

Algorytmy Geometryczne,

Ćwiczenie 3 – Triangulacja Wielokątów Monotonicznych

Maciej Kmak

Listopad 2024

1 Cel ćwiczenia

Celem ćwiczenia było zapoznanie się z zagadnieniem triangulacji wielokątów monotonicznych oraz praktyczną implementacją i analizą algorytmów związanych z tym tematem. Szczególną uwagę poświęcono algorytmowi sprawdzania, czy dany wielokąt jest monotoniczny względem osi OY , klasyfikacji wierzchołków w dowolnym wielokącie oraz algorytmowi triangulacji wielokąta y -monotonicznego. Ćwiczenie miało na celu umożliwienie lepszego zrozumienia teoretycznych podstaw tych algorytmów oraz przeprowadzenie eksperymentów i przeanalizowaniu wyników na różnych zestawach danych.

2 Dane techniczne

Program (w pliku „kmak_kod_3.ipynb”) jest napisany w języku Python w środowisku Jupyter Notebook. Wykresy punktów i wizualizacje w formacie GIF powstały dzięki narzędziu przygotowanemu przez koło naukowe Bit, które korzysta z biblioteki matplotlib. Narzędzia dostępne w bibliotece matplotlib i tkinter umożliwiły zadawanie wielokątów przy pomocy myszki. Przedstawiony program został wyegzekwowany na komputerze z procesorem Intel® Core™ i5-1235U z systemem operacyjnym Windows 11 Pro.

3 Wstęp teoretyczny

3.1 Y -monotoniczność wielokąta

Wielokąt prosty nazywamy ściśle monotonicznym względem prostej l (wyznaczającej kierunek monotoniczności), gdy:

1. Jego brzeg można podzielić na dwa spójne łańcuchy:
 - o „górnym” łańcuch, na którym punkty łańcucha układają się w sposób zgodny z rosnącym porządkiem względem kierunku l ,
 - o „dolnym” łańcuch, na którym punkty układają się w sposób zgodny z malejącym porządkiem względem kierunku l .
2. Każda pozioma linia prosta l' (prostopadła do l) przecina wielokąt w co najwyżej dwóch odcinkach.

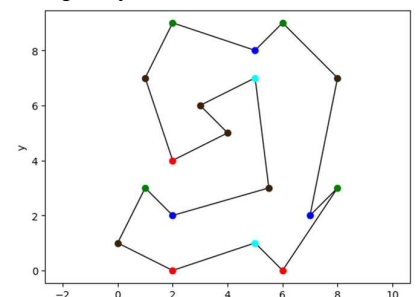
W szczególności, jeśli kierunek prostej l pokrywa się z osią OY , to wielokąt taki nazywamy **y -monotonicznym**. W przypadku y -monotoniczności każda pozioma linia prosta ($y = \text{const}$) przecina wielokąt w co najwyżej dwóch odcinkach – jednym na lewym łańcuchu i jednym na prawym.

3.2 Podział wierzchołków wielokąta na kategorie.

Wierzchołki naszego wielokąta możemy przypisać do kategorii:

- **początkowe**, gdy obaj jego sąsiedzi leżą poniżej i kąt wewnętrzny ma mniej niż 180°
- **końcowe**, gdy obaj jego sąsiedzi leżą powyżej i kąt wewnętrzny ma mniej niż 180°
- **dzielący**, gdy obaj jego sąsiedzi leżą poniżej i kąt wewnętrzny ma więcej niż 180°
- **łączący**, gdy obaj jego sąsiedzi leżą powyżej i kąt wewnętrzny ma więcej niż 180°
- **prawidłowy** – pozostałe przypadki (jeden sąsiad poniżej, drugi-powyżej)

Na rysunku 3.2.1 zostało przedstawione przykładowe kolorowanie wierzchołków. Przyjęta konwencja kolorowania wierzchołków odpowiada kolorom użytym w opisie nazw poszczególnych typów wierzchołków.



Rysunek 3.2.1 Przykładowe kolorowanie wierzchołków

3.3 Definicja triangulacji

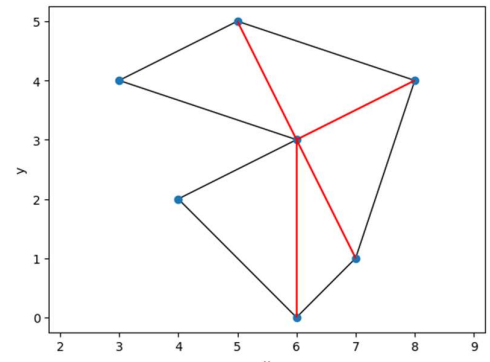
Niech:

K – określa sympleks w danej przestrzeni

Definicja:

T nazywamy triangulacją Ω , jeżeli spełnione są następujące warunki:

- $\Omega = \bigcup_{K \in \Omega} K$
- Każdy element K ma niepuste wnętrze,
- Wnętrza różnych elementów są rozłączne,
- Przecięcie dwóch różnych elementów jest:
 - albo puste
 - albo zredukowane do jednego punktu, który jest wspólnym wierzchołkiem tych elementów
 - albo zredukowane do jednej krawędzi, która jest wspólną krawędzią tych elementów
 - albo zredukowane do jednej ściany (w 3D), która jest wspólną ścianą tych elementów.



Rysunek 3.3.1 Przykładowa triangulacja wielokąta

Triangulacją wielokąta w 2D (Przykład takiej triangulacji na rysunku 3.3.1) nazywamy podział wielokąta na trójkąty, których wierzchołkami są wierzchołki wielokąta. W ćwiczeniu skupiamy się wyłącznie na triangulacji wielokątów y -monotonicznych, co upraszcza realizację tego procesu.

3.4 Opis algorytmu sprawdzającego y -monotoniczność wielokąta

1. Wyznaczamy wierzchołki o minimalnej i maksymalnej współrzędnej y .
2. Dzielimy wielokąt na lewy i prawy łańcuch względem tych wierzchołków.
3. Sprawdzamy, czy współrzędne y wierzchołków każdego łańcucha są monotoniczne.
4. Jeśli oba łańcuchy spełniają warunek monotoniczności, wielokąt jest y -monotoniczny.

3.5 Opis algorytmu klasyfikujący wierzchołki wielokąta

Algorytm klasyfikuje wierzchołki wielokąta na pięć kategorii: początkowe, końcowe, dzielące, łączące i prawidłowe, w zależności od położenia sąsiadów względem wierzchołka oraz kąta wewnętrznego. Kąt ten jest oceniany przy użyciu wyznacznika macierzy. Na podstawie tych informacji algorytm przypisuje każdemu wierzchołkowi odpowiednią kategorię, co umożliwia dalsze operacje, takie jak triangulacja.

3.6 Opis algorytmu wykonujący triangulację wielokątów monotonicznych

Algorytm dokonuje triangulacji wielokąta monotonicznego. Działanie algorytmu można opisać w następujących krokach:

- Określamy lewy i prawy łańcuch oraz porządkujemy wierzchołki wzdłuż kierunku monotoniczności.
- Wkładamy dwa pierwsze wierzchołki na stos.
- Jeśli kolejny wierzchołek pochodzi z innego łańcucha, łączymy go z wszystkimi wierzchołkami na stosie, pozostawiając dwa ostatnie.
- Jeśli należy do tego samego łańcucha, analizujemy trójkąty tworzone z wierzchołkami ze stosu:
 - Dodajemy przekątną i usuwamy wierzchołek, jeśli trójkąt należy do wielokąta,
 - W przeciwnym razie zachowujemy wierzchołki na stosie.

Algorytm, w zależności od potrzeb i implementacji może zwracać dane m.in. w jednej z dwóch form:

- Tablica krotek z indeksami trzech wierzchołków definiujących trójkąty triangulacji (zgodnie z orientacją przeciwną do ruchu wskazówek zegara - CCW).
- Tablica krotek z indeksami dwóch wierzchołków reprezentujących dodawane przekątne.

Obie formy zostały zaimplementowane w kodzie (kma_kod_3.ipynb) z programem. Trójki wierzchołków służą nam do wypisania wynikowej triangulacji, krotki przekątnych są natomiast wygodniejsze w wizualizacji GIF.

4 Zbiory i przebieg doświadczenia

4.1 Przebieg doświadczenia

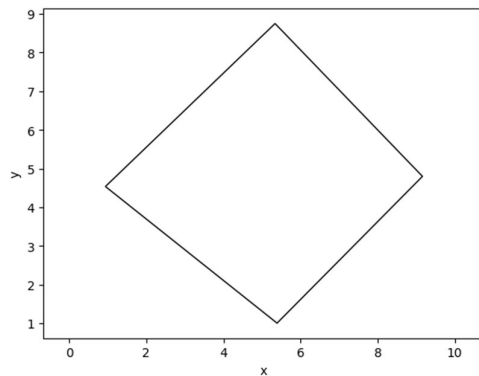
W ramach doświadczenia zaimplementowano algorytmy sprawdzania y-monotoniczności wielokąta, klasyfikacji wierzchołków oraz triangulacji wielokątów y-monotonicznych, zgodnie z przedstawionymi implementacjami.

Sprawdzanie y-monotoniczności polegało na podziale wierzchołków wielokąta na dwa łańcuchy – lewy i prawy – względem wierzchołków o minimalnej i maksymalnej współrzędnej y . Następnie dla każdego łańcucha weryfikowano monotoniczność współrzędnych y . Poprawność algorytmu została przetestowana na różnych wielokątach, w tym także niewypukłych.

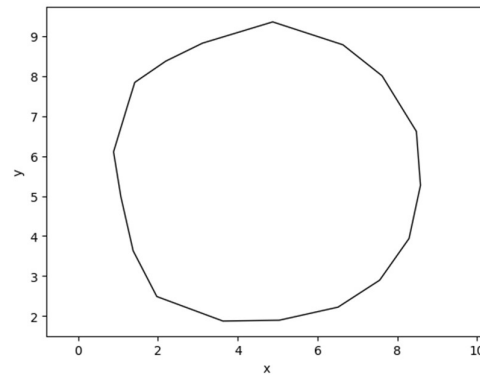
W algorytmie klasyfikacji wierzchołków wielokąta zastosowano reguły geometryczne określające typ wierzchołka na podstawie pozycji sąsiednich punktów i kąta wewnętrznego, obliczanego przy użyciu funkcji liczącej wyznacznik `orient()`. Wierzchołki zostały zaklasyfikowane do pięciu kategorii: początkowych, końcowych, łączących, dzielących i prawidłowych. Wyniki klasyfikacji były wizualizowane przez odpowiednie kolorowanie wierzchołków.

Algorytm triangulacji wielokąta monotonicznego dzieli wierzchołki na lewy i prawy łańcuch, a następnie sortuje liniowo według współrzędnej y . Wierzchołki są przetwarzane w kolejności malejących współrzędnych y , a stos zarządza bieżącymi wierzchołkami triangulacji. Dla wierzchołków z różnych łańcuchów stos jest opróżniany, generując trójkąty, natomiast dla wierzchołków z tego samego łańcucha funkcja `correct_triangle()` z pomocą funkcji `orient()` sprawdza, czy trójkąty są prawidłowe dla triangulacji, analizując orientację punktów. Na koniec brakujące trójkąty są tworzone przez opróżnienie stosu, co zapewnia pełną triangulację.

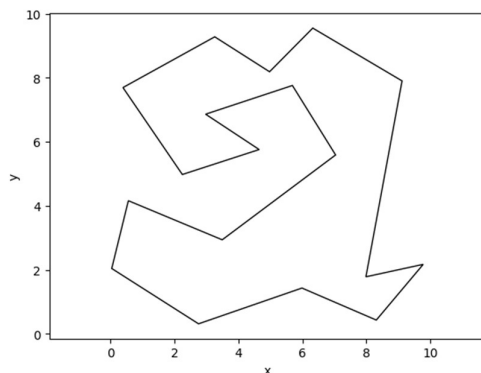
4.2 Testowane zbiory danych



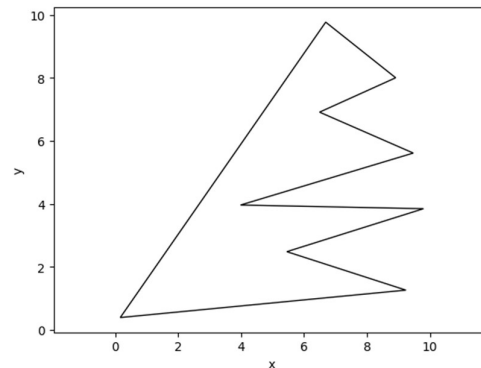
Wykres 4.2.1 Wielokąt „Czworokąt”



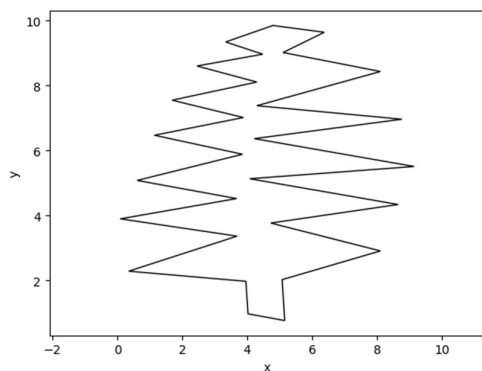
Wykres 4.2.2 Wielokąt „Okrąg”



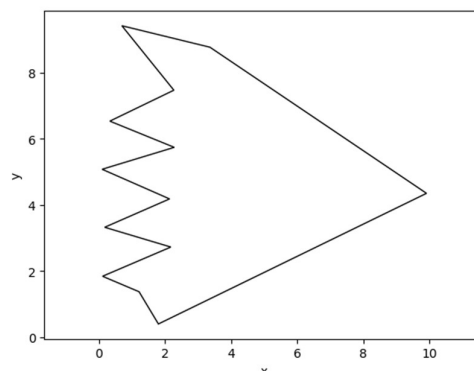
Wykres 4.2.3 Wielokąt „Figura niemonotoniczna zaproponowana na wykładzie”



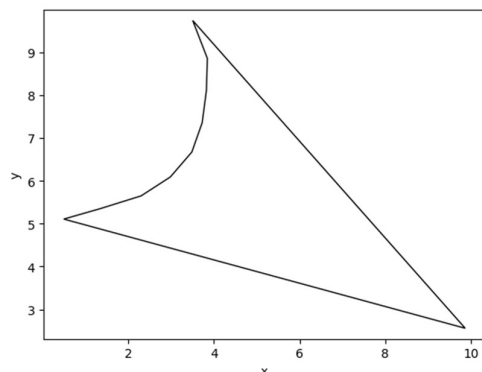
Wykres 4.2.4 Wielokąt „Grzebień”



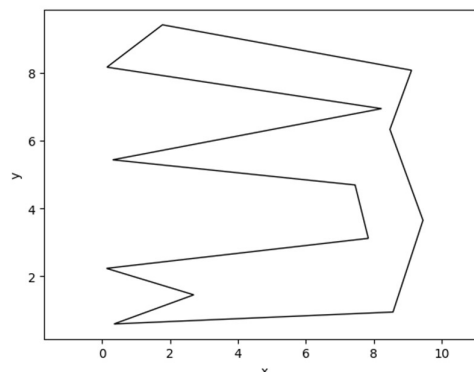
Wykres 4.2.5 Wielokąt „Choinka”



Wykres 4.2.6 Wielokąt „Lotka”



Wykres 4.2.7 Wielokąt „Kursor”



Wykres 4.2.8 Wielokąt „Wklęsły”

Figury takie jak „czworokąt” i „okrąg” (zaprezentowane na wykresach 4.2.1-4.2.2) przedstawiają podstawowe przypadki figur wypukłych, które stanowią prosty test dla algorytmu triangulacji. Z kolei „figura niemonotoniczna zaproponowana na wykładzie” (przedstawiona na wykresie 4.2.3) nie spełnia warunku y -monotoniczności. Pozostałe figury (zaprezentowane na wykresach 4.2.4-4.2.8) nie są wypukłe, co pozwala na sprawdzenie poprawności zarówno samej triangulacji, jak i klasyfikacji trójkątów wewnątrz figury. Testowane są także przypadki różnej proporcji długości gałęzi (lewego i prawego łańcucha) oraz nietypowego położenia punktów „ekstremalnych”. Dzięki temu weryfikowana jest niezależność działania algorytmu od symetrii i rozkładu wierzchołków w analizowanym wielokącie.

5 Wyniki doświadczenia

Dla wielokątów zaprezentowanych na wykresach 4.2.1-4.2.8 sprawdzono wyniki działania algorytmów omówionych w punktach 3.4-3.6. W następnych punktach przedstawię wyniki działania zaimplementowanych algorytmów.

5.1 Wyniki algorytmu sprawdzającego y -monotoniczność wielokąta

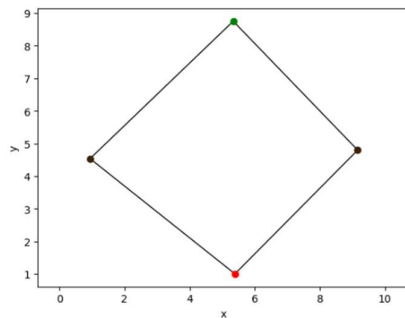
Wyniki działania przedstawiono w tabeli (Tabela 5.1) poniżej. Dla wszystkich figur przedstawionych na wykresach 4.2.1-4.2.8, z wyjątkiem figury z wykresu 4.2.3, algorytm zwrócił wartość `True`, co wskazuje na ich y -monotoniczność. Natomiast dla figury z wykresu 4.2.3 algorytm zwrócił `False`, co było zgodne z oczekiwaniami, ponieważ figura ta została zadana jako niespełniająca warunku y -monotoniczności. W związku z tym nie będziemy rozważać jej w przypadku algorytmu triangulacji.

Tabela 1. Wyniki działania programu sprawdzającego y -monotoniczność wielokąta

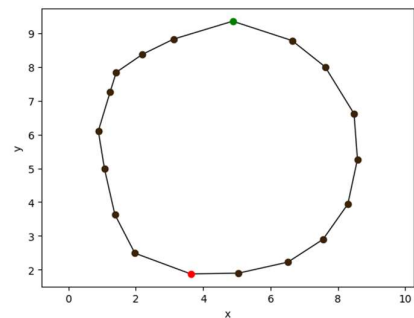
Figura 4.2.1	Figura 4.2.2	Figura 4.2.3	Figura 4.2.4	Figura 4.2.5	Figura 4.2.6	Figura 4.2.7	Figura 4.2.8
True	True	False	True	True	True	True	True

5.2 Wyniki algorytmu klasyfikujący wierzchołki wielokąta

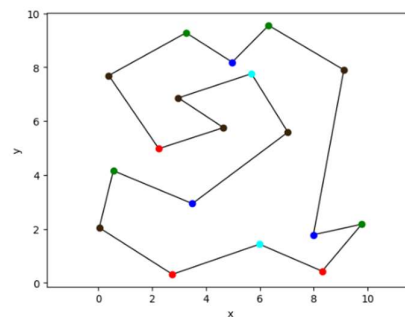
Poniżej na wykresach (Wykresy 5.2.1-5.2.8) możemy zobaczyć wyniki działania algorytmu klasyfikacji wierzchołków wielokąta dla poszczególnych zbiorów (widocznych na wykresach 4.2.1-4.2.8). Przyjęto te same oznaczenia jak w punkcie „3.2 Podział wierzchołków wielokąta na kategorie.”



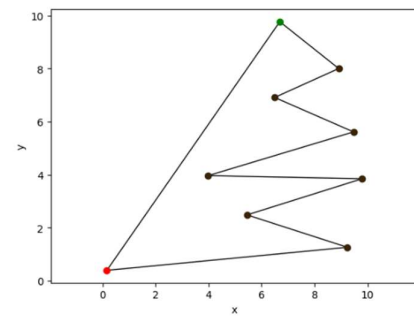
Wykres 5.2.1 Klasyfikacja punktów „Czworokąt”



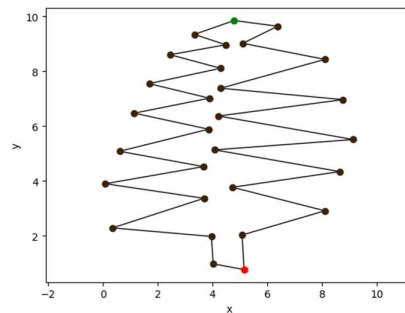
Wykres 5.2.2 Klasyfikacja punktów „Okrag”



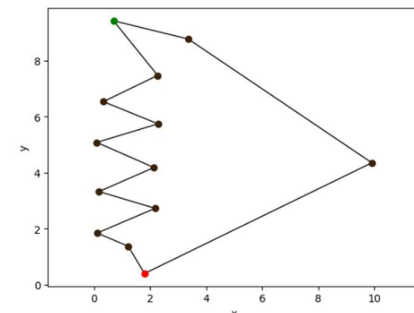
Wykres 5.2.3 Klasyfikacja punktów „Figura niemonotoniczna zaproponowana na wykładzie”



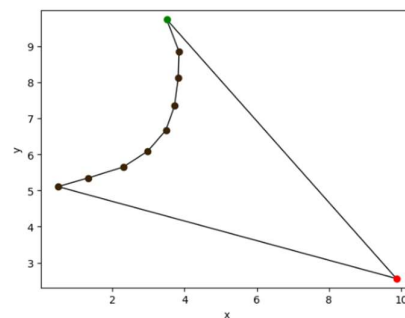
Wykres 5.2.4 Klasyfikacja punktów „Grzebień”



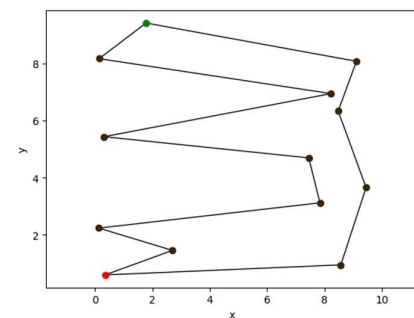
Wykres 5.2.5 Klasyfikacja punktów „Choinka”



Wykres 5.2.6 Klasyfikacja punktów „Lotka”



Wykres 5.2.7 Klasyfikacja punktów „Kursor”



Wykres 5.2.8 Klasyfikacja punktów „Wklęśły”

Na podstawie przedstawionych wyników działania algorytmu klasyfikacji wierzchołka (wykresy 5.2.1-5.2.8) można wyciągnąć następujące wnioski:

1. Poprawność klasyfikacji dla figur wypukłych:

- Na wykresach 5.2.1 („Czworokąt”) i 5.2.2 („Okrąg”) algorytm poprawnie sklasyfikował wierzchołki jako początkowe, końcowe oraz prawidłowe, co jest zgodne z monotonicznością tych figur. Klasyfikacja nie wymagała identyfikacji bardziej złożonych typów wierzchołków, ponieważ figury te są proste i wypukłe.

2. Klasyfikacja dla figury niemonotonicznej (Wykres 5.2.3):

- Dla „figury niemonotonicznej zaproponowanej na wykładzie” (Wykres 5.2.3) algorytm poprawnie zidentyfikował dodatkowe typy wierzchołków: dzielące i łączące. Są one zgodne z definicją klasyfikacji wierzchołków i wynikają z braku y -monotoniczności figury. Klasyfikacja ta podkreśla złożoność struktury figury, jednocześnie potwierdzając poprawność działania algorytmu.

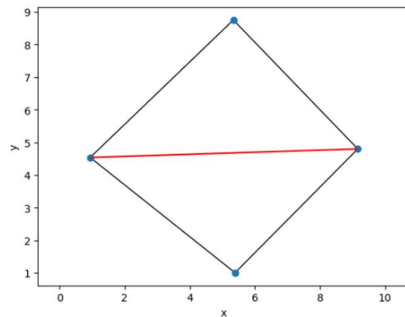
3. Poprawność klasyfikacji dla figur niewypukłych:

- Na wykresach 5.2.4-5.2.8 (figury niewypukłe: „Grzebień”, „Chojinka”, „Lotka”, „Kursor”, „Wklęsły”) algorytm poprawnie sklasyfikował wierzchołki jako początkowe, końcowe oraz prawidłowe. Brak identyfikacji typów dzielących i łączących potwierdza, że figury te są y -monotoniczne, mimo obecności wklęsłości i bardziej złożonej geometrii.

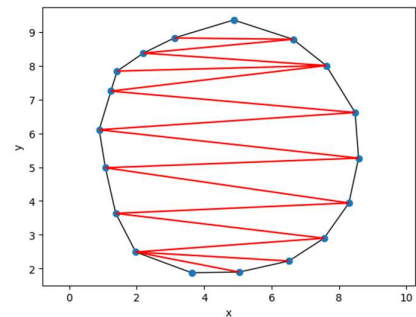
Algorytm klasyfikacji wierzchołków działa poprawnie zarówno dla figur wypukłych, jak i niewypukłych. Dla figury niemonotonicznej (Wykres 5.2.3) algorytm prawidłowo sklasyfikował niektóre wierzchołki jako dzielące i łączące, co wynika z braku y -monotoniczności i zgodne jest z definicjami tych kategorii. Dla pozostałych figur klasyfikacja ograniczyła się do wierzchołków początkowych, końcowych i prawidłowych, co potwierdza, że algorytm skutecznie identyfikuje cechy geometryczne wierzchołków każdego typu.

5.3 Wyniki algorytmu wykonujący triangulację wielokątów monotonicznych

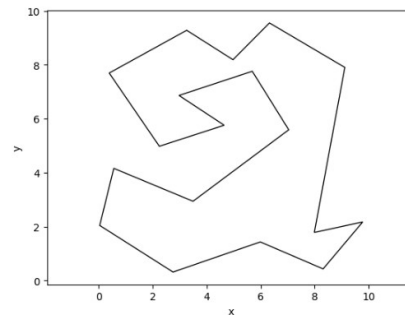
Poniżej na wykresach (Wykresy 5.3.1-5.3.8) możemy zobaczyć wyniki działania algorytmu triangulacji dla poszczególnych zbiorów (widocznych na wykresach 4.2.1-4.2.8). Triangulacja wielokąt z wykresu 4.2.3 została pominięta ponieważ wielokąt ten nie jest y-monotoniczny. Dla spójności oznaczeń wykres 5.3.3 zawiera jedynie wielokąt bez triangulacji.



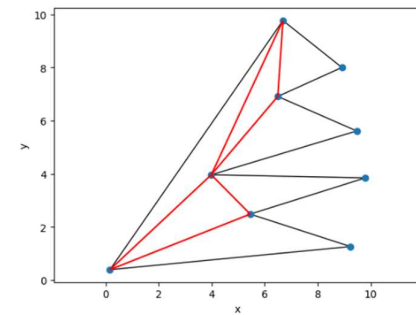
Wykres 5.3.1 Triangulacja „Czworokąt”



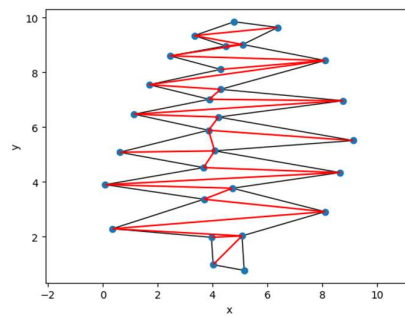
Wykres 5.3.2 Triangulacja „Okrąg”



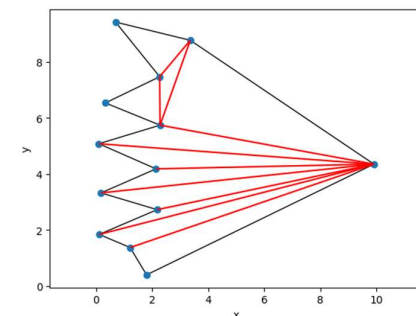
Wykres 5.3.3 Wielokąt bez triangulacji
„Figura niemonotoniczna zaproponowana na wykładzie”



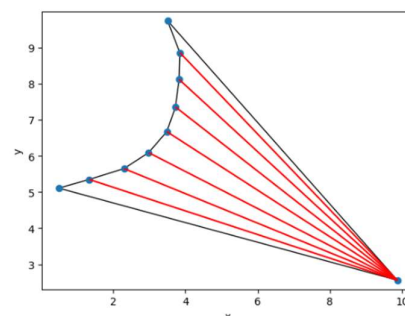
Wykres 5.3.4 Triangulacja „Grzebień”



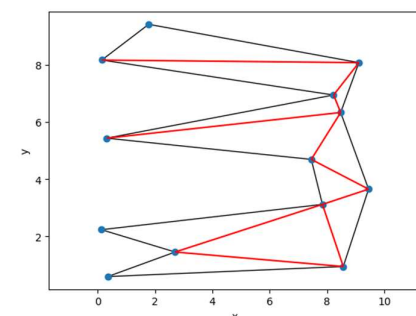
Wykres 5.3.5 Wielokąt „Choinka”



Wykres 5.3.6 Wielokąt „Lotka”



Wykres 5.3.7 Wielokąt „Kursor”



Wykres 5.3.8 Wielokąt „Wklęsły”

Na podstawie przedstawionych wyników działania algorytmu triangulacji (wykresy 5.3.1-5.3.8) można wyciągnąć następujące wnioski:

1. Poprawność triangulacji dla figur wypukłych:

- Wyniki na wykresach 5.3.1 („Czworokąt”) i 5.3.2 („Okrąg”) potwierdzają, że algorytm działa prawidłowo w przypadku prostych, wypukłych wielokątów. Figury te zostały poprawnie podzielone na trójkąty, zachowując zasadę, że wszystkie wierzchołki są wykorzystywane w triangulacji, a utworzone trójkąty pokrywają całą powierzchnię figury.

2. Niemonotoniczność figury z wykresu 4.2.3:

- Figura „niemonotoniczna” (wykres 4.2.3) została wykluczona z triangulacji, ponieważ nie spełniała warunku y-monotoniczności, co było zgodne z założeniami algorytmu.

3. Poprawność triangulacji dla figur niewypukłych:

- Wykresy 5.3.4-5.3.8 przedstawiają bardziej złożone figury niewypukłe, które testują zdolność algorytmu do generowania poprawnej triangulacji w trudniejszych przypadkach:
 - Na wykresie 5.3.4 („Grzebień”) algorytm poprawnie podzielił wielokąt na trójkąty, zachowując właściwe połączenia wierzchołków, mimo że figura miała znaczące „zagłębienia” i była wyraźnie asymetryczna.
 - Figura „Choinka” (wykres 5.3.5) zawierała wiele ostrych kątów i nierównomierne rozłożenie wierzchołków. Algorytm skutecznie poradził sobie z takim przypadkiem, generując spójny podział na trójkąty.
 - Na wykresie 5.3.6 („Lotka”) algorytm poradził sobie z nietypowym kształtem, poprawnie uwzględniając wszystkie wierzchołki i generując prawidłowe trójkąty.
 - Na wykresie 5.3.7 („Kursor”) algorytm poprawnie podzielił wielokąt na trójkąty, tworząc odpowiednie połączenia między wierzchołkami i zachowując spójność, mimo nietypowego kształtu figury z charakterystycznym pojedynczym punktem ekstremalnym.
 - Figura „Wklęsły” (wykres 5.3.8) została poprawnie podzielona na trójkąty, mimo obecności licznych wklęsłości i nieregularnego rozkładu wierzchołków, co potwierdza skuteczność algorytmu w obsłudze skomplikowanych niejednorodnych kształtów.

Dla prezentowanych zbiorów (4.2 Testowane zbiory danych) wygenerowano pliki .gif, które wizualizują proces powstawania triangulacji dla każdego wielokąta. Pliki te ukazują kolejne kroki algorytmu, od sortowania wierzchołków po sukcesywne dodawanie przekątnych i tworzenie trójkątów, co pozwala lepiej zrozumieć działanie i dynamikę algorytmu. Ze względu na ograniczenia formatu .pdf, pliki .gif nie zostały załączone w sprawozdaniu, ale są dostępne w folderze kmap_gifs.zip przesłanym razem ze sprawozdaniem. W tym samym folderze znajduje się także legenda do GIFów, która objaśnia oznaczenia użyte w wizualizacjach.

6 Wnioski

Na podstawie przeprowadzonych testów i analizy działania algorytmów można sformułować następujące wnioski:

- **Poprawność algorytmu sprawdzającego y -monotoniczność wielokąta:** Algorytm skutecznie zidentyfikował wszystkie y -monotoniczne wielokąty spośród testowanych zbiorów. Jedynym wyjątkiem była figura niemonotoniczna (Wykres 4.2.3), dla której algorytm zwrócił wartość False, co było zgodne z oczekiwaniami i założeniami teoretycznymi.
- **Efektywność klasyfikacji wierzchołków:** Algorytm klasyfikujący poprawnie przypisał wierzchołki wielokątów do kategorii (początkowe, końcowe, dzielące, łączące i prawidłowe) w zależności od ich geometrii. Dla y -monotonicznych wielokątów (Wykresy 5.2.1-5.2.8, z wyjątkiem 5.2.3) klasyfikacja była zgodna z oczekiwaniami i ograniczała się do kategorii zgodnych z monotonicznością tj. początkowe, końcowe i prawidłowe. W przypadku figury niemonotonicznej (Wykres 5.2.3) algorytm prawidłowo rozpoznał dodatkowe typy wierzchołków (dzielące i łączące), co wynikało z braku y -monotoniczności.
- **Poprawność triangulacji wielokątów y -monotonicznych:** Algorytm triangulacji prawidłowo podzielił wszystkie y -monotoniczne wielokąty na trójkąty (Wykresy 5.3.1-5.3.8, z wyjątkiem 5.3.3). Figury wypukłe („Czworokąt” i „Okrąg”) oraz niewypukłe (np. „Grzebień”, „Chojinka”, „Lotka”) zostały poprawnie ztriangulowane, niezależnie od ich kształtu, proporcji łańcuchów czy liczby wklęsłości. Figura niemonotoniczna została słusznie pominięta w triangulacji, co było zgodne z założeniami algorytmu.

7 Podsumowanie zagadnienia

Przeprowadzone testy potwierdziły poprawność i efektywność zaimplementowanych algorytmów sprawdzających y -monotoniczność wielokąta, klasyfikujących wierzchołki oraz dokonujących triangulacji wielokątów y -monotonicznych.

- **Algorytm sprawdzania y -monotoniczności** okazał się niezawodny w identyfikacji wielokątów monotonicznych i ich odróżnieniu od wielokątów niemonotonicznych, co zapewniło poprawną podstawę dla dalszych operacji.
- **Algorytm klasyfikacji wierzchołków** skutecznie przypisał wierzchołki do odpowiednich kategorii, umożliwiając precyzyjną analizę struktury geometrycznej wielokątów.
- **Algorytm triangulacji wielokątów monotonicznych** działał poprawnie zarówno dla figur wypukłych, jak i niewypukłych, niezależnie od ich złożoności geometrycznej, proporcji łańcuchów czy liczby wklęsłości.

Podsumowując, zaimplementowane algorytmy umożliwiły skuteczną realizację zadania triangulacji wielokątów y -monotonicznych, a ich wyniki były zgodne z założeniami teoretycznymi i przewidywaniami.