

Marceli Jędryka

# Algorytmy Geometryczne – Laboratorium 4

Środowisko oraz sprzęt:

→ Wszystkie ćwiczenia zostały wykonane w Jupyter Notebook przy użyciu języka python oraz bibliotek Numpy i Matplotlib.

→ Obliczenia przeprowadzone na systemie operacyjnym Windows 10 x64 z procesorem Intel Core i5-10210U CPU 2.11 GHz.

## 1. Procedura inicjalizująca odcinki

Zaimplementowane zostały dwie procedury pozwalające stworzenie odcinków oraz ich zapis do pliku. Pierwsza pozwala na interaktywne zadawanie odcinków, natomiast druga generuje podaną ilość w wybranym zakresie.

## 2. Algorytm zmiatania sprawdzający, czy choć jedna para odcinków w zadanym zbiorze się przecina

Algorytm działający na bazie algorytmu wyznaczającego punkty przecięcia odcinków. Zaimplementowane zostały dwie struktury - Struktura zdarzeń Q będąca kopcem oraz struktura stanu miotły T, do której stworzenia wykorzystany został Sorted Set. W strukturze stanu T w sposób uporządkowany przechowywane są odcinki względem współrzędnej y.

Schemat działania :

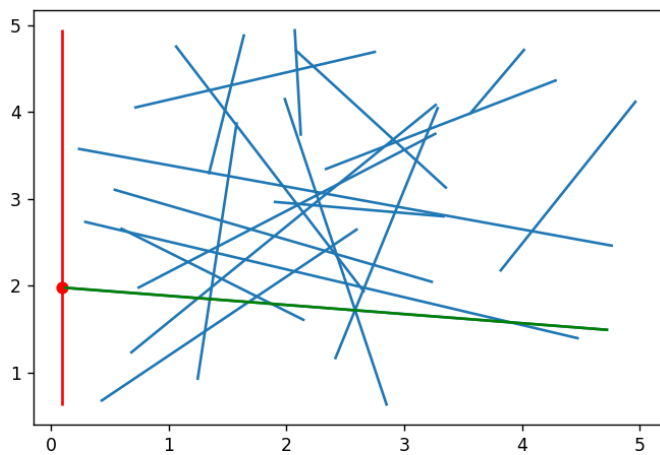
Do struktury zdarzeń Q wrzucane są wszystkie punkty końcowe każdego odcinka w zbiorze wraz z reprezentacją odcinka do którego należą (w postaci krotki). Następnie do momentu w którym kopiec Q nie będzie zbiorem pustym wybieramy kolejne elementy i sprawdzamy czy rozpatrywany obecnie punkt jest prawym czy lewym końcem odcinka. Jeżeli jest lewym to linia, do której należy odcinek dodawana jest do struktury stanu miotły T i sprawdzane jest, czy nowo dodany odcinek nie przecina się z poprzednio dodanym odcinkiem w strukturze - jeśli tak to flaga jest o tym informowana, a to wpływa na informację zwrotną programu. W przypadku kiedy rozpatrywany punkt jest prawym końcem odcinka, to odcinek ten jest usuwany ze struktury T po wcześniejszym sprawdzeniu, czy odcinki po lewej i prawej stronie odcinka w strukturze nie przecinają się. Przecinanie się odcinków sprawdzane jest przy użyciu wyznacznika .

Do programu została dodana wizualizacja pozwalająca na przeglądanie pozycji, w której obecnie znajduje się miotła, punktu będącego rozpatrywanym zdarzeniem oraz odcinka do którego ten punkt należy. W tym celu po wykryciu pierwszego przecięcia program zapisuje o tym informacje, jednakże kontynuuje działanie.

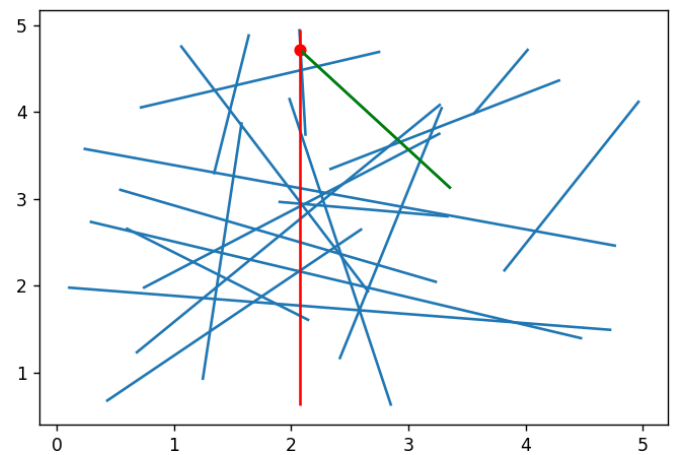
**Poniżej przedstawione są wybrane etapy wizualizacji algorytmu**

**Przykład a :**

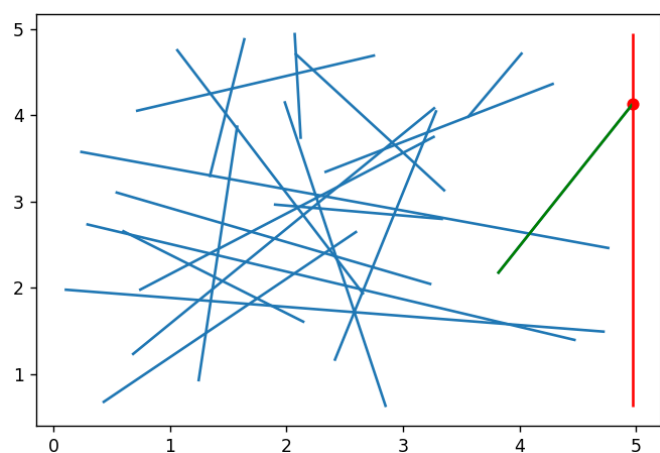
rys 2.1a



rys 2.2a



rys 2.3a

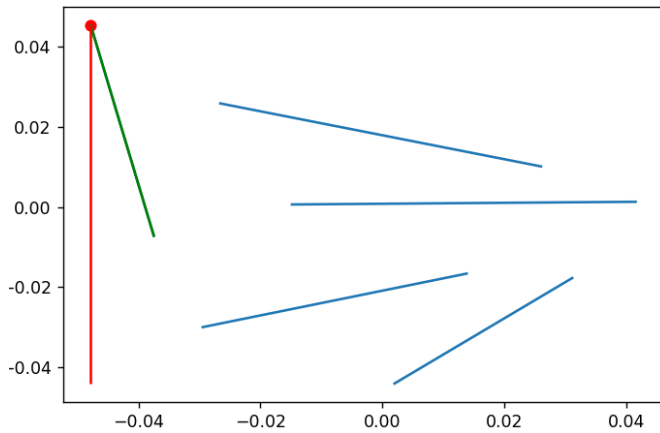


rys 2.4a

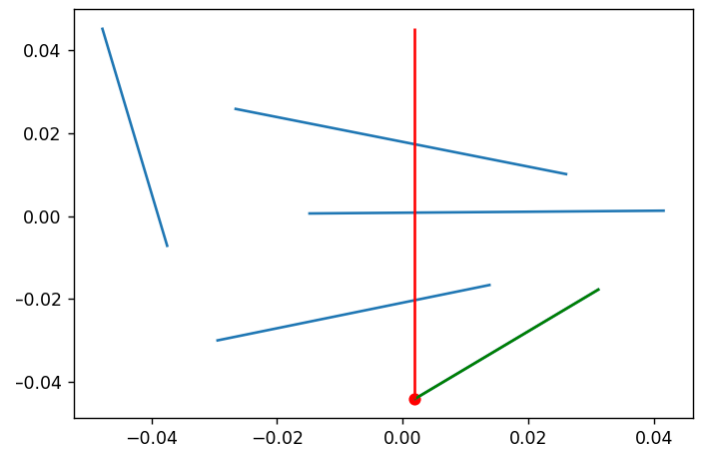
**Linie się przecinają**

### Przykład b:

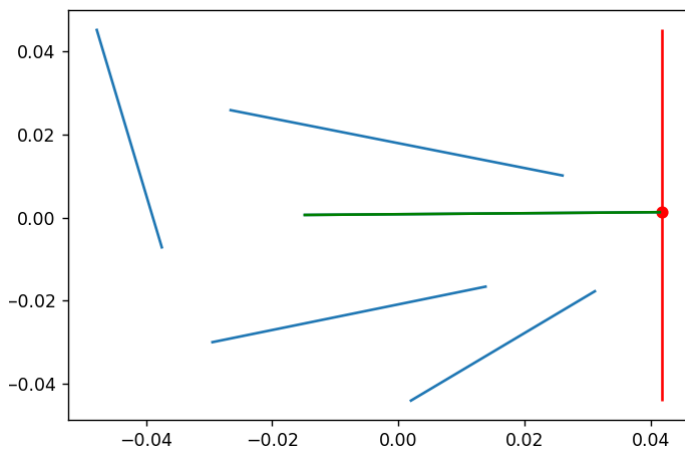
rys 2.1b



rys 2.2b



rys 2.3b



rys 2.4b

Linie się nie przecinają

### 3. Algorytm zmiatania wyznaczający wszystkie przecięcia odcinków

W tym przypadku w strukturze zdarzeń Q mogą znajdować się punkty przecięcia odcinków, a zatem potrzebujemy mieć dostęp do informacji czy dany punkt znajduje się już w tej strukturze. Problem rozwiązuje użycie struktury Sorted Set, jednak w tym wypadku do zdarzeń dodajemy tylko punkty. Informacje o odcinku, do którego należy dany punkt przetrzymywana jest w słowniku, gdzie punkt to klucz, a informacja o odcinku to wartość. Dalsza część algorytmu działa analogicznie do opisu w punkcie 2. Wyjątkiem jest przypadek, w którym rozpatrywane zdarzenie

będzie punktem przecięcia dwóch odcinków. W takim wypadku znalezione przecięcie dodawane jest zbioru z punktami przecięć, a odcinki przecinające się są zamieniane kolejnością (implementacja zgodna z instrukcjami z wykładu). W przypadku przecięć wykrywanych więcej niż jeden raz algorytm sprawdza czy przecięcie to znajduje się już w zbiorze wykrytych punktów i jeżeli tak jest, to nie pozwala na ponowne dodanie wykrytego punktu do struktury zdarzeń.

**Opis funkcji pomocniczych znajduje się w pliku z kodem**