

Do implementacji struktury stanu użyłam SortedList z biblioteki Pyhtona. Porównanie elementów stanu miotły polega na sprawdzeniu orientacji początku odcinka względem drugiego odcinka (powyżej lub poniżej). W przypadku przecięcia zmieniamy oba punkty początkowe odcinków na punkt w którym zaszło to przecięcie. Porządek zostaje zachowany

Aby wyznaczyć wszystkie przecięcia jako strukturę użyłam SortedSet z gdzie potrzebne operacje usuwania, dodawania nowego elementu zajmują czas  $O(\log n)$ . Struktura ta przechowuje informacje : punkt zdarzenia, odcinek lub odcinki w przypadku przecięcia oraz typ zdarzenia. Zdarzenia są posortowane względem współrzędnej  $x$ .

W przypadku algorytmu wykrywającego czy istnieje przecięcie, jako struktura stanu wystarczy zwykła lista przechowująca odcinki posortowane względem współrzędnej  $x$ , oraz typ zdarzenia, ponieważ nie wprowadzamy tam modyfikacji tylko przy pierwszym wykrytym przecięciu kończymy działanie algorytmu. Jednak warto pozostawić taką samą jak w przypadku pierwszym

Dzięki przechowywaniu przecięć w strukturze Set w szybki sposób  $O(1)$  sposób można sprawdzić czy właśnie wykryte przecięcie znajduje się już w tym zbiorze.