Algorytmy Geometryczne

Ćwiczenie 3 - Triangulacja wielokątów monotonicznych

Jakub Własiewicz - Grupa 2 - Poniedziałek 13:00
2025-11-18

1. Dane techniczne

Program został uruchomiony na komputerze z następującymi specyfikacjami:
- **System Operacyjny -** Fedora Linux 43
- **Architektura Procesora -** x86_64
- **Procesor -** AMD Ryzen 7 7840HS
- **Język -** Python 3.14.0

Ćwiczenie realizowane było w środowisku *Jupiter*, do wizualizacji zostało użyte narzędzie stworzone przez koło naukowe *BIT* oraz następujące biblioteki:
- **matplotlib**
- **numpy**
- **pandas**

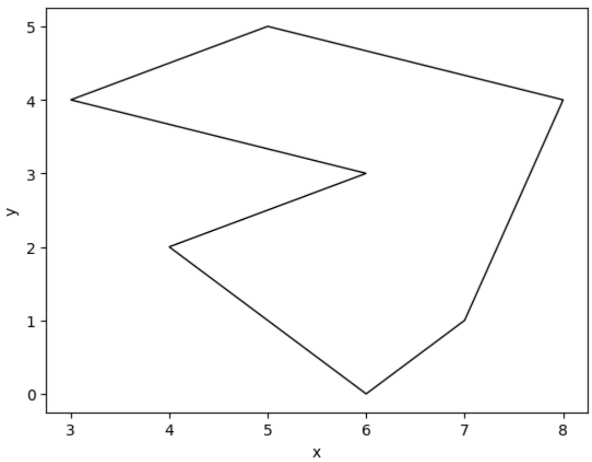
Do obliczeń została użyta tolerancja dla zera $\varepsilon = 10^{-16}$, oraz liczby zmienno-przecinkowe o rozmiarze 64 bitów.

2. Opis ćwiczenia

Ćwiczenie polegało na sprawdzeniu czy podany wielokąt jest y-monotoniczny, klasyfikacji wierzchołków w dowolnym wielokącie oraz triangulacji wielokąta y-monotonicznego, w tym wizualizacji, analizy i stwierdzenia wniosków.

3. Wstęp teoretyczny

3.1. Wielokąt y-monotoniczny



Rysunek 1: Przykładowy wielokąt y-monotoniczny

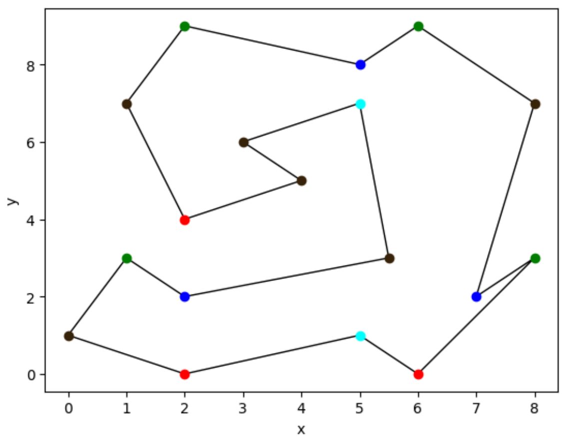
Wielokąt nazywamy y-monotonicznym, jeżeli możemy podzielić jego punkty na dwa takie łańcuchy, że współrzędne y-owe punktów są niemalejące w jednym z nich oraz nierosnące w drugim.

Zaimplementowany algorytm sprawdzający y-monotoniczność wielokąta polega na znalezieniu najniższego punktu względem osi y, a następnie ustawieniu dwóch indeksów, które poruszają się odpowiednio wzdłuż lewego i prawego łańcucha. Idąc kolejnymi punktami, jeżeli jeden z nich znajdzie się w punkcie o mniejszej y-owej współrzędnej niż poprzedni badany punkt, zapamiętujemy, że jeden z nich „zawrócił”. Jeżeli wydarzy się to po raz drugi, to wielokąt nie jest y-monotoniczny.

3.2. Klasyfikacja wierzchołków

Możemy zaklasyfikować wierzchołki wielokąta następująco:

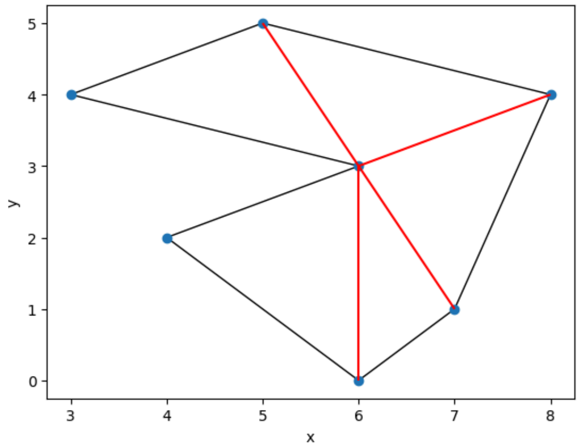
- **Początkowy** - obaj jego sąsiedzi leżą poniżej i kąt wewnętrzny $< \pi$
- **Końcowy** - obaj jego sąsiedzi leżą powyżej i kąt wewnętrzny $< \pi$
- **Łączący** - obaj jego sąsiedzi leżą powyżej i kąt wewnętrzny $> \pi$
- **Dzielący** - obaj jego sąsiedzi leżą poniżej i kąt wewnętrzny $> \pi$
- **Prawidłowy** - jeden sąsiad powyżej, drugi poniżej



Rysunek 2: Przykładowa klasyfikacja wierzchołków wielokąta

3.3. Triangulacja wielokąta monotonicznego

3.3.1. Definicja triangulacji



Rysunek 3: Przykładowa triangulacja wielokąta

Triangulacją wielokąta nazywamy zbiór tworzący ten wielokąt z trójkątów z niepustymi oraz rozłącznymi wnętrzami, gdzie ich przecięcie jest puste lub zredukowane do punktu albo do krawędzi. Na ogół, dla danego wielokąta istnieje wiele różnych triangulacji. Zajmiemy się triangulacją wyłącznie wielokątów monotonicznych, jednym ze sposobów triangulacji niemonotonicznych wielokątów jest podzielenie go na mniejsze, monotoniczne wielokąty.

3.3.2. Algorytm triangulacji

Pierwszym krokiem jest uporządkowanie punktów wielokątu względem monotoniczności, w tym określenie należności punktu do lewego lub prawego łańcucha. Następnie wkładamy dwa pierwsze punkty na stos i wywołujemy pętlę iterującą po naszej uporządkowanej liście punktów:

Jeżeli wybrany punkt nie należy do tego samego łańcucha co punkt na szczycie stosu to należy go połączyć ze wszystkimi wierzchołkami na stosie, a następnie zostawić na stosie dany punkt oraz jego szczyt.

W przeciwnym wypadku, analizujemy kolejne trójkąty które tworzy wybrany punkt ze szczytem stosu oraz następującymi jego elementami, jeżeli trójkąt należy do wielokąta to dodajemy przekątną i usuwamy „zewnętrzny" punkt. Innymi słowy, usuwamy tyle ile możemy „zewnętrznych" trójkątów, aby uzyskać mniejszy wielokąt.

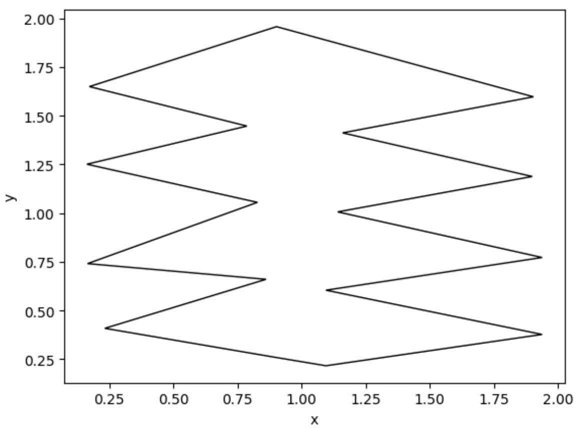
4. Realizacja obliczeń

Zaimplementowana została weryfikacja y-monotoniczności wielokąta. Sekcja 3.1 opisuje wybrany algorytm. Do klasyfikacji wierzchołków została użyta funkcja `orient` licząca wyznacznik 3x3. Sekcja 3.3.2 ukazuje postępowanie użytego algorytmu triangulacji. Wielokąty są przechowywane jako lista wierzchołków, gdzie i-ty tworzy bok z $(i+1)$-tym, oraz pierwszy i ostatni.

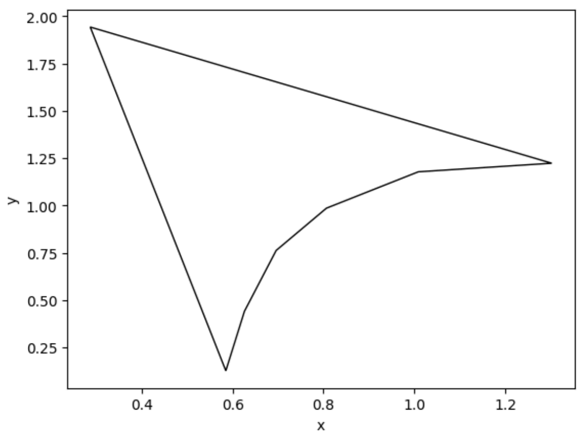
Użytkownik jest w stanie wprowadzić wielokąt za pomocą myszki wybierając kolejne punkty w kierunku przeciwnym do wskazówek zegara lub wczytać go z pliku. Przewidziane jest też zapisanie wielokątu do pliku oraz jego triangulacji w postaci boków i przekątnych wielokąta określanych przez listę par indeksów oznaczających numer wierzchołka. Istnieje możliwość wyświetlenia animacji triangulacji do pliku `.gif`

Do analizy zostały przyjęte następujące wielokąty:

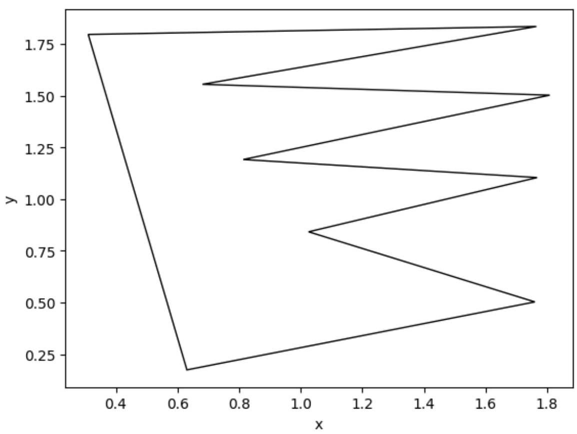
- **Choinka** - wielokąt z kątami ostrymi lub rozwartymi, na przemian łańcuchy
- **Grot** - wielokąt w którym podążamy tylko prawym łańcuchem z początku
- **Grzebień** - wielokąt z kolcami na prawym łańcuchu
- **Gwiazda** - niesymetryczny wielokąt przypominający gwiazdę
- **Okrąg** - wielokąt zbliżony kształtem do okręgu
- **Sześciokąt** - wielokąt będący sześciokątem
- **Krab** - niemonotoniczny wielokąt



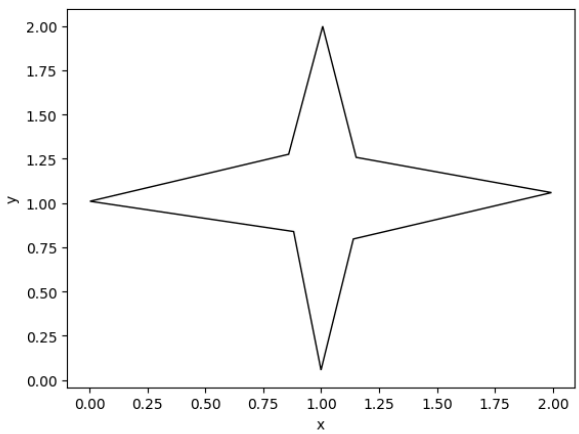
Rysunek 4: Wielokąt *choinka*



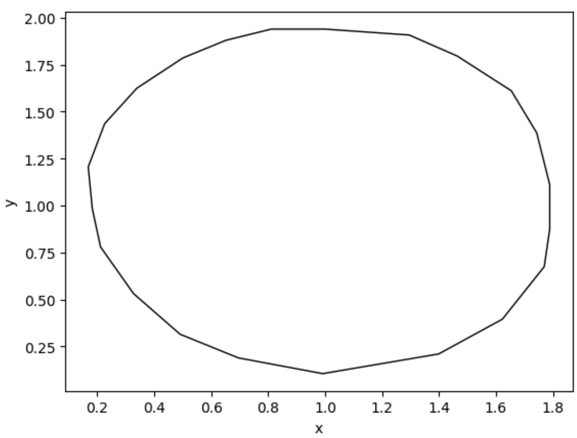
Rysunek 5: Wielokąt *grot*



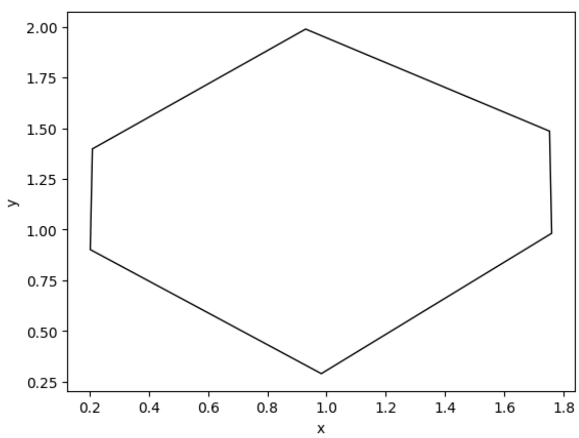
Rysunek 6: Wielokąt *grzebień*



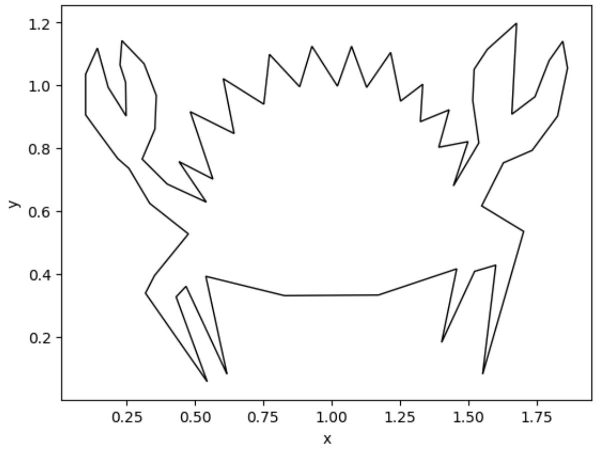
Rysunek 7: Wielokąt *gwiazda*



Rysunek 8: Wielokąt *okrąg*



Rysunek 9: Wielokąt *sześciokąt*



Rysunek 10: Wielokąt *krab*

5. Analiza obliczeń

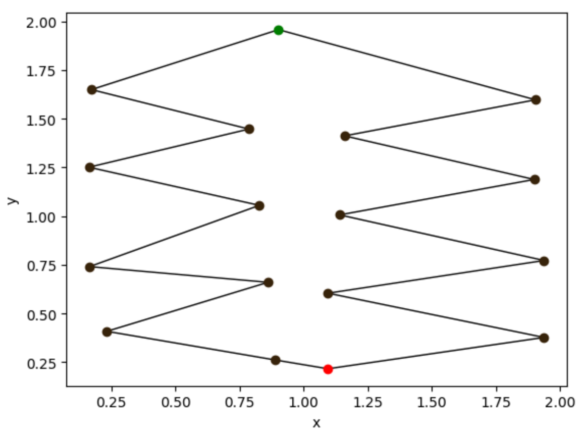
5.1. Sprawdzenie y-monotoniczności

Nazwa wielokąta	**Wynik funkcji `is_y_monotonic`**
Choinka	`true`
Grot	`true`
Grzbień	`true`
Gwiazda	`true`
Okrąg	`true`
Sześciokąt	`true`
Krab	`false`

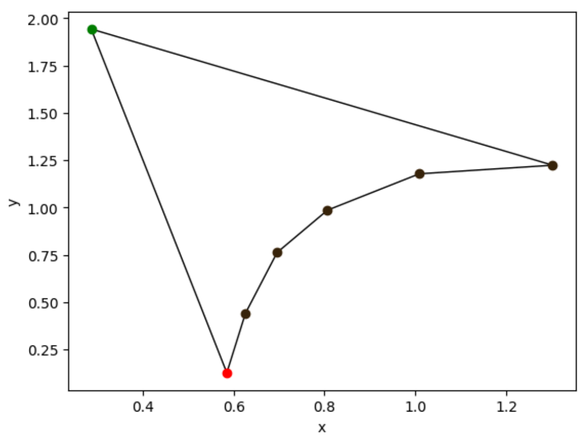
Tabela 1: Wyniki funkcji `is_y_monotonic` dla danych wielokątów

Zatem funkcja poprawnie weryfikuje monotoniczność wielokątów, ponieważ dla wielokąta *krab* oczekiwaliśmy wartości `false`, a dla reszty `true`.

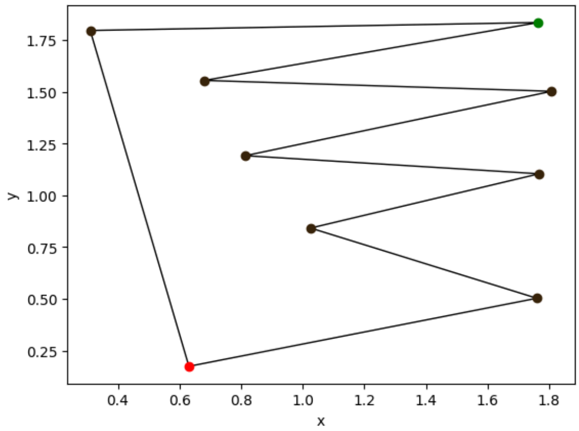
5.2. Klasyfikacja wierzchołków



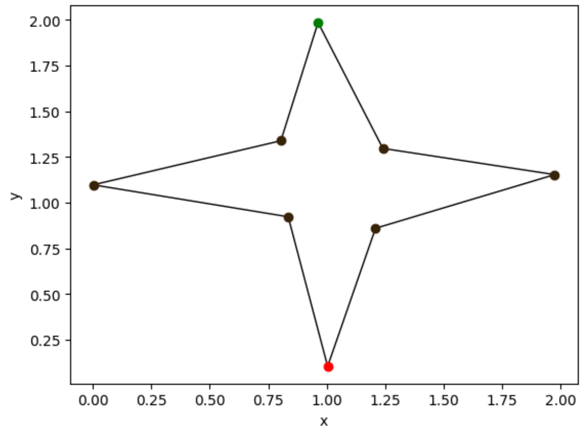
Rysunek 11: Klasyfikacja wierzchołków wielokątu *choinka*



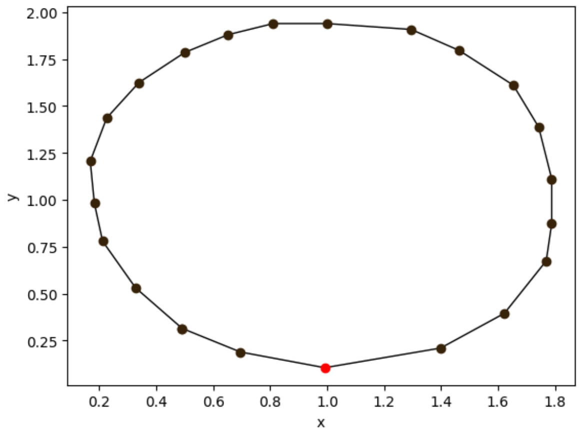
Rysunek 12: Klasyfikacja wierzchołków wielokątu *grot*



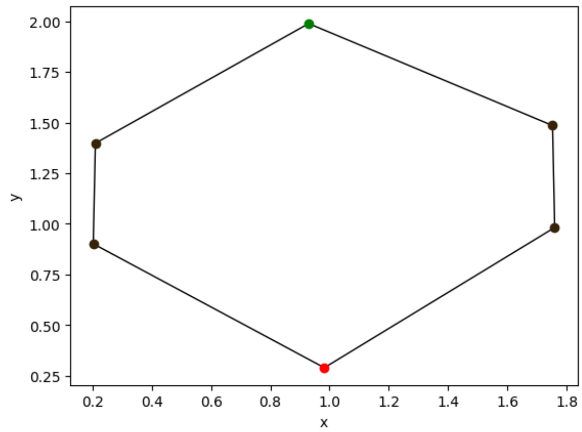
Rysunek 13: Klasyfikacja wierzchołków wielokątu *grzebień*



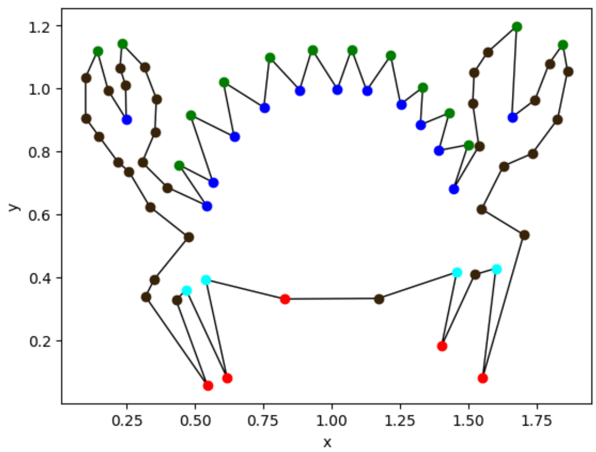
Rysunek 14: Klasyfikacja wierzchołków wielokątu *gwiazda*



Rysunek 15: Klasyfikacja wierzchołków wielokątu *okrąg*



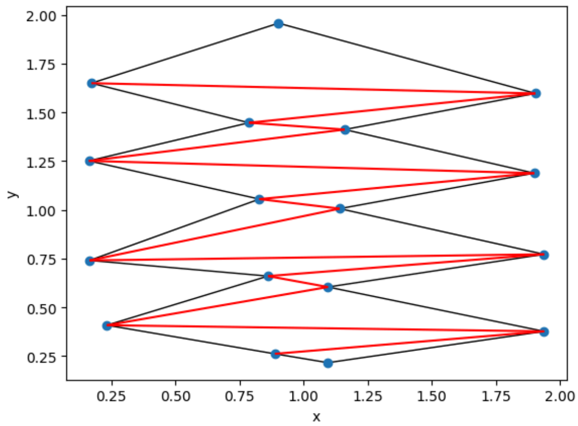
Rysunek 16: Klasyfikacja wierzchołków wielokątu *sześciokąt*



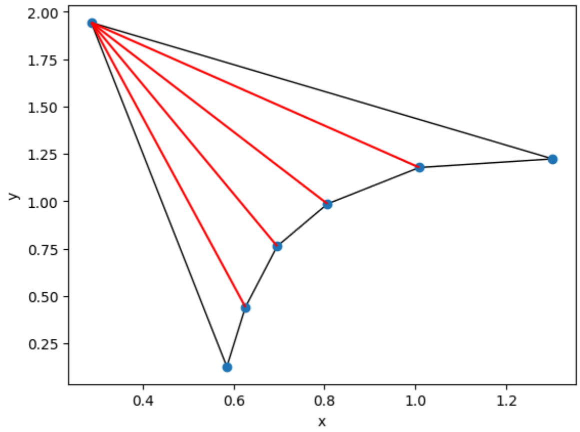
Rysunek 17: Klasyfikacja wierzchołków wielokątu *krab*

Możemy zauważyć, że wielokąty y-monotoniczne nie posiadają żadnych wierzchołków łączących i dzielących. Rysunek 15 przedstawia wielokąt *okrąg* gdzie nie ma żadnych wierzchołków początkowych. Wynika to z dobranej tolerancji dla zera. Nie ma to wpływu na jego monotoniczność.

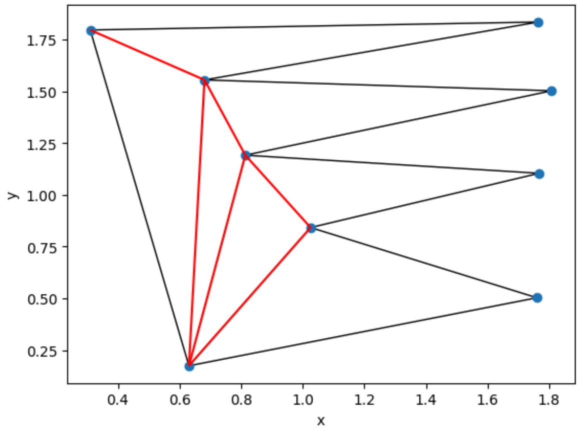
5.3. Triangulacja



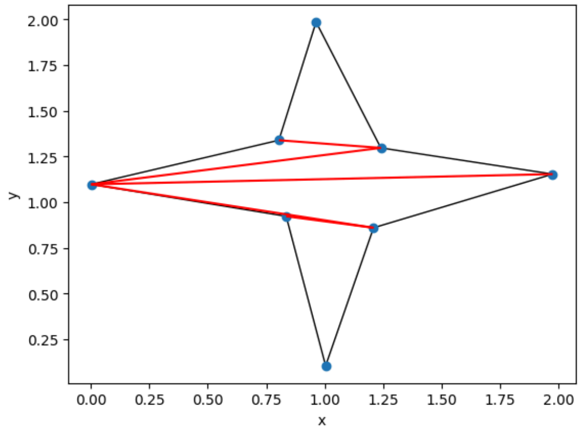
Rysunek 18: Triangulacja wielokąta *choinka*



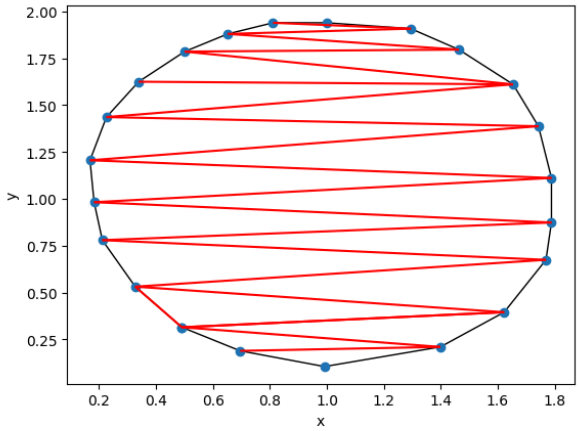
Rysunek 19: Triangulacja wielokąta *grot*



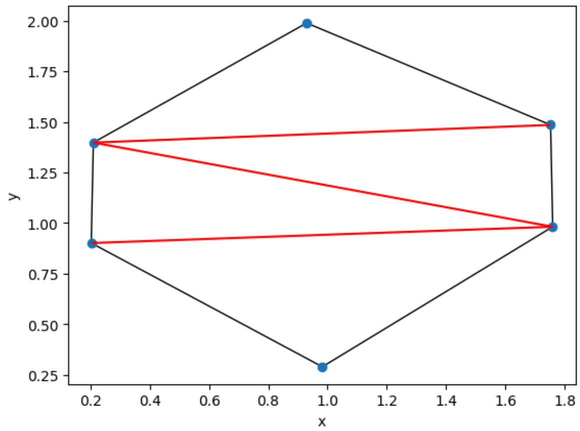
Rysunek 20: Triangulacja wielokąta *grzebień*



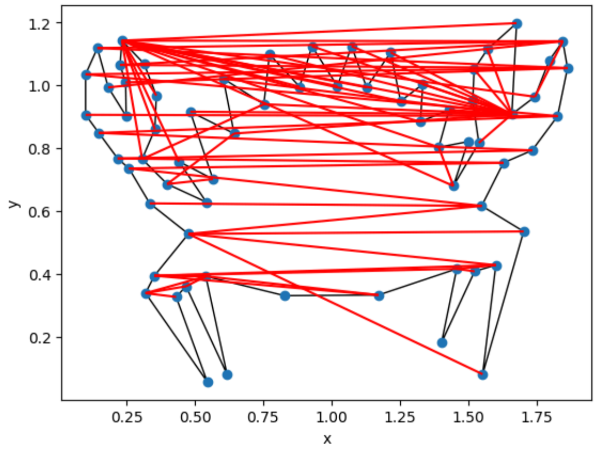
Rysunek 21: Triangulacja wielokąta *gwiazda*



Rysunek 22: Triangulacja wielokąta *okrąg*



Rysunek 23: Triangulacja wielokąta *sześciokąt*



Rysunek 24: Triangulacja wielokąta *krab*

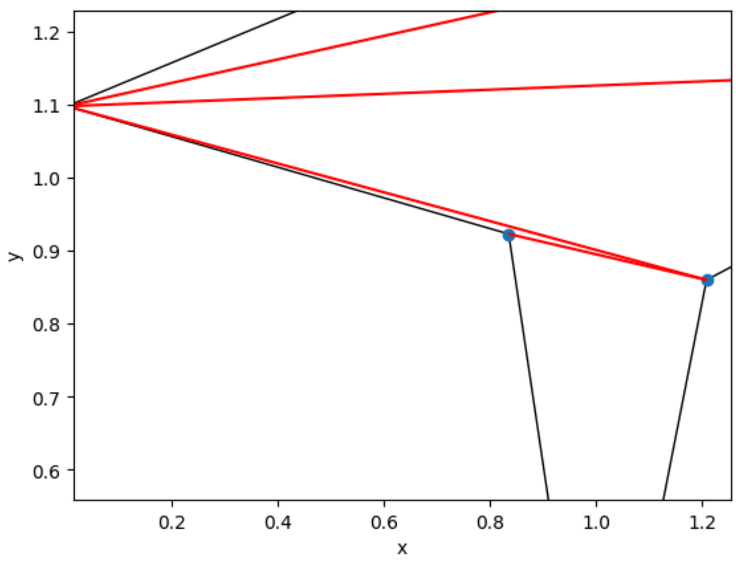
Wykresy triangulacji wielokątów powyżej potwierdzają poprawność zaimplementowanego algorytmu. Dodatkowo został on uruchomiony na wielokącie *krab*, który nie spełnia założenia y-monotoniczności.

Rysunek 18 pokazuje jak algorytm poradził sobie w przypadku punktów z lewego i prawego łańcucha na zmianę ustawionych względem monotoniczności oraz wielu kątów ostrych i rozwartych.

Wielokąt *grot* ukazuje przypadek gdzie stos stanie się najdłuższy ze względu na ustawienie prawie wszystkich punktów na prawym łańcuchu oraz wspólnej niemożliwości stworzenia trójkątów z wierzchołków na stosie, ponieważ nie zawierałyby się one we wnętrzu wielokąta

Rysunek 20 przedstawia możliwość powstania trójkąta w triangulacji którego bokami są wyłącznie przekątne.

Rysunek 21 z pozoru przedstawia błędną triangulację, po przybliżeniu wykresu można zauważyć poprawność triangulacji, ukazuje to rysunek 25.



Rysunek 25: Przybliżony wykres triangulacji wielokąta *gwiazda*

Dla wielokątów *okrąg* i *sześciokąt* triangulacja została przeprowadzona pomyślnie.

Rysunek 24 przedstawia nieudaną triangulację wielokąta niemonotonicznego.

6. Wnioski

Zaimplementowany algorytm weryfikowania y-monotoniczności poprawnie zadziałał dla wszystkich wielokątów. Klasyfikacja wierzchołków przebiegła pomyślnie dla każdego przypadku. Algorytm triangulacji poradził sobie ze wszystkimi zadanymi wielokątami monotonicznymi. Ze sprawozdaniem zostało dołączone archiwum z animacjami triangulacji w formacie `.gif`