Ćwiczenie III - Triangulacja wielokątów monotonicznych

Jakub Frączek - grupa nr 4

1. Wstęp

Opis ćwiczenia

Ćwiczenie polegało na:

1) Zmodyfikowaniu narzędzia graficznego tak, aby można było zadawać wielokąty z myszki
2) Zaimplementowaniu procedury sprawdzającej czy wielokąt jest y - monotoniczny
3) Zaimplementowaniu algorytmu, który dla zadanego wielokąta będzie wyszukiwał wierzchołki początkowe, końcowe, łączące, dzielące i prawidłowe.
4) Zaimplementowaniu procedury triangulacji wielokąta monotonicznego

Dane techniczne - software

Ćwiczenie zostało zrealizowane w języku Python przy użyciu środowiska Jupyter notebook. Wykorzystane biblioteki to: numpy, random, pandas, matplotlib, bitalg, functools, collections, time.

Dane techniczne - hardware

Laptop z systemem operacyjnym Linux Mint, procesorem AMD Ryzen 5 5500U 2.1 GHZ oraz 8 GB pamięci RAM.

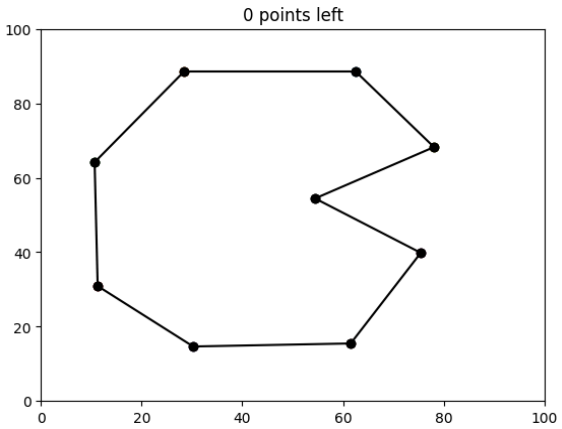
2. Klasyfikacja wierzchołków wielokąta

W wielokącie wyróżniamy 5 różnych typów wierzchołków:

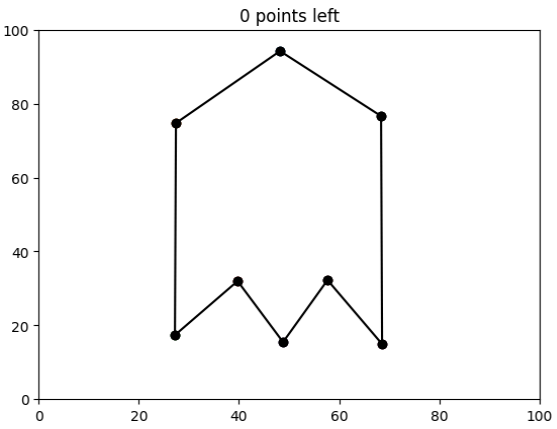
- Początkowy, gdy obaj sąsiedzi leżą poniżej i kąt wewnętrzny < pi
- Końcowy, gdy obaj sąsiedzi leżą powyżej i kąt wewnętrzny < pi
- Łączący, gdy obaj sąsiedzi leżą powyżej i kąt wewnętrzny > pi
- Dzielący, gdy obaj sąsiedzi leżą poniżej i kąt wewnętrzny > pi
- Prawidłowy, gdy ma jednego sąsiada poniżej, a drugiego powyżej

3. Algorytm sprawdzający y - monotoniczność wielokąta

Algorytm, wykorzystuje własność wielokątów y - monotonicznych polegającą na tym, że nie zawierają one wierzchołków dzielących, ani wierzchołków łączących. Zatem dla każdej trójki wierzchołków leżących obok siebie algorytm dokonuje klasyfikacji środkowego wierzchołka, jeśli okaże się, że nie było wierzchołków ani dzielących, ani łączących, to wielokąt zostaje określony jako y - monotoniczny. Poniżej znajdują się przykładowe wielokąty wraz z rezultatem wykonania dla nich algorytmu (Wykres 1. i Wykres 2.).



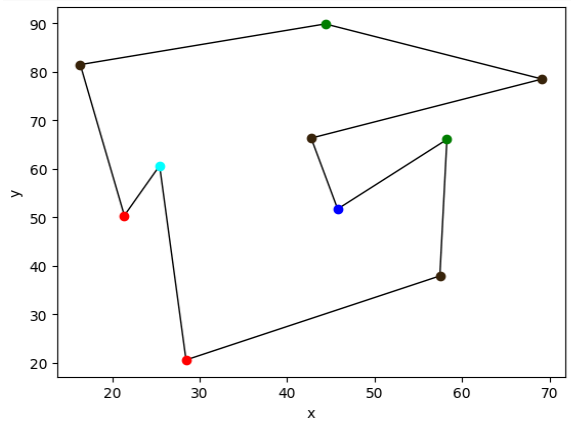
Wykres 1. y - monotoniczny



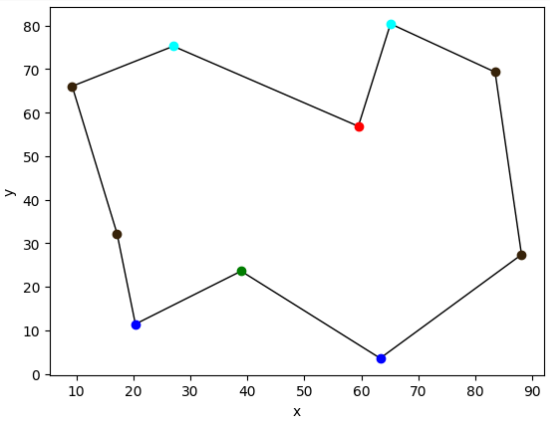
Wykres 2. nie y - monotoniczny

4. Algorytm klasyfikujący wierzchołki wielokąta

Algorytm podobnie jak powyższy dla każdej trójki wierzchołków leżących obok siebie dokonuje klasyfikacji tego środkowego na podstawie własności z punktu 2. Na wykresach 1 i 2 przedstawiona jest klasyfikacja wierzchołków dla dwóch różnych wielokątów (1 i 2).



Wykres 3. Wielokąt 1.



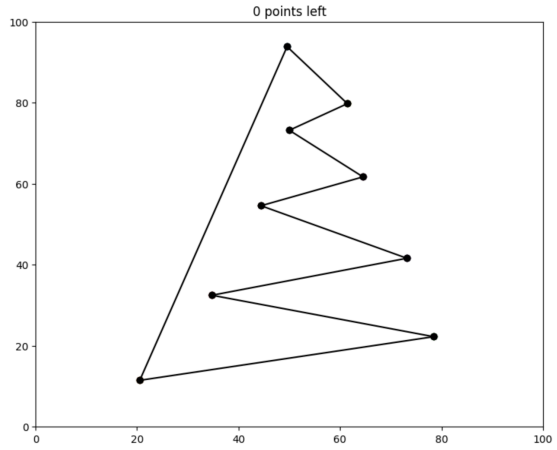
Wykres 4. Wielokąt 2.

5. Algorytm triangulacji wielokąta monotonicznego

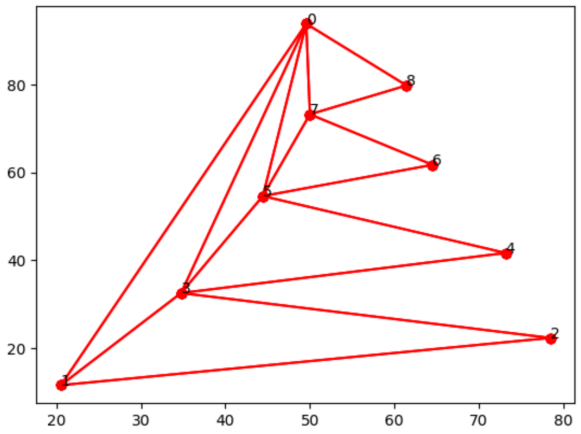
Algorytm dokonuje triangulacji wielokąta y - monotonicznego. Wynikowy wielokąt jest przechowywany w zadeklarowanej przeze mnie strukturze danych YMonotonicTriangulation, która w konstruktorze przyjmuje graf w postaci macierzowej oraz listę wierzchołków wielokąta. W grafie każda krawędź oznacza bok lub przekątną. Takie podejście pozwala całkowicie wyeliminować możliwość powstawania duplikatów przekątnych, ponieważ przy próbie próbie dodania zduplikowanej krawędzi do takiego grafu, nie będzie to miało żadnego efektu (Krawędzie są oznaczane jako 1, a brak krawędzi jako 0) oraz nie stracimy na wydajności. Natomiast do rekonstrukcji wielokąta potrzebny będzie zarówno graf jak i wejściowa lista wierzchołków wielokątów. Wierzchołki zostały zmapowane w kolejności występowania na liście do kolejnych liczb naturalnych oraz te same liczby reprezentują wierzchołki wielokąta w grafie. Algorytm do triangulacji wielokąta y - monotonicznego można przedstawić następująco:

- Należy zweryfikować, czy wielokąt jest y - monotoniczny
- Rozdzielenie wielokąta na dwa łańcuchy prawy oraz lewy. Podział polega na znalezieniu punktów o największej oraz najmniejszej współrzędnej y - owej, na następnie umieszczenie w jednej tablicy punktów iterując po tablicy punktów, od największego wierzchołka, aż do napotkania najmniejszego wierzchołka, a w drugiej reszty punktów
- Posortowanie tablicy punktów po współrzędnej y - owej malejąco
- Umieszczenie dwóch pierwszych wierzchołków z posortowanej tablicy na stosie
- Iteruję po pozostałych punktach punktach, rozważając ich relację z aktualnym szczytem sotsu
    - Jeśli kolejny wierzchołek jest w innym łańcuchu niż wierzchołek znajdujący się na szczycie stosu, to mogę połączyć przekątnymi ten wierzchołek z wszystkimi wierzchołkami znajdującymi się na stosie
    - Jeśli kolejny wierzchołek jest w tym samym łańcuchu co wierzchołek znajdujący się na szczycie stosu, to:
        - jeśli trójkąt, który tworzy wierzchołek z dwoma najwyższymi punktami stosu należy do wielokąta, to usuwam wierzchołek ze szczytu stosu i powtarzam procedurę dopóki wielkość stosu pozwala na stworzenie trójkąta
        - W przeciwnym wypadku umieszczam badany wierzchołek na stosie

Poniżej na wykresie 6 przedstawiono triangulację wielokąta 3. (Wykres 5.)

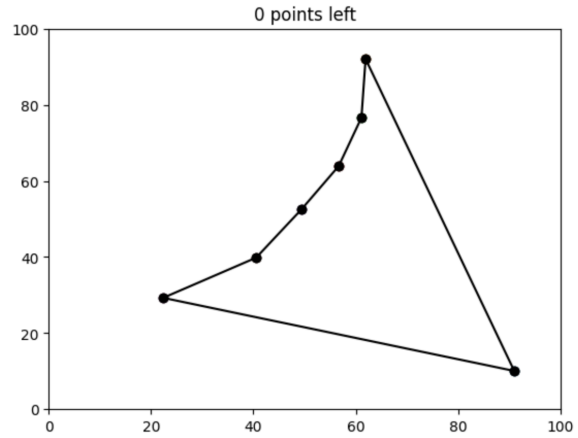


Wykres 5. Wielokąt 3.

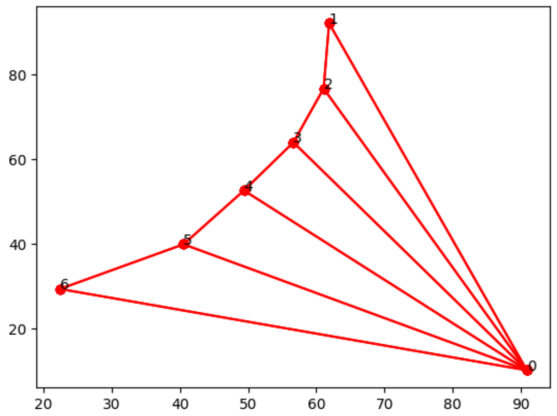


Wykres 6. Triangulacja wielokąta 3.

Na wykresie 8 przedstawiono triangulację wielokąta 4. (Wykres 7.)

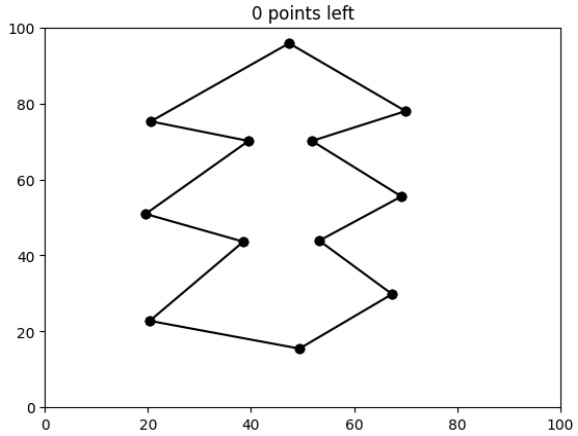


Wykres 7. Wielokąt 4.

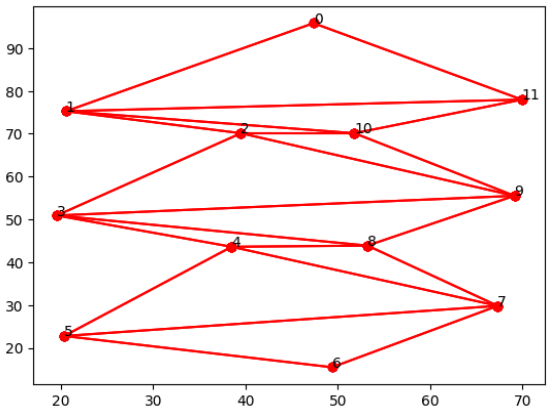


Wykres 8. Triangulacja wielokąta 4.

Na wykresie 10 przedstawiono triangulację wielokąta 5. (Wykres 9.)



Wykres 9. Wielokąt 5.



Wykres 10. Triangulacja wielokąta 5.

6. Wnioski

Zaimplementowane algorytmu działają poprawnie, co zostało w dużej mierze sprawdzone poprzez testy przygotowane przez koło naukowe BIT oraz przez testy przeprowadzone ręcznie. Dodatkowo jak wiadomo każdy wielokąt o n wierzchołkach można podzielić na n - 2 trójkątów, a powstałe triangulacje jedynie potwierdzają tę tezę. Użyte struktury danych zapewniają szybki dostęp do krawędzi oraz nie dopuszczają powstawania duplikatów.