AKADEMIA GÓRNICZO-HUTNICZA
WYDZIAŁ INFORMATYKI, ELEKTRONIKI I TELEKOMUNIKACJI
KIERUNEK INFORMATYKA

ALGORYTMY GEOMETRYCZNE

Laboratorium 4

Przecinanie się odcinków

Kyrylo Iakymenko

Kraków, 5 grudnia 2023

1 Wprowadzenie

Zadaniem analizy przecinania się odcinków jest rozwiązanie fundamentalnego problemu w geometrii komputerowej, mającego szerokie zastosowanie w różnych dziedzinach, takich jak grafika komputerowa, analiza obrazów, planowanie tras czy symulacje fizyczne. Problem ten skupia się na identyfikacji punktów przecięcia się dwóch odcinków na płaszczyźnie, co stanowi kluczowy element w wielu algorytmach i systemach przetwarzania danych geometrycznych.

W dalszej części raportu omówimy algorymt zamiatania służący rozwiązania problemu przecinania się odcinków oraz przeanalizujemy złożoność czasową i przestrzenną algorymtu.

2 Krótki opis ćwiczenia

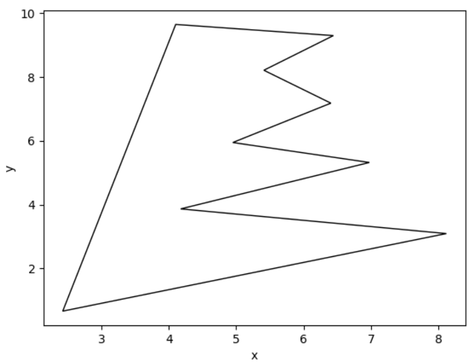
2.1 Wykorzystane algorytmy

W danym ćwiczeniu testujemy działanie następujących algorytmów:

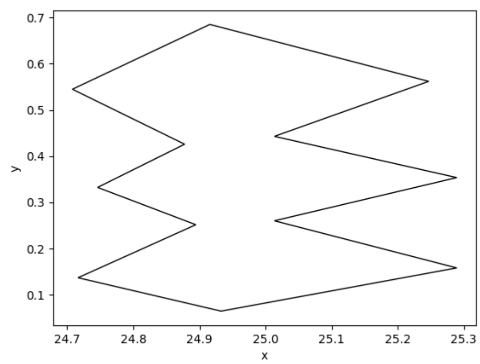
- Algorytm sprawdzenia monotoniczności wielokąta względem osi OY.
- Algorytm klasyfikacji wierzchołków wielokąta ze względu na położenie jego sąsiedzi.
- Algorytm triangulacji wielokąta monotonicznego.

2.2 Zbiory testowe

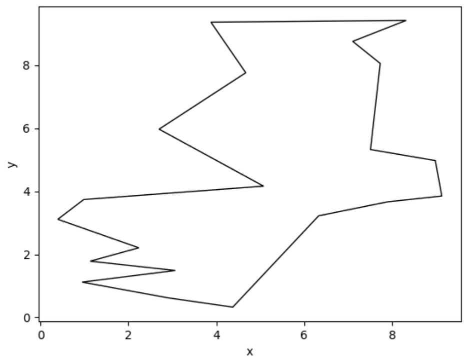
Na potrzeby ćwiczenia stworzyliśmy 6 wielokątów. Zostały wybrane odpowiednio, żeby przetestować działanie algorytmów w przypadkach zarówno lososwych jak i ekstremalnych.



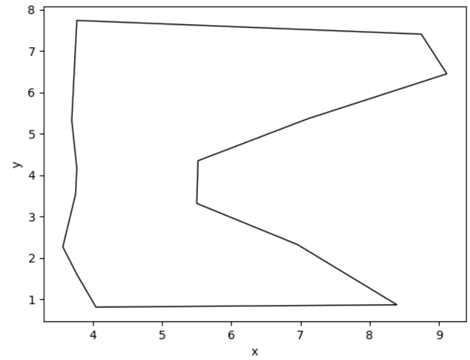
Rys. 1: Wielokąt z wykładu.



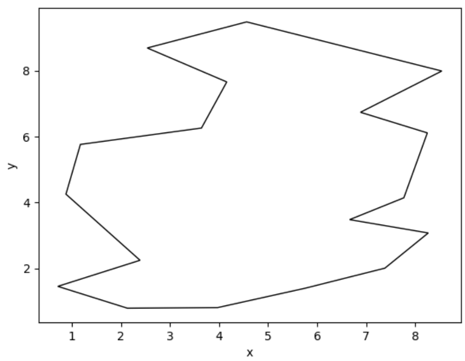
Rys. 2: Choinka.



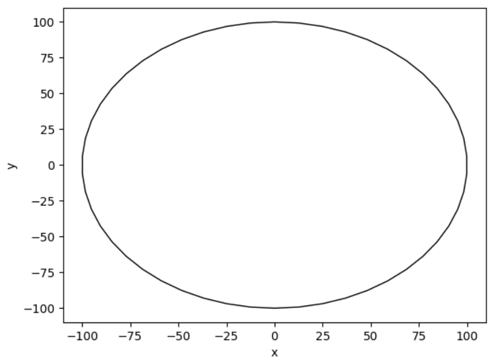
Rys. 3: Wielokąt 1.



Rys. 4: Wielokąt 2.



Rys. 5: Wielokąt 3.



Rys. 6: 50-kąt foremny.

2.3 Dalczego wybraliśmy te zbiory testowe

1. Wielokąt z wykładu to wielokąt wklęsły, który wykorzystujemy do sprawdzenia czy nasz alogrytm dobrze sprawdza czy odcinek pomiędzy wierzchołkami jest zawarty w wielokącie.
2. Choinka to przykład wielokąta, który ma wierzchołki naprziemiennie z różnych łańcuchów oraz jest wielokątem wklęsłym, co pozwala na testowanie czy nasz algorytm dobrze sprawdza czy przekątna poprowadzona między wierzchołkami leży wewnątrz wielokąta.
3. Wielokąt 1 to losowy, skomplikowany, wklęsły wielokąt.
4. Wielokąt 2 to losowy, skomplikowany, wklęsły wielokąt.
5. Wielokąt 3 to losowy, skomplikowany, wklęsły wielokąt.
6. 50-kąt foremny - przypadek ekstremalny dla wielokąta w którym wierzchołki naprzemian są z lewego i prawego łancucha.

3 Algorytmy

3.1 Algorytm triangulacji

Triangulacja wielokąta monotonicznego to proces podziału wielokąta na trójkąty, zachowując przy tym monotoniczność. Wielokąt y-monotoniczny to taki, którego prosta pozioma przecina go maksymalnie dwukrotnie. Algorytm triangulacji wielokąta monotonicznego można opisać w kilku krokach:

1. **Sortuj wierzchołki:** Posortuj malejąco wierzchołki wielokąta względem ich współrzędnej y. Dodaj dwa pierwsze wierzchołki do stosu.

2. **Podziel wierzchołki na lewy i prawy łańcuchy:** Za pomocą algorytmu działającego w $O(n)$ dzielimy wierzchołki na należące do prawego ido lewego łancuchów. Wyniki przechowujemy w liscie.

3. **Główna pętla:** Dla każdego z kolenych wierzchołków od $i = 3$ do $i = n - 1$ sprawdzamy czy jest na tym samym łancuchu co poprzedni

   - jeżeli tak, dodajemy nasz punkt na stos sprawdzamy za pomocą wyznacznika czy prosta pomiędzy pierwszym a trzecim wierzchołkami na stosie zawiera się w wielokącie, jeśli tak to tworzymy prostą między nimi, następnie usuwamy drugi wierzchołek na stosie.
   - jeżeli nie, ususwamy pierwszy wierzchołek p ze stosu, tworzymy prostą między nim a wszystkimi wierzchołkami na stosie oprócz ostatniego, a następnie dodajemy na stos badany wierzchołek oraz wierzchołek poprzedzający p.

4. **Połacz wierzchołki na stosie:** W kroku ostatnim łączymy wierzchołki, które pozostały na stosie, oprócz pierwszego i ostatniego, z wierzchołkiem o najmniejszej wspólrzędnej y.

5. **Zwróć listę połaczonych par wierżchołków**

Przedstawiony algorytm jest efektywny i działa w czasie $O(n \log n)$, gdzie n to liczba wierzchołków wielokąta. Triangulacja wielokąta monotonicznego jest często stosowana w grafice komputerowej i w problemach geometrii obliczeniowej.

Struktura przechowująca wielokąt opiera się na posortowanej liście wierzchołków oraz listy, za pomocą której możemy szybko sprawdzić do którego z łancuchów należy rozpatrywany wierzchołek. Jest to potrzebne przy sprawdzaniu czy linia między punktami jest wewnątrz wielokąta.

Wynikiem końcowym algorytmu jest krotka, której pierwszym elementem jest wielokąt początkowy, a drugim triangulacja w postaci par indeksów punktów, pomiędzy którymi należy przeprowadzić przekątne, żeby uzyskać triangulacje.

3.2 Algorytm sprawadzenia y-monotoniczności wielokąta

Nasz algorytm najpierw dzieli wierzchołki na 5 rodzajów:

1. Początkowy gdy obaj jego sąsiedzi leżą poniżej i kąt wewnętrzny $< \pi$.

2. Końcowy, gdy obaj jego sąsiedzi leżą powyżej i kąt wewnętrzny $< \pi$.

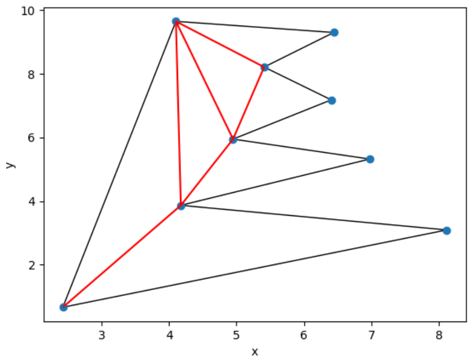
3. Łączący, gdy obaj jego sąsiedzi leżą powyżej i kąt wewnętrzny $> \pi$.

4. Dzielący, gdy obaj jego sąsiedzi leżą poniżej i kąt wewnętrzny $> \pi$.

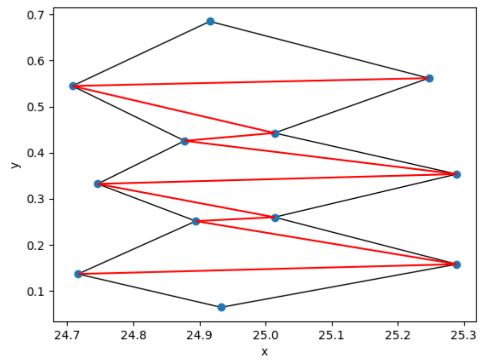
5. prawidłowy, gdy ma jednego sąsiada powyżej, drugiego - poniżej.

Robi to licząc wyznacznik i odpowiednio porównując współrzędne swoich sąsiadów. Wtedy sprawdzenie czy wielokąt jest $y-$monotoniczny sprowadza się do sprawdzenia, czy nasz wielokąt posiada wierzchołki łączące lub dzielące. Jeśli tak - nie jest monotoniczny, w przeciwnym wypadku - jest.

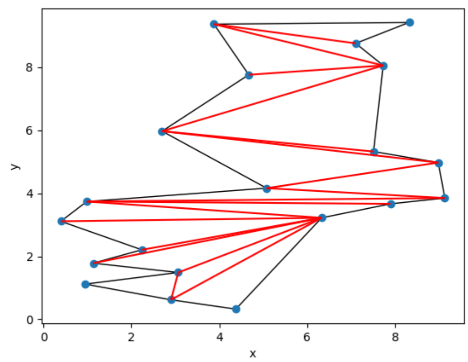
4 Wyniki triangulacji



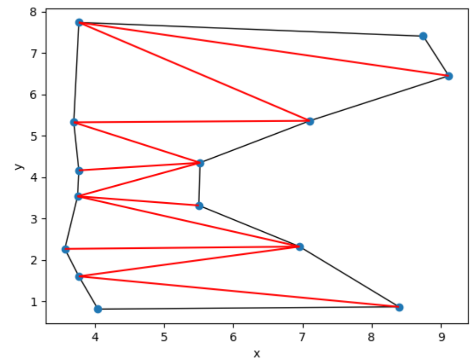
Rys. 7: Triangulacja pięciokąta.



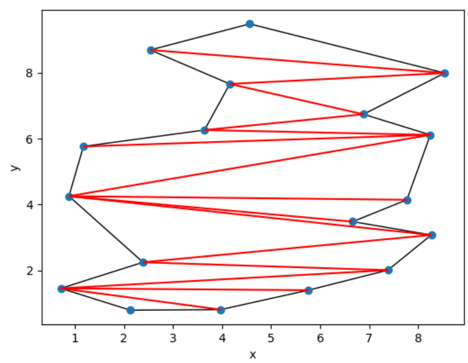
Rys. 8: Triangulacja choinki.



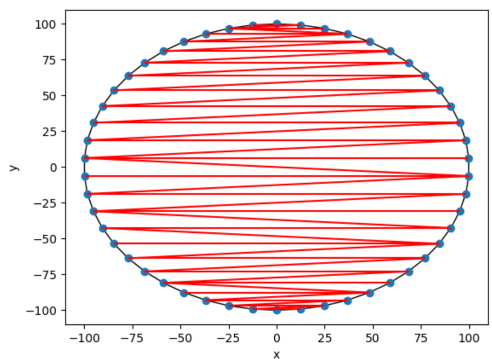
Rys. 9: Triangulacja wielokąta 1.



Rys. 10: Triangulacja wielokąta 2.



Rys. 11: Triangulacja wielokąta 3.



Rys. 12: Triangulacja 50-kąta foremnego.

5 Podsumowanie

Wszystkie zaimplementowane procedury i algorytmy zadziałały poprawnie dla testowanych wielokątów. Ponadto algorytm poprawnie nie generuje duplikatów krawędzi. Użyte struktury danych doskonale pasowały do zadanych problemów - pozwalały na szybki dostęp do potrzebnych elementów, nie wymagały dodatkowych konwersji.

Jedną ze zmian, którą trzeba było zrealizować było zwracanie w wyniku triangulacji, zarówno oryginalnego wielokąta jak i triangulacji po indeksach. To pozwala na wygodny odczyt tej struktury i możliwości szybkiej wizualizacji wyników. Także ten sposób przechowywania wyniku algorytmu uniemożliwia błędy związane ze złym dopasowaniem triangulacji do właściwego wielokąta, gdyż zarówno wielokąt, jak i wynik są przechowywane razem. Jeszcze jedną rzeczą na którą trzeba było zwrócić uwagę była reprezentacja triangulacji jako listy par indeksów wierzchołków, zamiast listy par współrzędnych wierzchołków. To pozwala przede wszystkim na zmiejszenie ilości używanej pamięci, ale również jest bardziej eleganckim rowiązaniem, które upraszcza implementacje algorytmu.