1. Sprawdzanie czy choć jedna para odcinków się przecina

Strukturą stanu jest drzewo czerwono-czarne z biblioteki bintrees (https://github.com/mozman/bintrees). Każdy węzeł zawiera y-ową współrzędną początku odcinka (jako klucz do porównywania wartości) oraz odcinek. Jako strukturę zdarzeń wybrałem listę końców wszystkich odcinków posortowaną rosnąco po x-owej współrzędnej, ponieważ tylko raz - na początku wykonywania algorytmu - wprowadzamy wszystkie zdarzenia i później nie potrzebujemy wprowadzać nowych, bo przy wykryciu pierwszego przecięcia przerywamy działanie programu i zwracamy informację o istnieniu przecięcia.

2. Wyznaczanie wszystkich przecięć

Należy zmienić strukturę zdarzeń, ponieważ teraz jest potrzebna możliwość wprowadzania nowych zdarzeń (przecięć odcinków) do struktury w czasie O(logn). Wykorzystałem w tym celu kopiec minimum z biblioteki heapq. Wprowadzam do struktury 3 informacje: współrzędną x-ową zdarzenia, informację czy jest to początek odcinka lub jego koniec lub przecięcie odcinków, listę odcinków biorących udział w zdarzeniu. Kluczem do porównywania wartości w kopcu jest współrzędna x-owa zdarzenia. Struktura stanu pozostała taka sama.

3. Zrobione na laboratorium

- Generowanie odcinków
- Algorytm dla szukanie czy istnieją odcinki przecinające się
- Wizualizacja algorytmu, sprawdzone kilka przypadków - działało
- Drugi algorytm zaimplementowany jak na razie poza obsługą zdarzenia przecięcia