

# Algorytmy geometryczne – laboratorium 3.

## Triangulacja wielokątów monotonicznych

### 1. Dane techniczne:

- System operacyjny: Windows 11 Pro (x64)
- Procesor: AMD Ryzen 5 7530U (2.0 – 4.5 GHz)
- Pamięć RAM: 16GB (2133 MHz)
- Środowisko: Jupyter Notebook
- Język programowania: Python 3.13.7

Podczas przeprowadzenia doświadczenia użyto narzędzia do wizualizacji danych autorstwa Koła Naukowego BIT oraz bibliotek *numpy*, *pandas*, *matplotlib*.

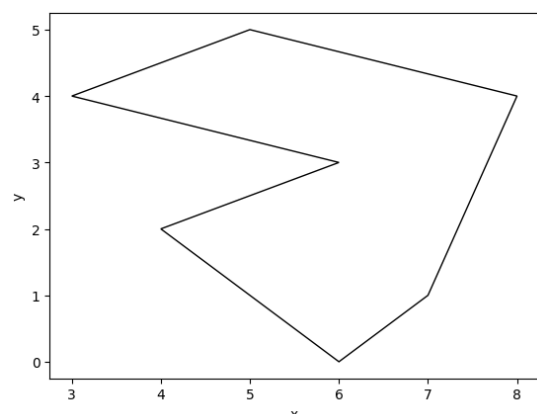
### 2. Cel ćwiczenia

- Wdrożenie do zagadnień związanymi ze sprawdzaniem monotoniczności wielokąta, klasyfikacją jego wierzchołków oraz triangulacją wielokątów y-monotonicznych.
- Analiza danych oraz wizualizacja algorytmów, triangulacji wielokątów.

### 3. Wstęp teoretyczny

#### 3.1. Definicja monotoniczności wielokąta

Wielokąt można nazwać monotonicznym względem zadanej prostej  $l$  (prostej wyznaczającej kierunek monotoniczności), kiedy możliwy jest podział jego wierzchołków na dwa łańcuchy, które są monotoniczne względem kierunku prostej  $l$ , tzn., gdy dla każdej prostej  $l'$  prostopadłej do  $l$  łańcuch przecinany jest w maksymalnie jednym punkcie.



Rys. nr 1: Przykładowy wielokąt y-monotoniczny.

Wielokątem y-monotonicznym nazwać można wielokąt, którego ułożenie wierzchołków pozwala na przejście z wierzchołka o najmniejszej współrzędnej y do wierzchołka o największej współrzędnej y, poruszając się wzdłuż lewego lub prawego łańcucha, po

punktach o stale rosnącej współrzędnej y. Przykład takiego wielokąta zobrazowano na rysunku o nr 1.

### 3.2. Algorytm sprawdzania y-monotoniczności wielokąta

Algorytm polega na fakcie zadania wierzchołków w kolejności przeciwnej do ruchu wskazówek zegara.

Weryfikacja y-monotoniczności zadanego wielokąta odbywa się poprzez następujące kroki algorytmu:

- Wyznaczenie punktów o najmniejszej oraz największej współrzędnej y.
- Przejście z punktu o najmniejszej wartości y do punktu o największej wartości zgodnie z kolejnością indeksowania ze sprawdzeniem warunku ciągłego rośnięcia współrzędnej y względem poprzedniego punktu.
- Przejście z punktu o największej wartości y do punktu o najmniejszej wartości zgodnie z kolejnością indeksowania ze sprawdzeniem warunku ciągłego malenia współrzędnej y względem poprzedniego punktu.

Jeśli algorytm napotka punkt, który nie spełnia warunku monotoniczności współrzędnej y przerwie swoje wykonanie oraz zwróci wartość False. W przeciwnym wypadku zwróci wartość True. Przejście po indeksach punktów odbywa się z wykorzystaniem operacji modulo tak, by nie umożliwić programowi wyjście poza zakres tablicy przechowywanych wierzchołków wielokąta.

Złożoność obliczeniowa zastosowanego algorytmu wynosi  $O(n)$ , gdzie n to liczba wierzchołków zadanego wielokąta.

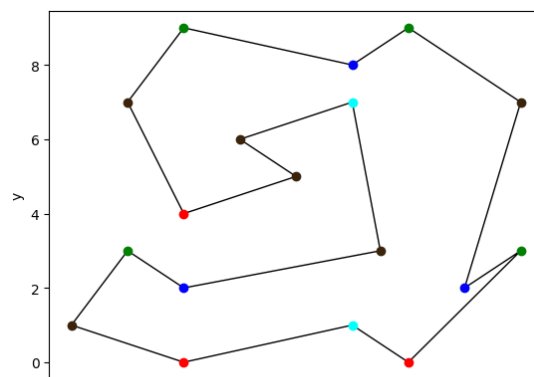
Zależy ona od głównych operacji:

- wyboru punktów o najmniejszej oraz największej współrzędnej y -  $O(n)$ ,
- przejście po kolejnych punktach względem ich indeksów -  $O(n)$ .

### 3.3. Kategoryzacja wierzchołków wielokąta

Dla wierzchołków każdego wielokąta można wprowadzić podział na poniższe kategorie:

- **wierzchołek początkowy** – obaj jego sąsiedzi leżą poniżej, a kąt wewnętrzny utworzony przez te 3 wierzchołki  $< \pi$ ,
- **wierzchołek końcowy** – obaj jego sąsiedzi leżą powyżej, a kąt wewnętrzny utworzony przez te 3 wierzchołki  $< \pi$ ,



Rys nr. 2: Przykładowy podział wierzchołków wielokąta nie-y-monotonicznego.

- **wierzchołek dzielący** – obaj jego sąsiedzi leżą poniżej a kąt wewnętrzny utworzony przez te 3 wierzchołki  $> \pi$ ,
- **wierzchołek łączący** – obaj jego sąsiedzi leżą powyżej, a kąt wewnętrzny utworzony przez te 3 wierzchołki  $> \pi$ ,
- **wierzchołek prawidłowy** – pozostałe wierzchołki.

Przykładowy podział wierzchołków wielokąta nie-y-monotonicznego z zachowaniem kolorowego sposobu oznaczeń zobrazowano na rys. o nr 2.

Wielokąt y-monotoniczny posiada dokładnie 1 wierzchołek początkowy oraz 1 wierzchołek końcowy. Nie posiada on także wierzchołków dzielących oraz łączących.

### 3.4. Algorytm kategoryzacji wierzchołków wielokąta

Algorytm grupujący punkty wykorzystuje funkcję obliczającą wartość wyznacznika, która określa ich położenie względem siebie. Oblicza ona oraz zwraca wartość wyrażenia:

$$\det(A, B, C) = \begin{vmatrix} A_x & A_y & 1 \\ B_x & B_y & 1 \\ C_x & C_y & 1 \end{vmatrix} = (B_x - A_x) \cdot (C_y - B_y) - (B_y - A_y) \cdot (C_x - B_x)$$

Weryfikacja własności wierzchołka wielokąta odbywa się poprzez powtórzenie poniższych kroków dla każdego wierzchołka wielokąta:

- Weryfikacja położenia sąsiadów względem aktualnie rozpatrywanego poprzez porównanie współrzędnych y.
- Obliczenie wartości  $\det(\text{wierzchołek poprzedni}, \text{wierzchołek aktualnie rozpatrywany}, \text{wierzchołek kolejny})$  dzięki określić można kąt występujący między nimi:
  - kąt jest wklęsły,  $\text{kąt} > \pi \Leftrightarrow \det(\text{poprzedni}, \text{aktualny}, \text{kolejny}) < 0$ ,
  - kąt jest wypukły,  $\text{kąt} < \pi \Leftrightarrow \det(\text{poprzedni}, \text{aktualny}, \text{kolejny}) > 0$ .
 Określenie wyznacznika odbywa się z poprawką  $\varepsilon = 10^{-24}$  związaną z koniecznością operowania na zmiennych typu float.

Dopasowanie kategorii do każdego z punktów odbyło się zgodnie zasadami opisanymi w punkcie 3.3.

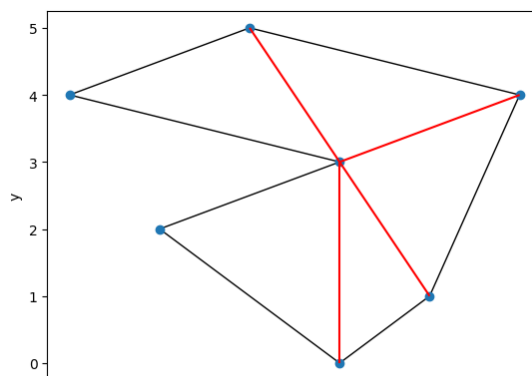
Złożoność obliczeniowa zastosowanego algorytmu wynosi  $O(n)$ , gdzie  $n$  to liczba wierzchołków zadanego wielokąta.

Zależy ona od głównych operacji:

- przejście po kolejnych punktach względem ich indeksów -  $O(n)$ ,
- działania matematyczne stosowane do określenia kategorii punktów -  $O(1)$ .

### 3.5. Triangulacja wielokąta

Triangulacją wielokąta polega na podziale jego pola na nieprzecinające się trójkąty poprzez dodanie nieprzecinających się przekątnych między niektórymi z jego wierzchołków. Wszystkie dodane przekątne w procesie triangulacji muszą należeć do wnętrza wielokąta z końcami leżącymi w miejscu jego wierzchołków. Dla  $n$ -wierzchołkowego wielokąta triangulacja zawsze będzie składać się z dokładnie  $n - 2$  trójkątów otrzymanych poprzez dodanie dokładnie  $n - 3$  przekątnych.



Rys. nr 3: Przykładowa triangulacja wielokąta y-monotonicznego.

Przykład triangulacji wielokąta y-monotonicznego zobrazowano na rys. o nr 3.

### 3.6. Algorytm triangulacji wielokąta y-monotonicznego

Do wyznaczenia triangulacji wielokątów y-monotonicznych użyty został algorytm, który także wykorzystuje funkcję obliczającą wartość wyznacznika określoną w punkcie 3.4.

Algorytm zwraca tablicę krotek, par indeksów, które stanowią końce kolejnych przekątnych.

Do przechowywania listy poprzednich wierzchołków wykorzystano stos ze względu na jego łatwość w implementacji oraz wykonywania operacji usuwania oraz dodawania kolejnych elementów.

Kroki algorytmu:

- Wyznaczenie punktów o najmniejszej oraz największej współrzędnej y.
- Podzielenie punktów na 2 łańcuchy - lewy oraz prawy do których należą odpowiednio punkt o największej oraz o najmniejszej współrzędnej y.
- Scalenie łańcuchów z uporządkowaniem wierzchołków malejąco względem współrzędnej y.
- Inicjalizacja stosu z 2 pierwszymi wierzchołkami ze scalonej listy.
- Dla każdego kolejnego wierzchołka ze scalonej listy:
  - jeśli leży on na tym samym łańcuchu, co wierzchołek poprzedni, to algorytm cofa się i dodaje przekątną między aktualnym wierzchołkiem oraz tymi znajdującymi się wcześniej na stosie, dopóki dodana przekątna byłaby poprawna. Następnie na stos zostaje dodany aktualny wierzchołek wielokąta.
  - jeśli leży on na innym łańcuchu, niż wierzchołek poprzedni, to algorytm dodaje poprawne przekątne między aktualnym wierzchołkiem, a wszystkimi wierzchołkami ze stosu. Na stosie zostają 2 wierzchołki - pierwszy występujący ze stosu oraz aktualnie rozpatrywany.

W obu przypadkach warunkiem poprawnej przekątnej jest brak boku wielokąta między wierzchołkami, które mają być jej końcami. Dodatkowo w pierwszym przypadku rozpatrywana jest wartość wyznacznika obliczonego na podstawie 2 wierzchołków ze stosu oraz wierzchołka aktualnie rozpatrywanego, aby upewnić się, że proponowany trójkąt nie będzie trójkątem zewnętrznym.

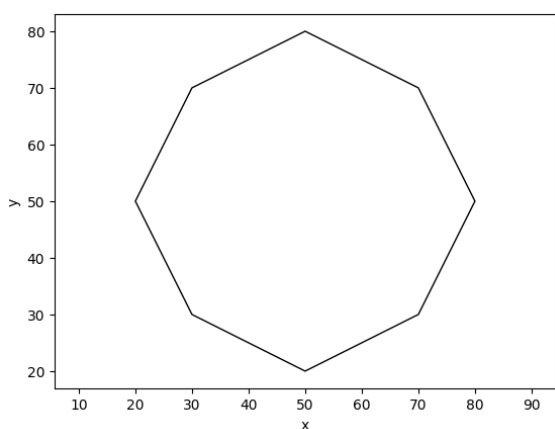
Złożoność obliczeniowa zastosowanego algorytmu wynosi  $O(n)$ , gdzie  $n$  to liczba wierzchołków zadanego wielokąta.

Zależy ona od głównych operacji:

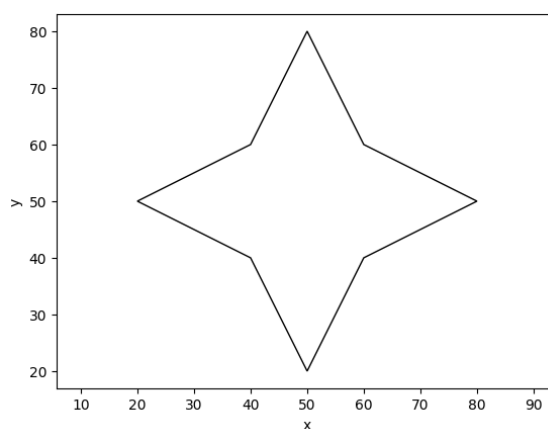
- wyboru punktów o najmniejszej oraz największej współrzędnej  $y$  -  $O(n)$ ,
- rozdzielenie punktów na lewy oraz prawy łańcuch -  $O(n)$ ,
- scalenie łańcuchów -  $O(n)$ ,
- przejście po kolejnych punktach względem ich indeksów -  $O(n)$ ,
- działania matematyczne oraz operacje na stosie -  $O(n)$ .

## 4. Przebieg doświadczenia

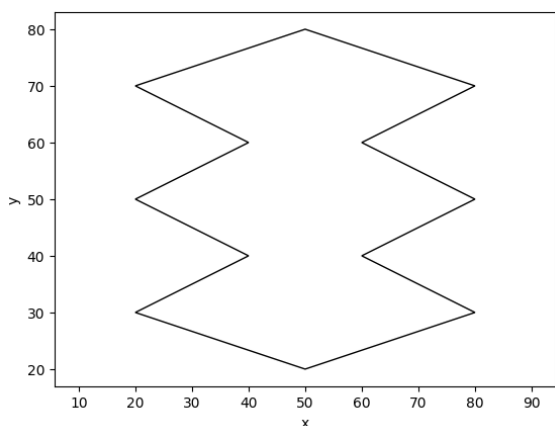
Doświadczenie rozpoczęto od wprowadzenia oraz zapisania do pliku 6 wielokątów testowych z użyciem wcześniej zaimplementowanego interaktywnego narzędzia. Wprowadzone dane zobrazowane zostały na rysunkach o nr od 4 do 10.



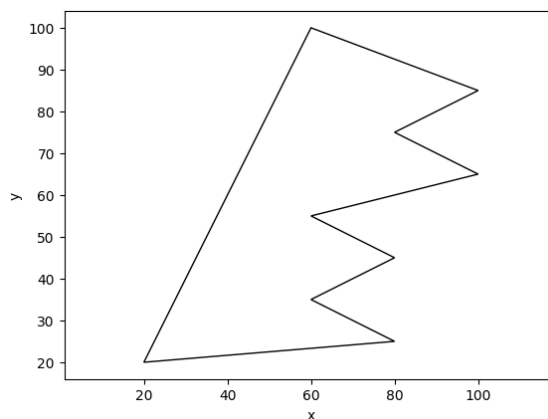
*Rys. nr 4: Wielokąt A o 8 wierzchołkach.*



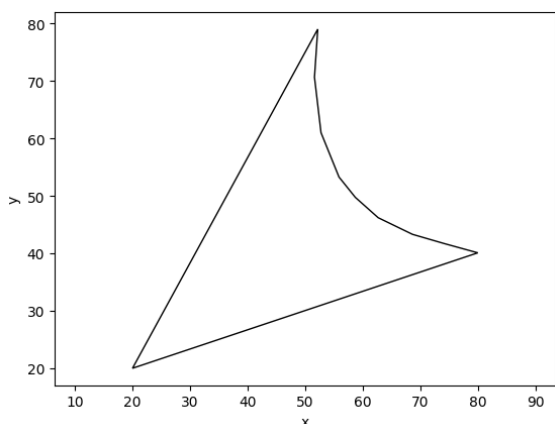
*Rys. nr 5: Wielokąt B o 8 wierzchołkach.*



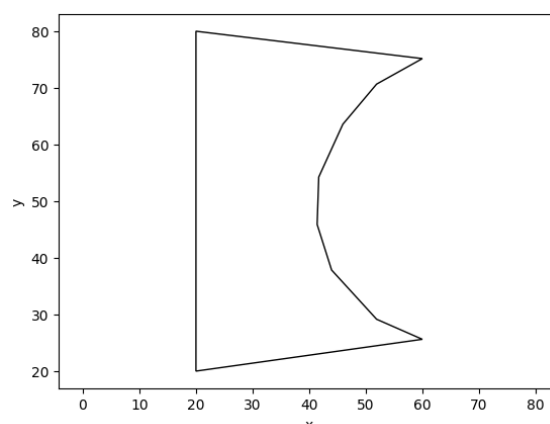
*Rys. nr 6: Wielokąt C o 12 wierzchołkach.*



*Rys. nr 7: Wielokąt D o 9 wierzchołkach.*



*Rys. nr 8: Wielokąt E o 10 wierzchołkach.*



*Rys. nr 9: Wielokąt F o 10 wierzchołkach.*

Powyższe wielokąty zostały dobrane tak, by swoimi cechami wpłynąć na jak najdokładniejsze zbadanie algorytmu w zróżnicowanych przypadkach testowych.

Wielokąt A został dobrany tak by ukazać pracę algorytmu na prostym, wypukłym wielokącie. Jego wierzchołki rozłożone są symetrycznie względem obu osi układu współrzędnych. Wielokąt B został dodany do testu jako rozwinięcie poprzednika. Połowa jego wierzchołków została przesunięta tak, by wprowadzić do wielokąta kąty wklęsłe. Jego kształt natomiast wzorowany był na matematycznej asteroidzie. Obie figury należą do grupy mało skomplikowanych wielokątów.

Wielokąt C został dobrany tak, by ukazać sposób działania algorytmu w przypadku następujących po sobie kątów wklęsłych i wypukłych zarówno w lewym, jak i prawym łańcuchu. Jego założeniem było wymuszenie na algorytmie jak najczęstszych przeskoków z jednego łańcucha na przeciwny.

Wielokąt D został dobrany tak, by ukazać sposób działania algorytmu w przypadku następujących po sobie kątów wklęsłych i wypukłych tylko po jednej ze stron figury.

Wielokąt E został dobrany tak, by ukazać sposób działania algorytmu w przypadku następujących po sobie kątów wklęsłych naprzemiennie naprzeciwko jednego wierzchołka.

Wielokąt F został dobrany jako rozszerzenie przypadku wielokąta E, by ukazać sposób działania algorytmu w przypadku następujących po sobie kątów wklęsłych naprzemiennie dwóch wierzchołków.

Kolejnym krokiem doświadczenia było zaimplementowanie algorytmów opisanych w punkcie 3 opracowania. Posłużono się nimi w następnych etapach doświadczenia.

Każdy z wielokątów przykładowych odczytano z pliku oraz zweryfikowano pod względem y-monotoniczności. Następnie ich wierzchołki pogrupowano zgodnie z kryteriami punktu 3.3. Wygenerowano także wizualizacje tego dopasowania dla wszystkich figur.

Drugim etapem doświadczenia było wyznaczenie triangulacji dla każdego z wprowadzonych wielokątów. Wizualizacje wyniku oraz działania algorytmu wygenerowano w postaci grafik

oraz gifów. Posłużono się nimi podczas rozważań związanymi z zapisaniem wniosków doświadczenia.

## 5. Analiza wyników

### 5.1. Y-monotoniczność zadanych wielokątów

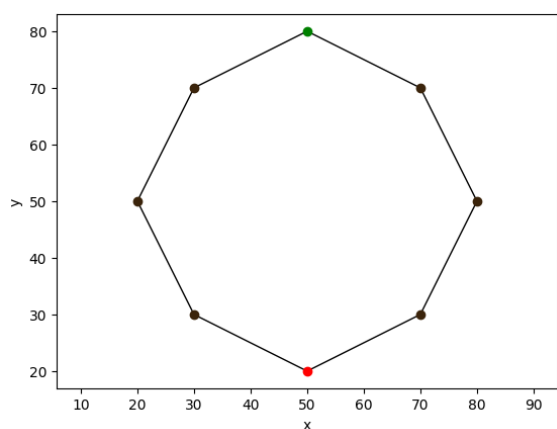
Każdy z wielokątów wprowadzono i poddano weryfikacji y-monotoniczności z użyciem wcześniej opisanego algorytmu. Otrzymane wyniki wprowadzono do tabeli nr 1.

Czy zadany wielokąt jest y-monotoniczny?					
A	B	C	D	E	F
Tak	Tak	Tak	Tak	Tak	Tak

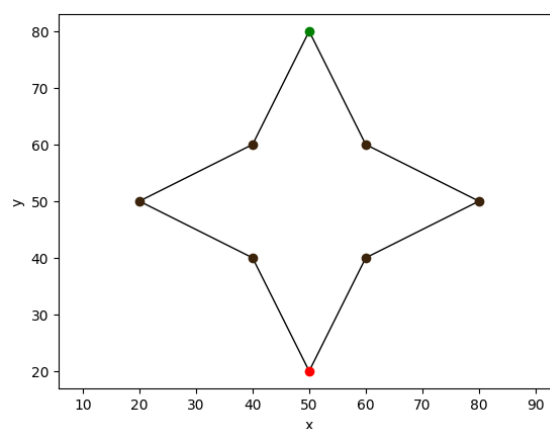
**Tabela nr 1:** Wyniki algorytmu weryfikującego y-monotoniczność zadanych wielokątów.

### 5.2. Kategoryzacja punktów zadanych wielokątów

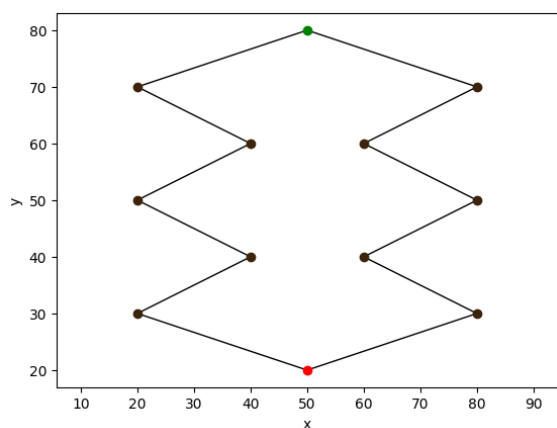
Wierzchołki każdego z wielokątów poddano kategoryzacji ze względu na zasady opisane w punkcie 3.3. Otrzymane wyniki przedstawiono na rysunkach o nr od 10 do 15, a ich numeryczną formę wprowadzono do tabeli nr 2.



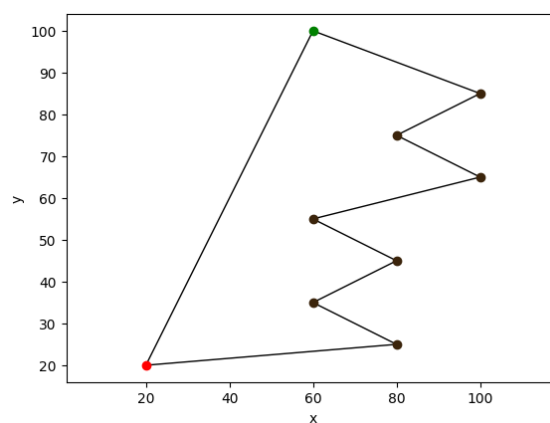
**Rys. nr 10:** Kategoryzacja wierzchołków wielokąta A.



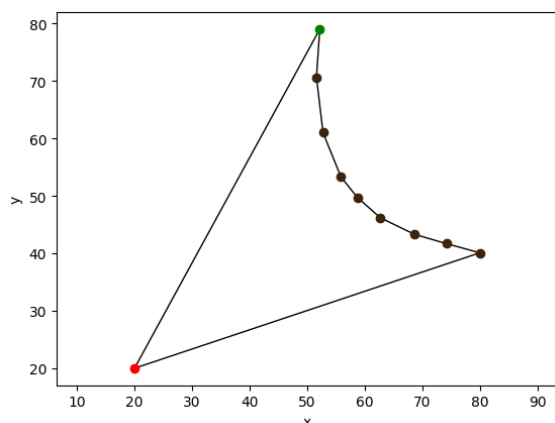
**Rys. nr 11:** Kategoryzacja wierzchołków wielokąta B.



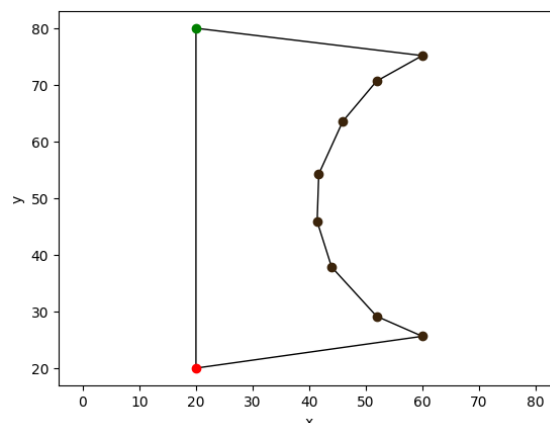
**Rys. nr 12:** Kategoryzacja wierzchołków wielokąta C.



**Rys. nr 13:** Kategoryzacja wierzchołków wielokąta D.



Rys. nr 14: Kategoriezacja wierzchołków wielokąta E.



Rys. nr 15: Kategoriezacja wierzchołków wielokąta F.

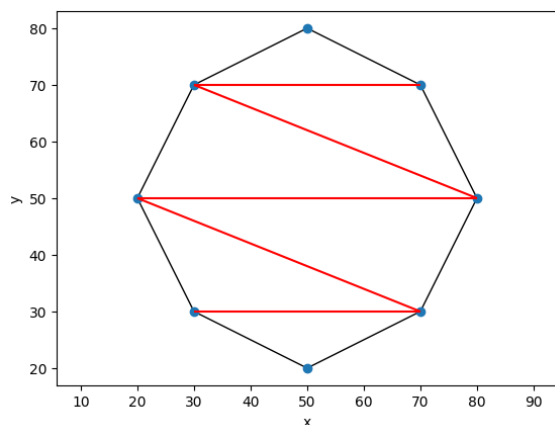
Liczba wierzchołków zadanych wielokątów dopasowanych do poszczególnych kategorii						
Kategoria wierzchołków	Wielokąt					
	A	B	C	D	E	F
Początkowe	1	1	1	1	1	1
Końcowe	1	1	1	1	1	1
Dzielące	0	0	0	0	0	0
Łączące	0	0	0	0	0	0
Prawidłowe	6	6	10	7	8	8

Tabela nr 2: Liczba wierzchołków zadanych wielokątów dopasowanych do kategorii.

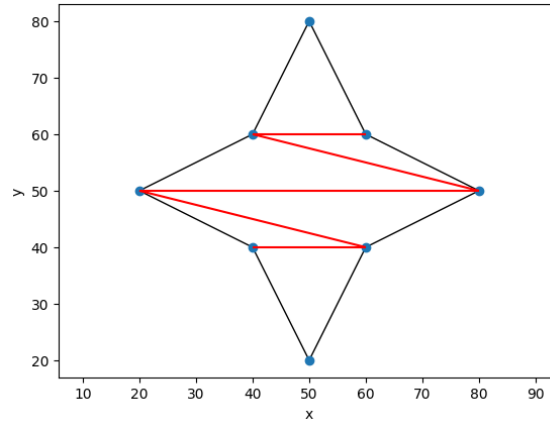
Każdy z wielokątów zawierał dokładnie 1 wierzchołek początkowy oraz dokładnie 1 wierzchołek końcowy. Żaden z punktów wielokątów nie został zakwalifikowany jako wierzchołek dzielący lub łączący.

### 5.3. Wyznaczenie triangulacji zadanych wielokątów

Dla każdego z wielomianów wyznaczono poprawną triangulację przy użyciu algorytmu opisanego w punkcie 3.6. Otrzymane wyniki przedstawiono na rysunkach o nr od 16 do 21. Numeryczne dane związane z triangulacją zadanych wielokątów wprowadzono do tabeli o nr 3. Dodane przekątne oznaczono kolorem czerwonym.

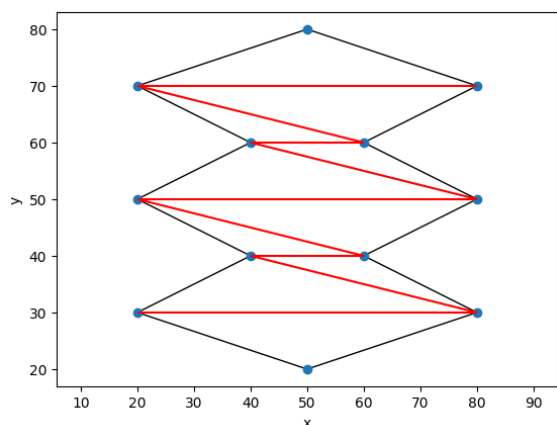


Rys. nr 16: Triangulacja wielokąta A.

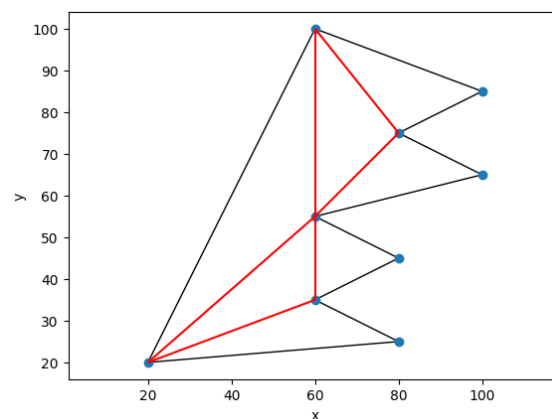


Rys. nr 17: Triangulacja wielokąta B.

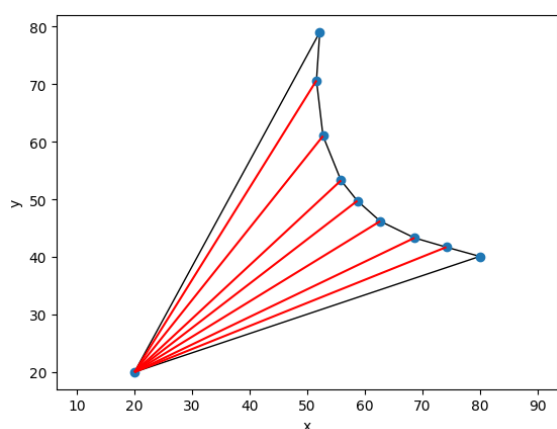




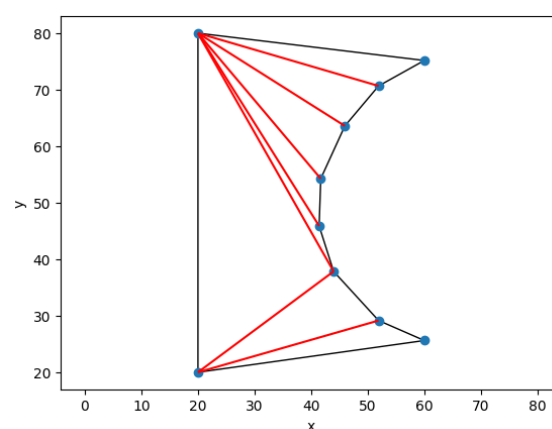
Rys. nr 18: Triangulacja wielokąta C.



Rys. nr 19: Triangulacja wielokąta D.



Rys. nr 20: Triangulacja wielokąta E.



Rys. nr 21: Triangulacja wielokąta F.

Wartości liczbowe związane z triangulacją zadanych wielokątów						
Wartość	Wielokąt					
	A	B	C	D	E	F
Liczba trójkątów	6	6	10	7	8	8
Liczba dodanych przekątnych	5	5	9	6	7	7

Tabela nr 3: Wartości liczbowe związane z triangulacją zadanych wielokątów.

Twierdzenie z punktu 3.4. o liczbie trójkątów równej  $n - 2$  w triangulacji oraz liczbie dodanych przekątnych równej  $n - 3$ , gdzie  $n$  to liczba wierzchołków wielokąta, zostało potwierdzone metodą doświadczalną. Żaden z wyników nie odbiega od przyjętej zasady. Dla każdego wielokąta algorytm wygenerował poprawną triangulację.

Zarówno wielokąt A, jak i B został wypełniony powtarzalnym wzorem, którego naśladownictwa można doszukać się także na wizualizacji triangulacji wielokąta C. Zadaniem testów z użyciem wymienionych wielokątów było sprawdzenie poprawności

implementacji dla prostych przypadków oraz poprawności zarządzania stosem (w przypadku wielokąta C).

Zadaniem testu z użyciem wielokąta D było sprawdzenie zachowania algorytmu w przypadku działania na wielokącie, którego zagęszczenie punktów różni się względem obu łańcuchów oraz poprawnej obsługi mechanizmu łączenia krawędzi

Zadaniem testu z użyciem wielokątów E oraz F było sprawdzenie poprawności implementacji mechanizmu łączenia wierzchołków ze stosu z nowym wierzchołkiem oraz mechanizmu odrzucania trójkątów zewnętrznych.

## 6. Wnioski

Na podstawie analizy wyników doświadczenia nasuwają się następujące wnioski:

- 1) Dowiedziono poprawności implementacji weryfikacji y-monotoniczności, kategoryzacji wierzchołków wielokątów oraz samego wyznaczania triangulacji wielokąta poprzez wykazanie zbieżności wyników przypadków testowych z teoretycznymi przewidywaniami.
- 2) Cecha y-monotoniczności wielokątów znacząco ułatwia proces wyznaczania ich triangulacji.
- 3) Stos okazał się użyteczną strukturą danych. Jego zastosowanie pozytywnie wpłynęło na łatwość samej implementacji oraz uniknięcie komplikacji związanych z problemami obsługi przecinania się dodanych przekątnych.
- 4) Liczba trójkątów, na które podzielone zostały obszary wielokątów oraz liczba dodanych przekątnych zgadzały się z teoretycznymi przewidywaniami oraz wzorem.

Wizualizacja wyników algorytmów oraz dokładna selekcja przypadków testowych pozwoliła na weryfikację poprawności zaimplementowanych algorytmów. Posługiwanie się narzędziem graficznym zdecydowanie wspomaga ocenę jakości zaproponowanych rozwiązań, a także wspomaga dogłębne zrozumienie idei stosowanych mechanizmów algorytmicznych.

Należy pamiętać, że problem wyznaczenia poprawnej triangulacji ma często wiele poprawnych rozwiązań. Dla testowanych wielokątów zwizualizowane wyniki są jedną z wielu poprawnych możliwości.