Adam Dyda grupa czwartek 16:15 A

Algorytmy geometryczne laboratorium 4

1. Cel ćwiczenia

Zapoznanie się z algorytmem detekcji i wyznaczania przecięć odcinków na płaszczyźnie.

2. Sposób implementacji struktury stanu i struktury zdarzeń.

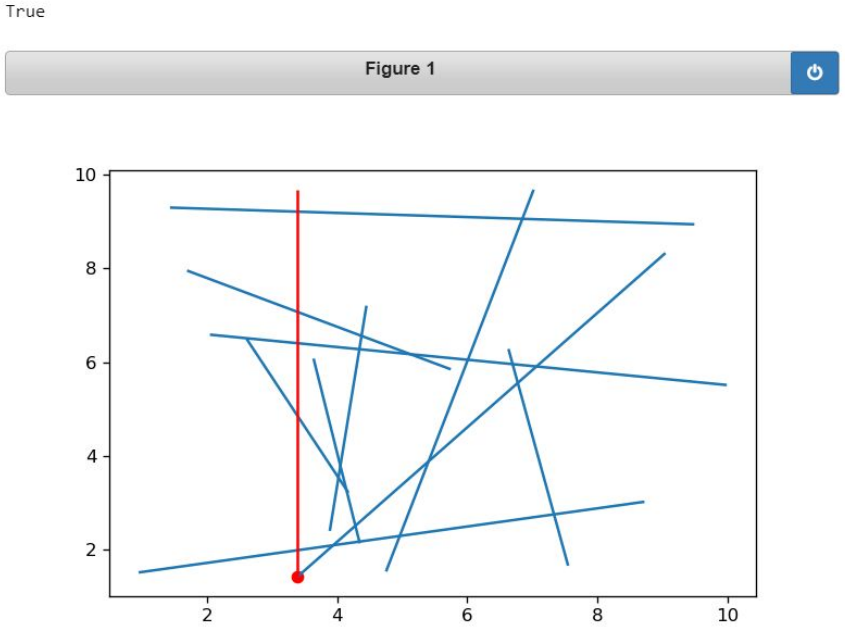
Zarówno struktura stanu jak i struktura zdarzeń zostały zaimplementowane za pomocą struktury sortedset w której elementy są zawsze posortowane, zapewnia ona złożoność dodawania i usuwania O(log n)

3. Algorytm wykrywania przecięcia.

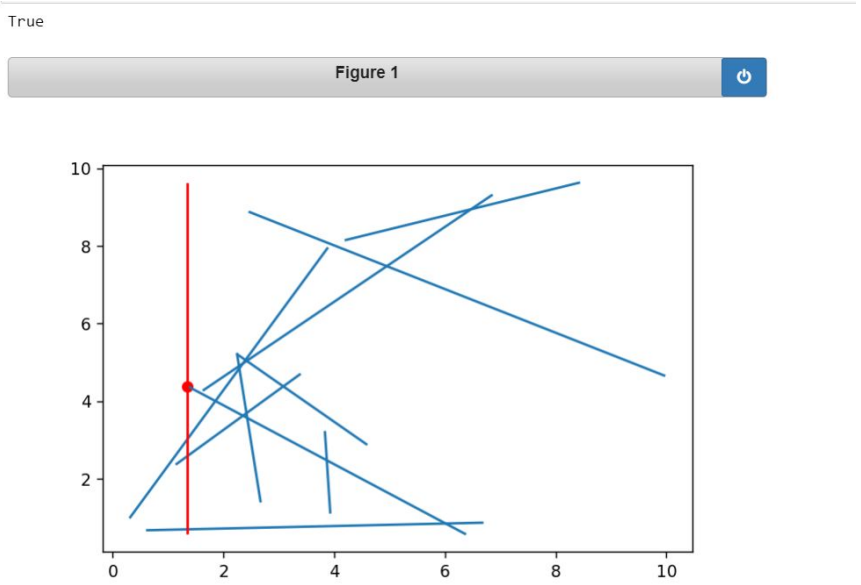
Warto dodać że w generowanych zbiorach żaden odcinek nie jest pionowy oraz żadna para odcinków nie posiada tej samej współrzędnej x.

Pierwszym zaimplementowanym algorytmem jest algorytm wykrywający czy w danym zbiorze odcinków znajduję się przecięcie.
Prezentacja wyniku algorytmu na dwóch wygenerowanych zbiorach:

rys 1.1 algorytm wykrywania przecięcia w zbiorze pierwszym



rys 1.2 algorytm wykrywania przecięcia w zbiorze drugim

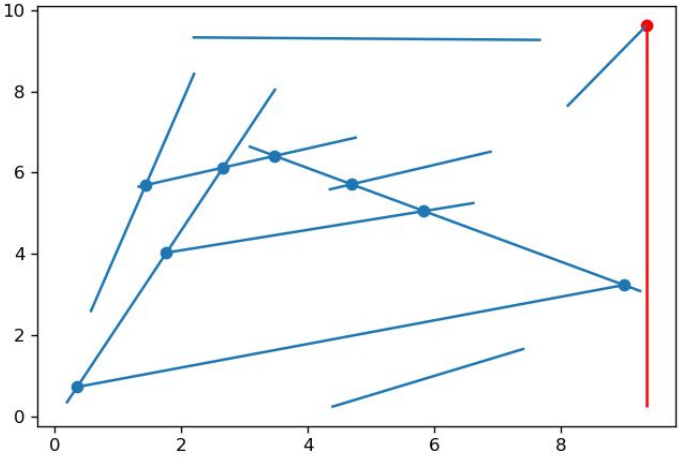


4. Algorytm znajdowania wszystkich punktów przecięcia.

Drugim zaimplementowanym algorytmem jest algorytm znajdowania wszystkich punktów przecięcia w danych zbiorze odcinków.

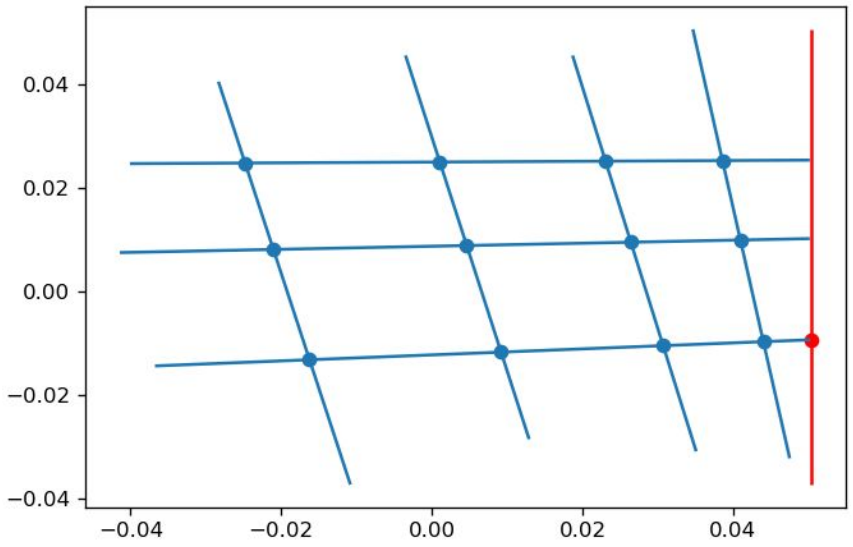
W przypadku drugiego algorytmu jako implementacji struktury zdarzeń nadal używam struktury sortedset lecz w tym wypadku musiałem dodać typ zdarzenia “przecięcie”.
Prezentacja wyników dla trzech zbiorów odcinków:

rys 2.1 algorytm znajdowania punktów przecięcia w zbiorze pierwszym



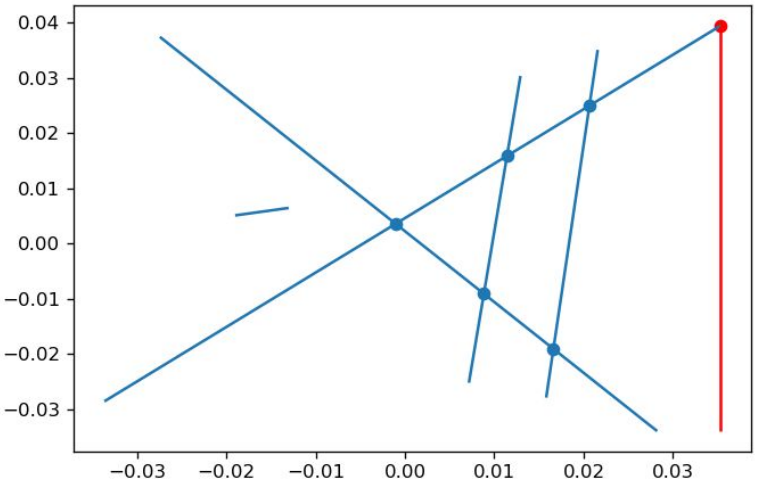
```
Number of intersections 8
Point (4.694616713562629, 5.714733737371759)
Line ((4.332913449487173, 5.583037099440556), (6.913572410523816, 6.522658448128896)) ((3.066733330624026, 6.65066954782031
6), (9.28048714083875, 3.0781318775893816))
Point (0.3524453608447044, 0.7281480469853958)
Line ((0.18064482345582, 0.326583140731902), (3.493325666113215, 8.069611144295818)) ((0.30427883021220126, 0.71418203150234
84), (9.097282631514163, 3.2637372022583833))
Point (9.004469230441213, 3.236825706259035)
Line ((0.30427883021220126, 0.7141820315023484), (9.097282631514163, 3.2637372022583833)) ((3.066733330624026, 6.65066954782
0316), (9.28048714083875, 3.0781318775893816))
Point (2.6614668779970656, 6.125232498320644)
Line ((0.18064482345582, 0.326583140731902), (3.493325666113215, 8.069611144295818)) ((1.3046819751320904, 5.64861852886636
1), (4.778594500705944, 6.868941097944563))
Point (5.840770724207635, 5.055763571514147)
Line ((1.756053065452533, 4.029033112711687), (6.640622388923925, 5.2568134690334665)) ((3.066733330624026, 6.65066954782031
6), (9.28048714083875, 3.0781318775893816))
Point (1.7656914920285738, 4.031455817771691)
Line ((0.18064482345582, 0.326583140731902), (3.493325666113215, 8.069611144295818)) ((1.756053065452533, 4.02903311271168
7), (6.640622388923925, 5.2568134690334665))
Point (1.4372675833387327, 5.695193452339082)
Line ((0.5629783643813402, 2.5702375367343366), (2.210149582388979, 8.45769140224711)) ((1.3046819751320904, 5.6486185288663
61), (4.778594500705944, 6.868941097944563))
Point (3.480321207095773, 6.412881197368261)
Line ((1.3046819751320904, 5.648618528866361), (4.778594500705944, 6.868941097944563)) ((3.066733330624026, 6.65066954782031
6), (9.28048714083875, 3.0781318775893816))
```

rys 2.2 algorytm znajdowania punktów przecięcia w zbiorze drugim



```
Number of intersections 12
Point (-0.021030416442252873, 0.008067884399318553)
Line ((-0.0414128654233871, 0.007467916760886081), (0.05017987651209678, 0.01016399519225862)) ((-0.028328188004032265, 0.04
04948775451998), (-0.01080802671370968, -0.03735438716068256))
Point (0.02642904350159598, 0.00946487747557484)
Line ((-0.0414128654233871, 0.007467916760886081), (0.05017987651209678, 0.01016399519225862)) ((0.01868794102822581, 0.0455
5002460402334), (0.03509923135080646, -0.03095120088617276))
Point (0.044020411345568354, -0.009752687228341553)
Line ((0.03465568296370969, 0.05060517166284688), (0.04751858618951614, -0.03229924010185903)) ((-0.03675560735887097, -0.01
443720494015885), (0.05040165070564517, -0.00938257343519236))
Point (0.03071718189708388, -0.010524278518760314)
Line ((0.01868794102822581, 0.0455500246040233 4), (0.03509923135080646, -0.03095120088617276)) ((-0.03675560735887097, -0.01
443720494015885), (0.05040165070564517, -0.00938257343519236))
Point (0.001067505597585965, 0.024962696594592826)
Line ((-0.040082220262096784, 0.024655416760886076), (0.05017987651209678, 0.025329436368729225)) ((-0.003489478326612907 6,
0.04555002460402334), (0.012921811995967744, -0.028592132258721772))
Point (0.009198771073866713, -0.011772352788180555)
Line ((-0.003489478326612907 6, 0.04555002460402334), (0.012921811995967744, -0.028592132258721772)) ((-0.03675560735887097,
-0.014437720494015885), (0.05040165070564517, -0.00938257343519236))
Point (-0.024789180536808415, 0.02476961540025688)
Line ((-0.040082220262096784, 0.024655416760886076), (0.05017987651209678, 0.025329436368729225)) ((-0.028328188004032265,
0.0404948775451998), (-0.01080802671370968, -0.03735438716068256))
Point (0.040972320984495283, 0.00989296616376729)
Line ((-0.0414128654233871, 0.007467916760886081), (0.05017987651209678, 0.01016399519225862)) ((0.03465568296370969, 0.0506
0517166284688), (0.04751858618951614, -0.03229924010185903))
Point (0.023069156284058957, 0.02512699084985145)
Line ((-0.040082220262096784, 0.024655416760886076), (0.05017987651209678, 0.025329436368729225)) ((0.01868794102822581, 0.0
4555002460402334), (0.03509923135080646, -0.03095120088617276))
Point (-0.016233355586760 06, -0.013247423765166464)
Line ((-0.028328188004032265, 0.0404948775451998), (-0.01080802671370968, -0.03735438716068256)) ((-0.03675560735887097, -0.
014437720494015885), (0.05040165070564517, -0.00938257343519236))
Point (0.03859072683349315, 0.025242896011250616)
Line ((-0.040082220262096784, 0.024655416760886076), (0.05017987651209678, 0.025329436368729225)) ((0.03465568296370969, 0.0
5060517166284688), (0.04751858618951614, -0.03229924010185903))
Point (0.004639899520817724, 0.008823503062239491)
Line ((-0.0414128654233871, 0.007467916760886081), (0.05017987651209678, 0.01016399519225862)) ((-0.003489478326612907 6, 0.0
4555002460402334), (0.012921811995967744, -0.028592132258721772))
```

rys 2.3 algorytm znajdowania punktów przecięcia w zbiorze trzecim




```
Number of intersections 5
Point (0.008828495643939852, -0.009089986287508336)
Line ((0.007155682963709678, -0.02522203164832533), (0.012921811995967744, 0.030384585998733504)) ((-0.02744109122983871, 0.
03746179188108645), (0.028224231350806456, -0.03398428655028612))
Point (0.020646659257187644, 0.025000092358635005)
Line ((-0.03365076864919355, -0.028592129687541018), (0.03532100554435484, 0.03948385070461587)) ((0.015804876512096777, -0.
027918110079697883), (0.021571005544354843, 0.03510272325363546))
Point (0.01661335928399378, -0.019081807989413353)
Line ((0.015804876512096777, -0.027918110079697883), (0.021571005544354843, 0.03510272325363546)) ((-0.02744109122983871, 0.
03746179188108645), (0.028224231350806456, -0.03398428655028612))
Point (0.011419035819160804, 0.015892315741630386)
Line ((-0.03365076864919355, -0.028592129687541018), (0.03532100554435484, 0.03948385070461587)) ((0.007155682963709678, -0.
02522203164832533), (0.012921811995967744, 0.030384585998733504))
Point (-0.0010483383558081757, 0.0035868657252922324)
Line ((-0.03365076864919355, -0.028592129687541018), (0.03532100554435484, 0.03948385070461587)) ((-0.02744109122983871, 0.0
3746179188108645), (0.028224231350806456, -0.03398428655028612))
```

5. Opis obsługi zdarzeń

W trakcie działania algorytmu wyjmując kolejne zdarzenia z struktury zdarzeń musimy je odpowiednio obsługiwać, jeżeli zdarzenie jest:

1. Początkiem odcinka, wtedy wstawiamy ten odcinek do struktury stanu miotły, sprawdzając czy nie tworzy on z sąsiadami przecięcia i czy takie przecięcie nie było już rozpatrywane.
2. Końcem odcinka, wtedy usuwamy ten odcinek z struktury stanu, sprawdzając czy sąsiedzi tego odcinka nie tworzą przecięcia i czy takie przecięcie nie było już rozpatrywane.
3. Przecięciem odcinków, wtedy oba odcinki są dodawane do zbioru końcowego oraz ich kolejność w strukturze stanu jest zamieniana.

6. Podsumowanie

Algorytmy działają poprawnie i spełniają swoje założenia. Algorytm znajdowania punktów przecięcia działa w czasie $O((P+n) \log n)$ gdzie P - liczba przecięć. Uzyskanie takiego czasu jest możliwe przy użyciu odpowiednich struktur danych do przechowywania struktury stanu i zdarzeń. Użytą przeze mnie implementacja sortedset jest zbalansowane drzewo binarne.