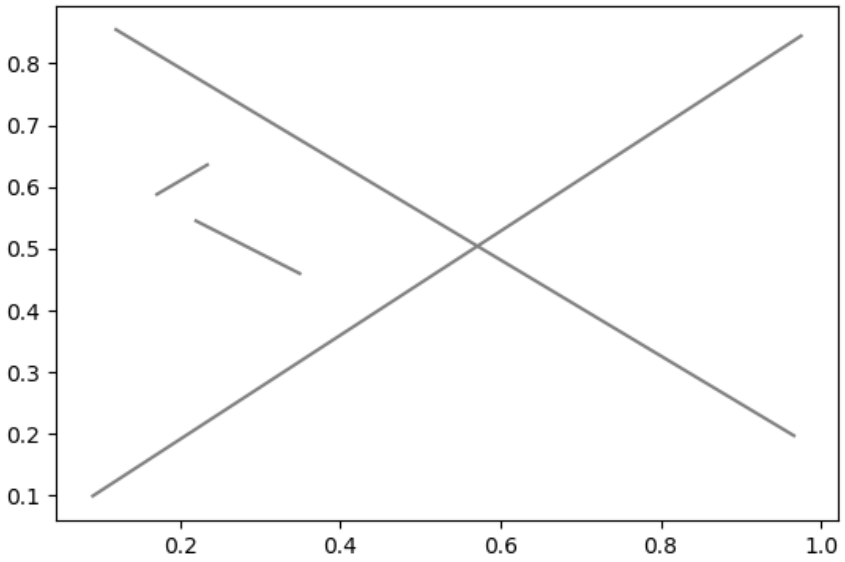
Struktura zdarzeń w mojej implementacji realizowana jest przy pomocy kolejki priorytetowej. Każde zdarzenie składa się z

- Punktu w którym zaszło – który jest jednocześnie priorytetem w kolejce
- Typu zdarzenia – określającego czy zdarzenie jest początkiem, końcem czy przecięciem
- Odcinków biorących udział w zdarzeniu

Stan miotły jest realizowany z użyciem SortedSet dostarczanego przez bibliotekę sortedcontainers. W stanie miotły przechowywane są klucze będące reprezentantami odcinków.
Porównywanie odcinków w stanie polega na sprawdzeniu czy początek odcinka leży nad czy pod prosta definiującą drugi odcinek. Wiemy że do punktu przecięcia względny porządek prostych będzie zachowany.
W przypadku przecięcia 2 odcinków usuwamy oba i wrzucamy odcinki ze zmodyfikowanym punktem początkowym. Możemy to interpretować jako podzielenie odcinka na połowę i zachowanie tyko interesującej nas części.

W przypadku obu zadań struktura zdarzeń może pozostać taka sama. Zauważyć natomiast trzeba że w przypadku pierwszego zadania zdarzenia zawierały tylko 1 odcinek co jest potencjalnym miejscem na mikro optymalizacje, gdy w drugim przy zdarzeniu przecinania się lini trzeba było pamiętać o 2 odcinkach. Jednak podstawowa budowa struktury zdarzeń powinna pozostać taka sama, ponieważ zdarzenia muszą być ułożone względem x, oraz potrzebujemy przechowywać informacje o jakiego typu jest wydarzenie.



Przy wykryciu punktu przecięcia sprawdzam czy podane odcinki już się nie przecinają. Jako że proste mogą przecinać się maksymalnie w 1 punkcie (nie licząc prostych pokrywających się) nie dodaje duplikatu.