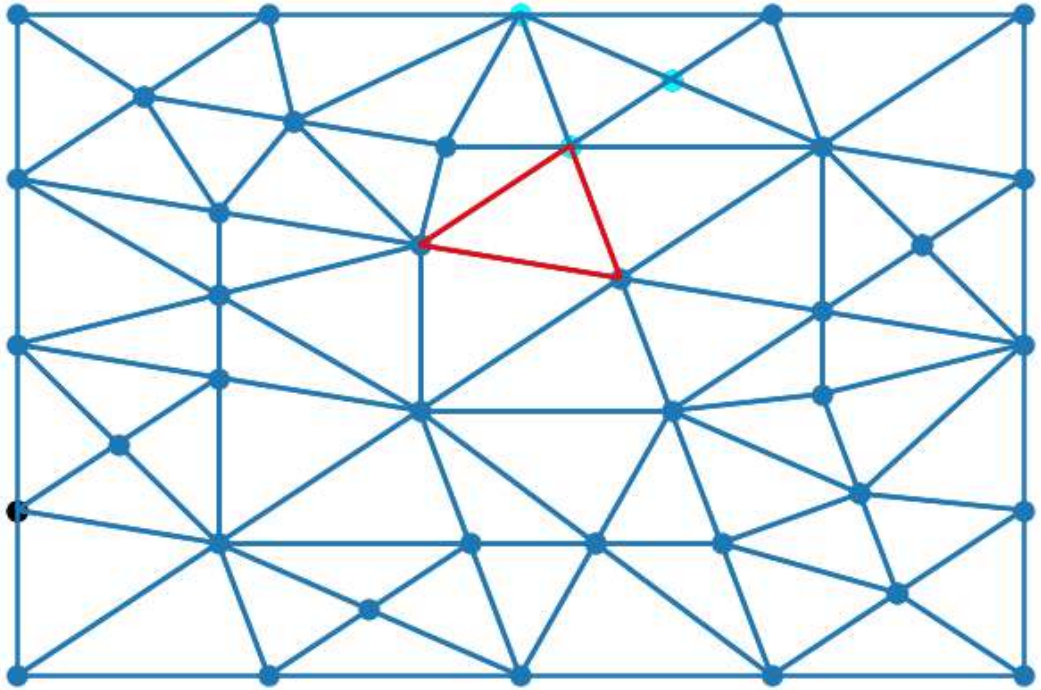
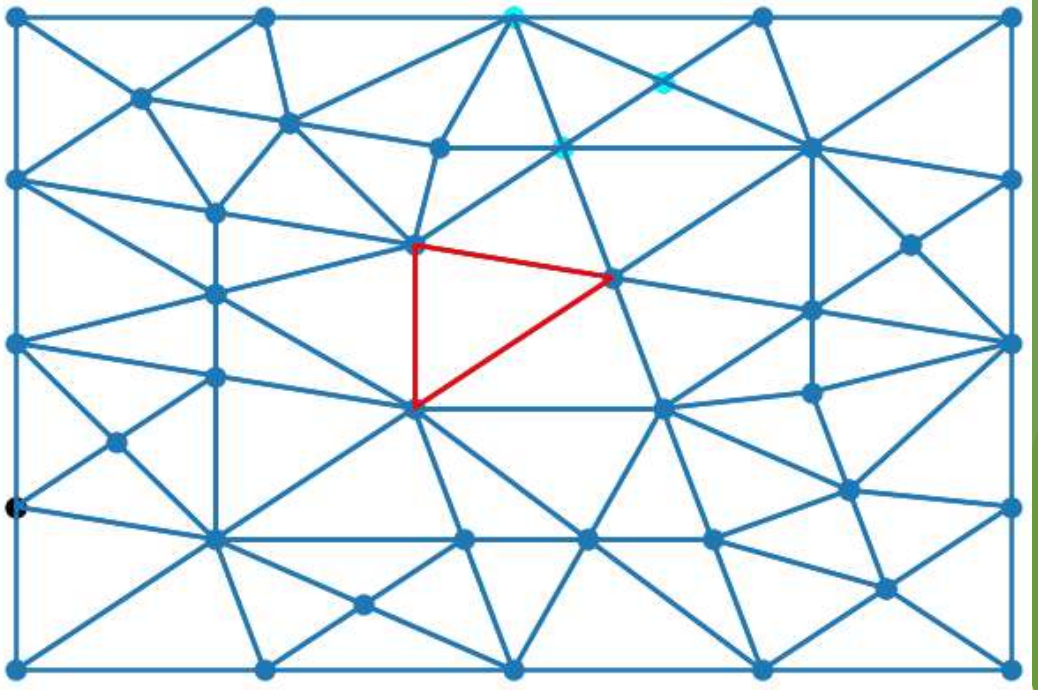
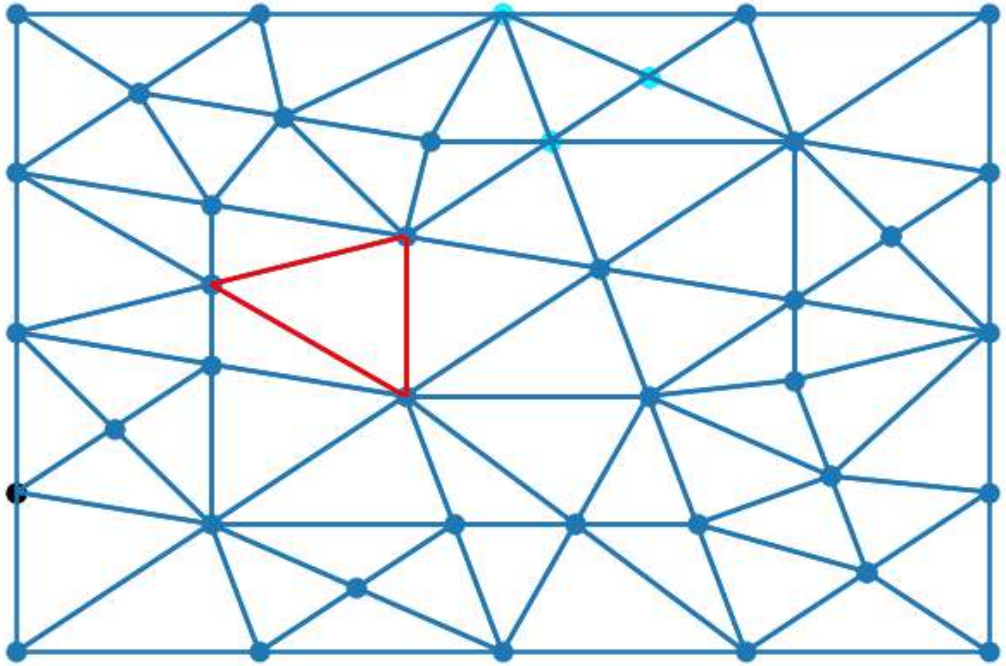
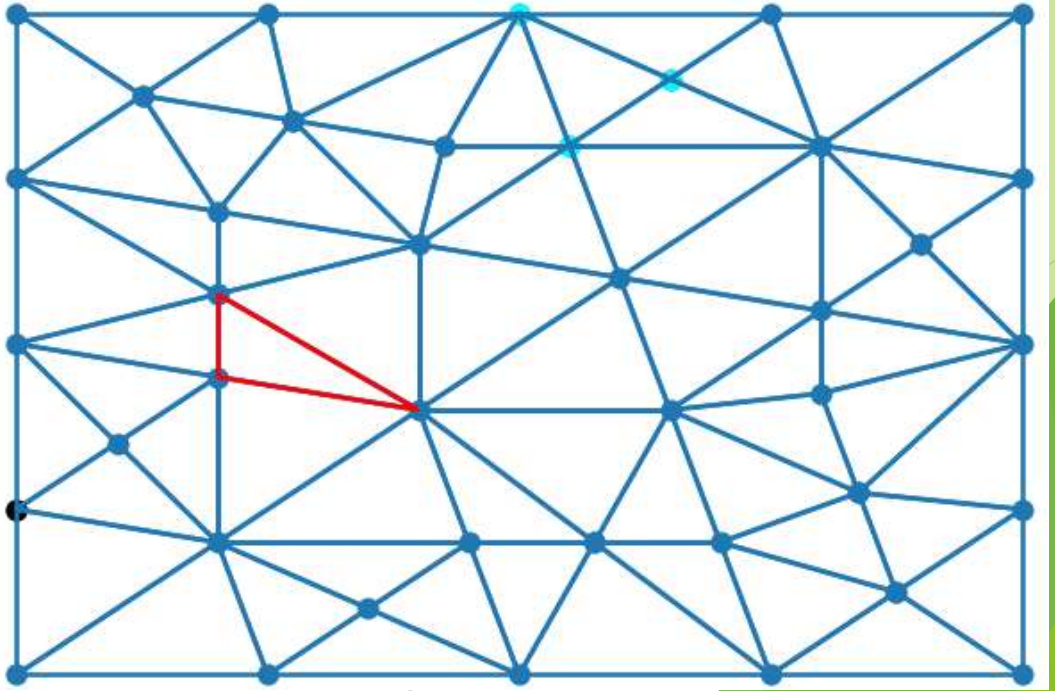





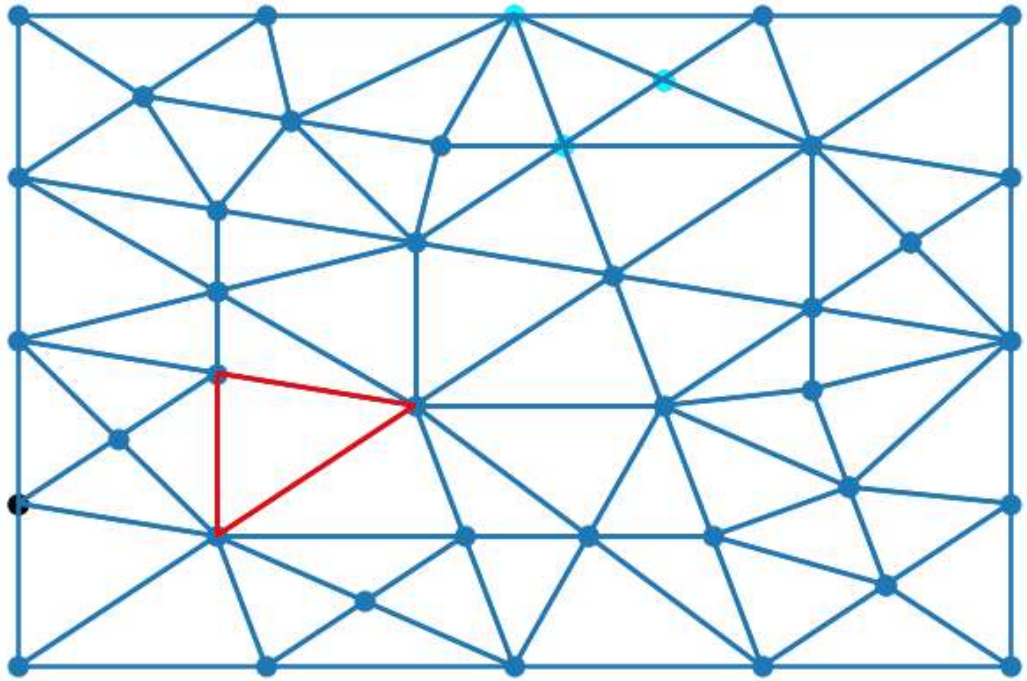


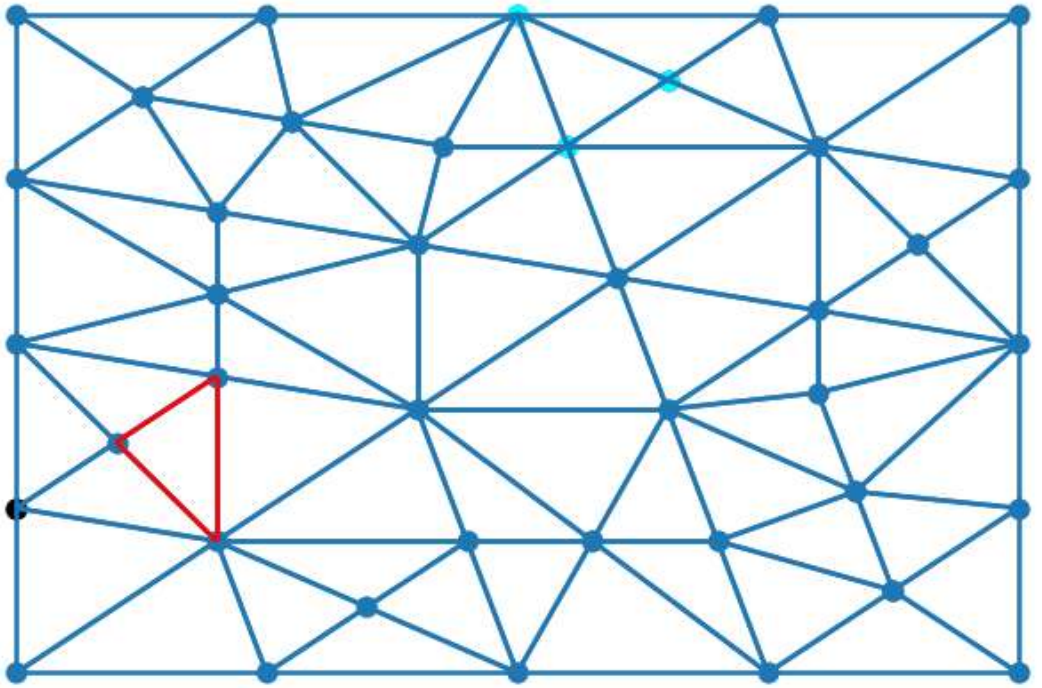


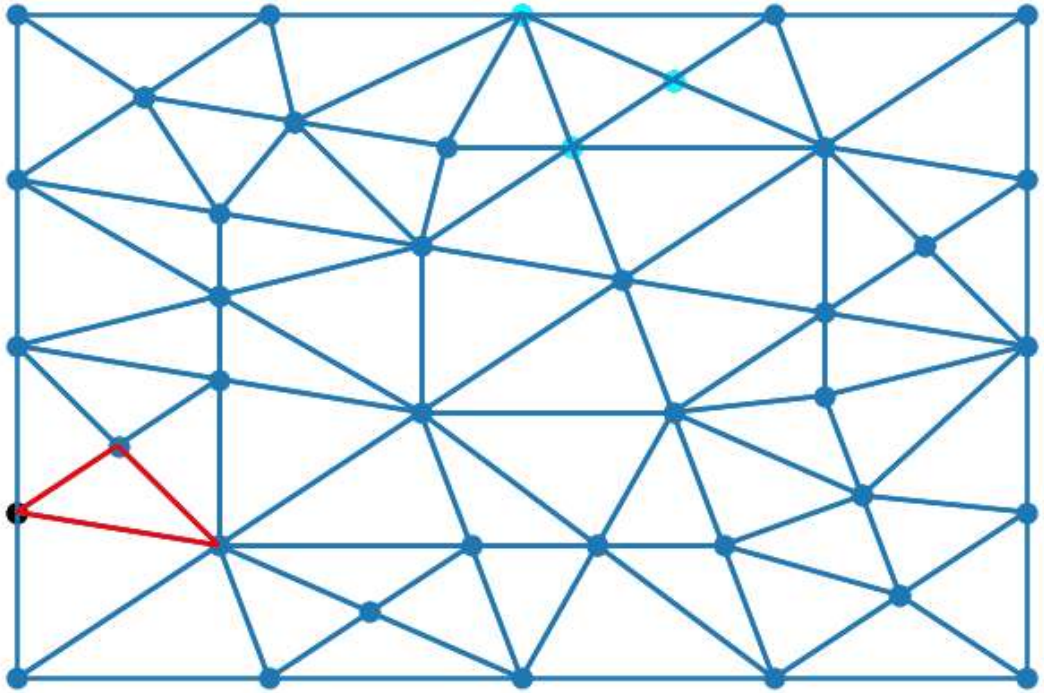










Reprezentacja druga

Algorytm polega na:

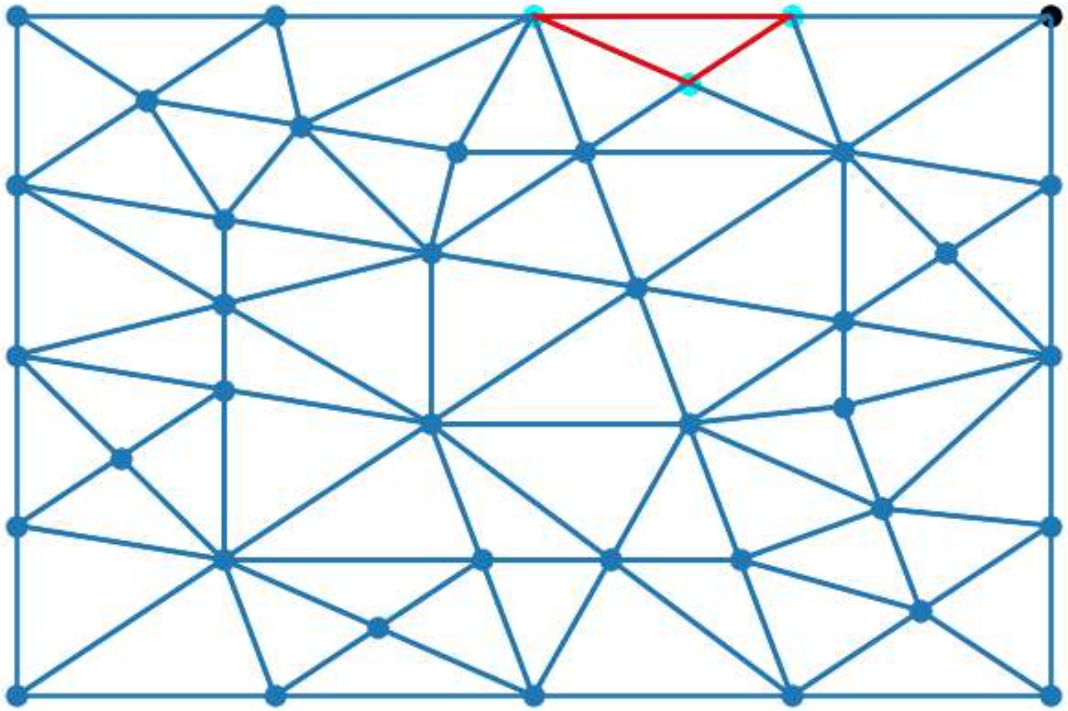
- przeglądaniu nie odwiedzonych sąsiednich trójkątów i wsadzenie ich do kolejki

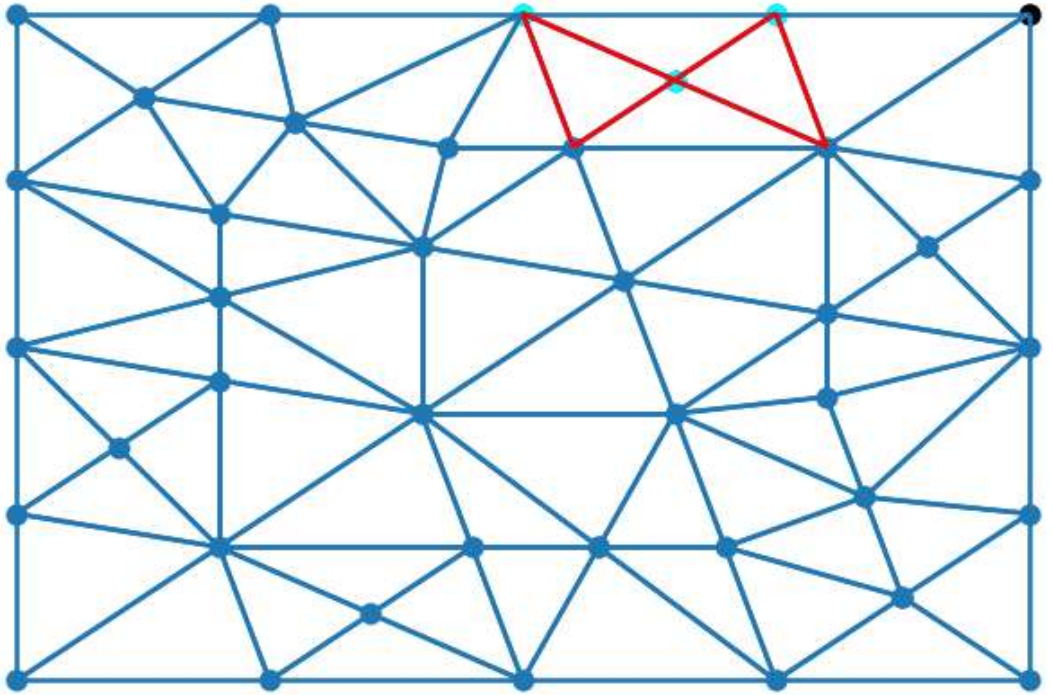
-powtórzeniu powyższej operacji do momentu aż kolejka będzie pusta

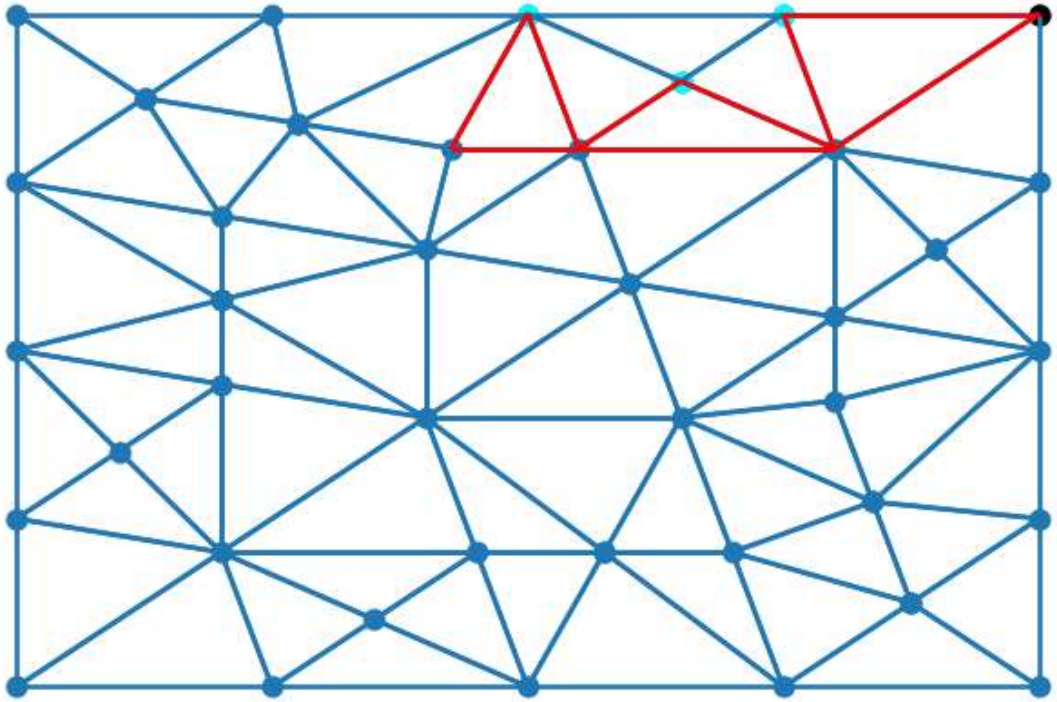
Złożoność czasowa : O(n)

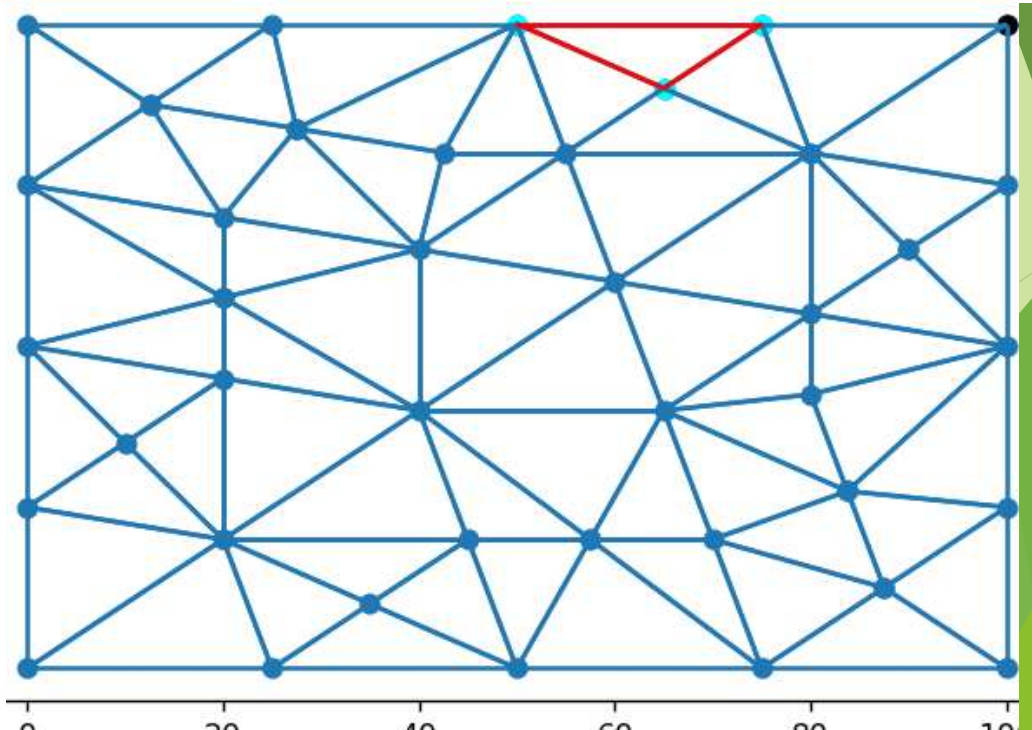
Złożoność pamięciowa : O(n)

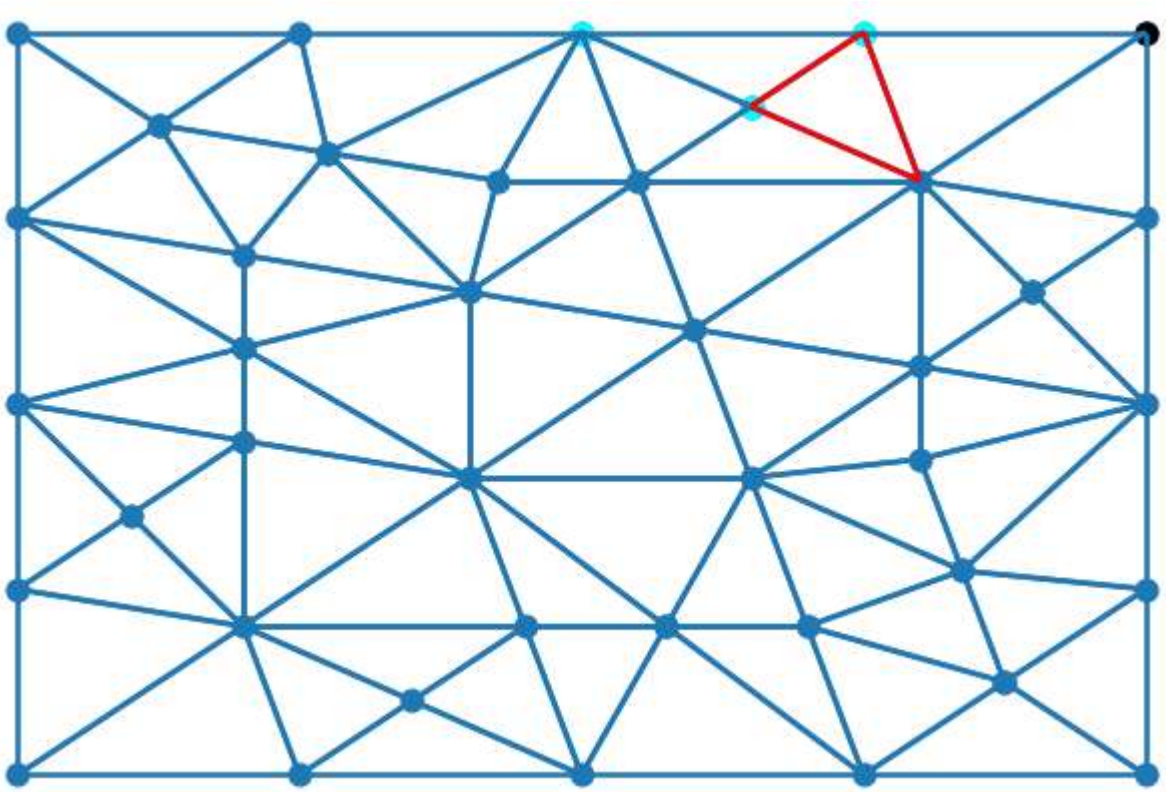
n - liczba punktów

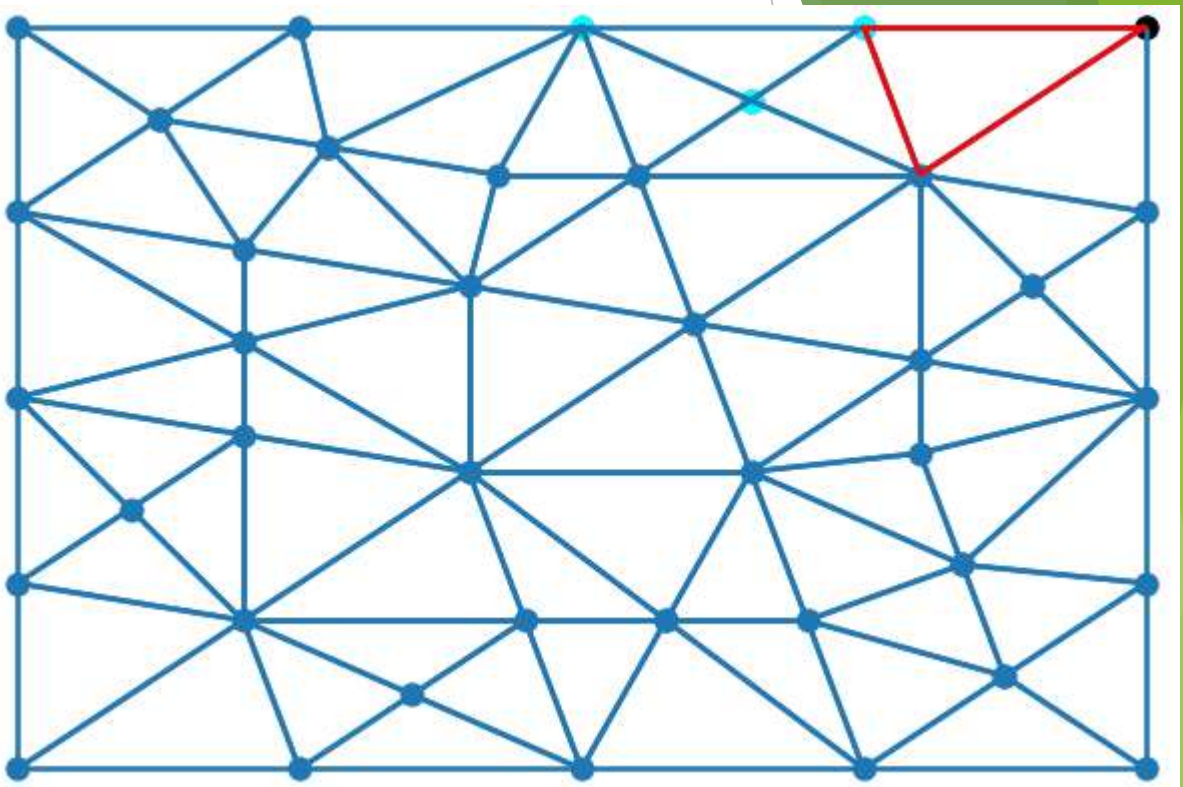







0
20
40
60
80





OP4: zamiana krawędzi dla wskazanej pary incydentnych trójkątów

Reprezentacja pierwsza

Algorytm polega na:

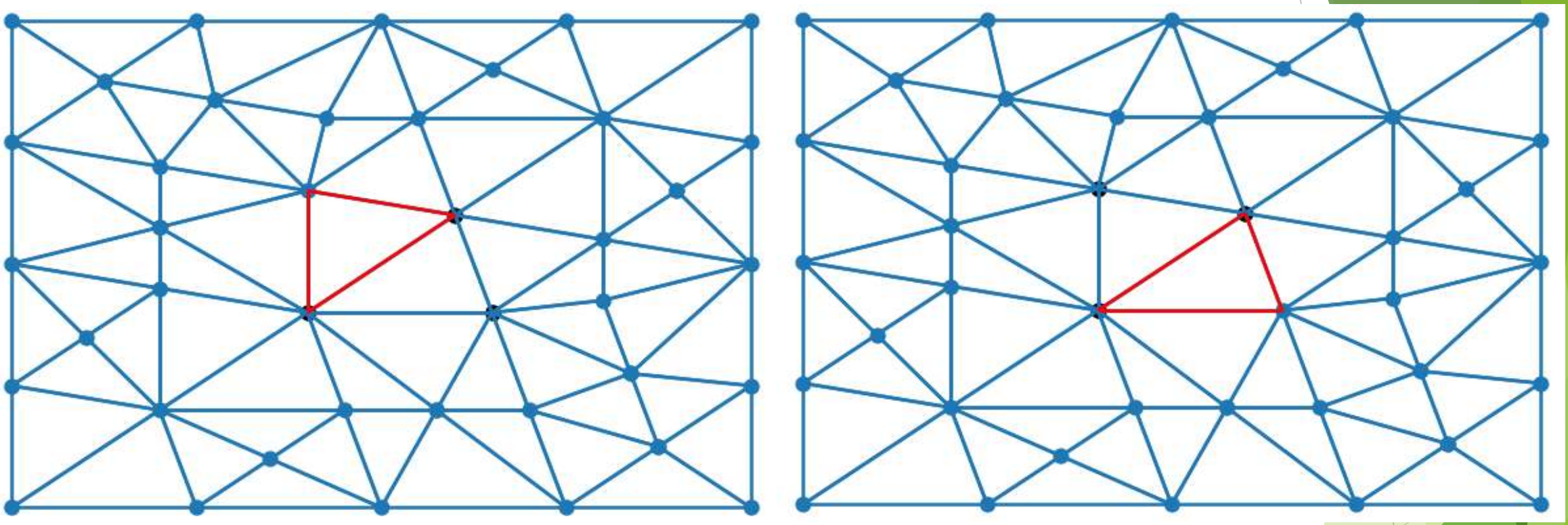
- przeglądaniu tablicy z trójkątami w celu znalezienia incydentnych trójkątów i wybranie jednego z nich

- zamiana punktów dla wskazanej pary trójkątów

Złożoność czasowa : O(n)

Złożoność pamięciowa : O(1)

n - liczba punktów



Reprezentacja druga

Algorytm polega na:

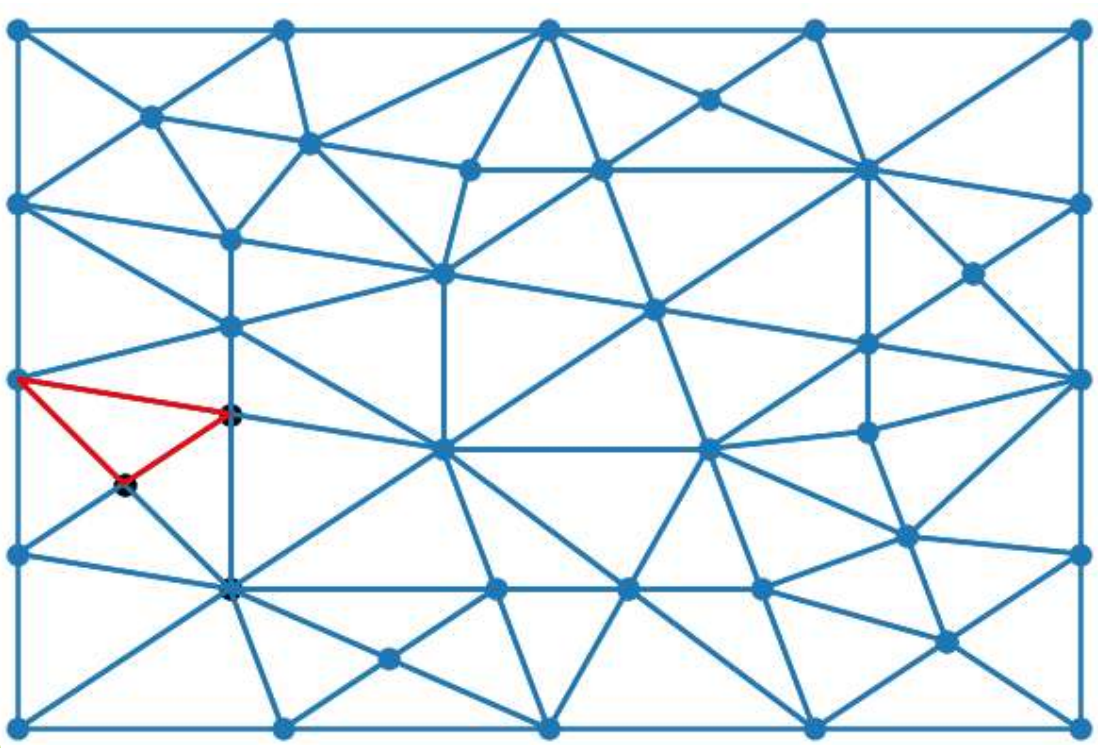
- zamianie ,,pół krawędzi'' definiujących dane trójkąty

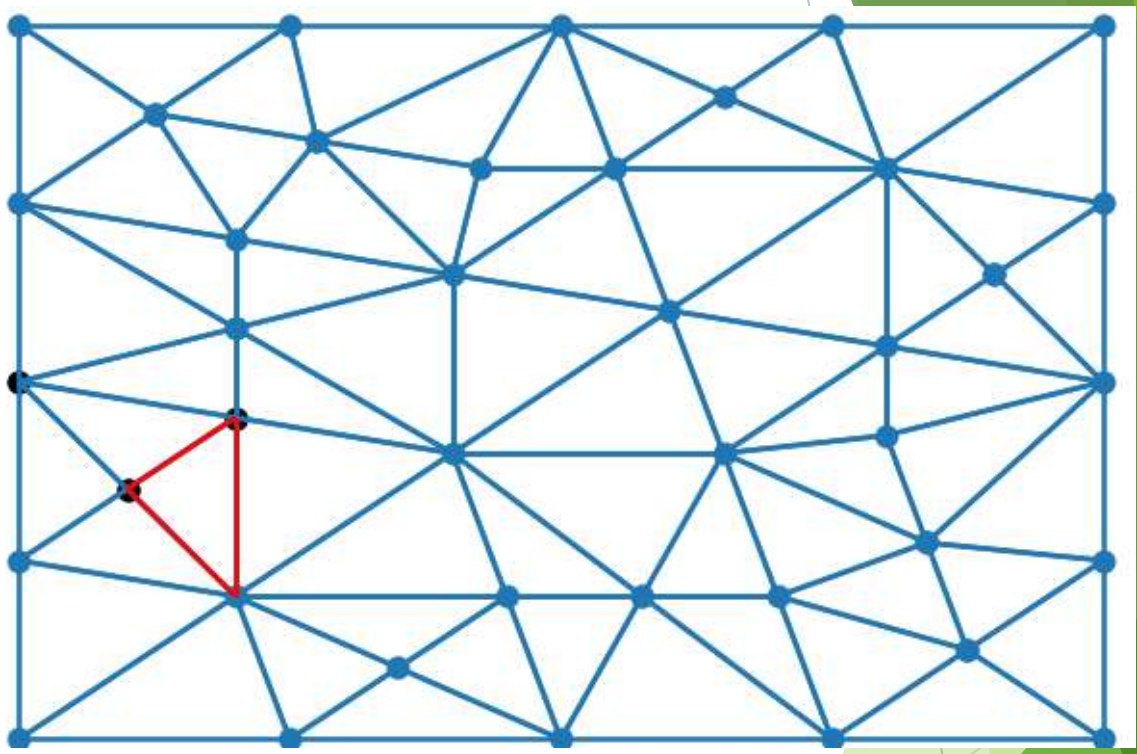
-zamianie ,,twarzy'' przypisanej danej ,,pół krawędzi'

-zamianie wszystkich wierzchołków wychodzących dla danych ,, pół krawędzi''

Złożoność czasowa : O(1)

Złożoność pamięciowa : O(1)





Porównanie czasu wykonania operacji dla obydwu prezentacji

Reprezentacja	Otoczenie wierzchołka	Otoczenie trójkąta	Przeglądanie incydentnych trójkątów	Zamiana krawędzi	Ilość punktów/ trójkątów
Podstawowa	0s	0s	0.000998s	0s	974/1421
Half-edge	0s	0s	0s	0s	
Podstawowa	0.010004s	0.0s	5.462557s	0.002233s	3896/5684
Half-edge	0s	0s	0.010002s	0s	
Podstawowa	0.079552s	0.0119984s	88.725710s	0.005125s	15154/22498
Half-edge	0.00399s	0s	0.220156s	0s	
Podstawowa	0.229392s	0.037896s	-	0.024363s	34431/50798
Half-edge	0.00567s	0.00099s	0.803517s	0s	

Wnioski

DCEL umożliwia łatwy dostęp do informacji o krawędziach i wierzchołkach siatki, a także umożliwia łatwe operacje na siatce takie jak przeszukiwanie sąsiednich krawędzi i trójkątów, dzięki czasu znacznie spada czas wykonanania tych operacji oraz ilość potrzebnej pamięci.