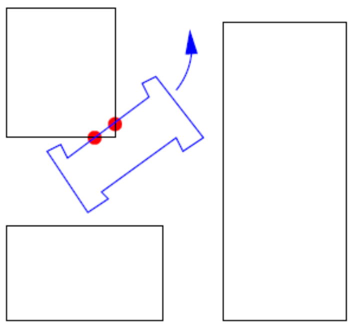
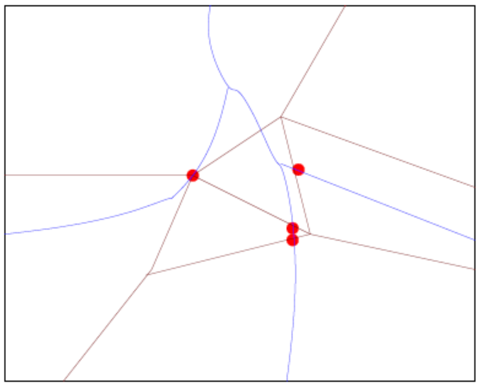
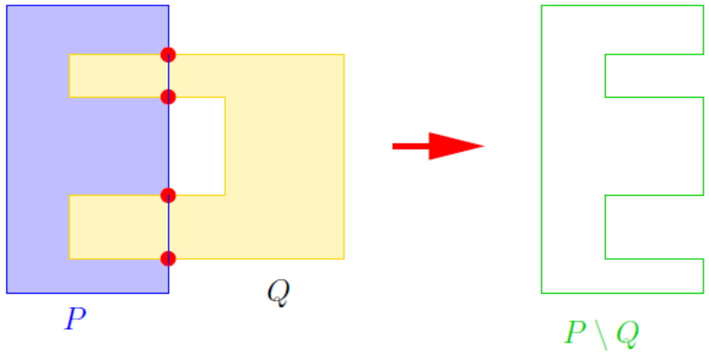
Problem przecinania się odcinków



Planowanie ruchu robota



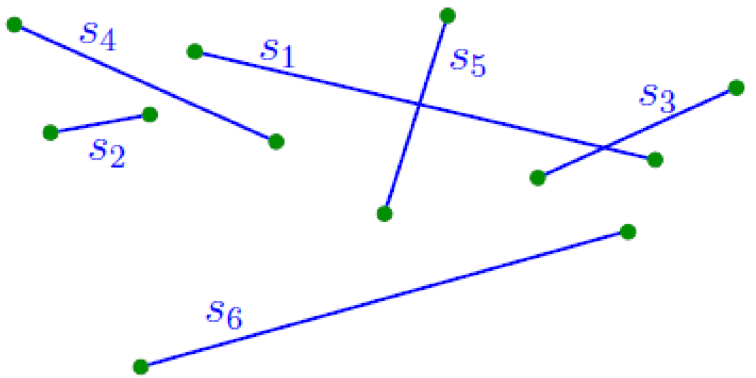
Nakładanie się map



CAD – operacje boolowskie

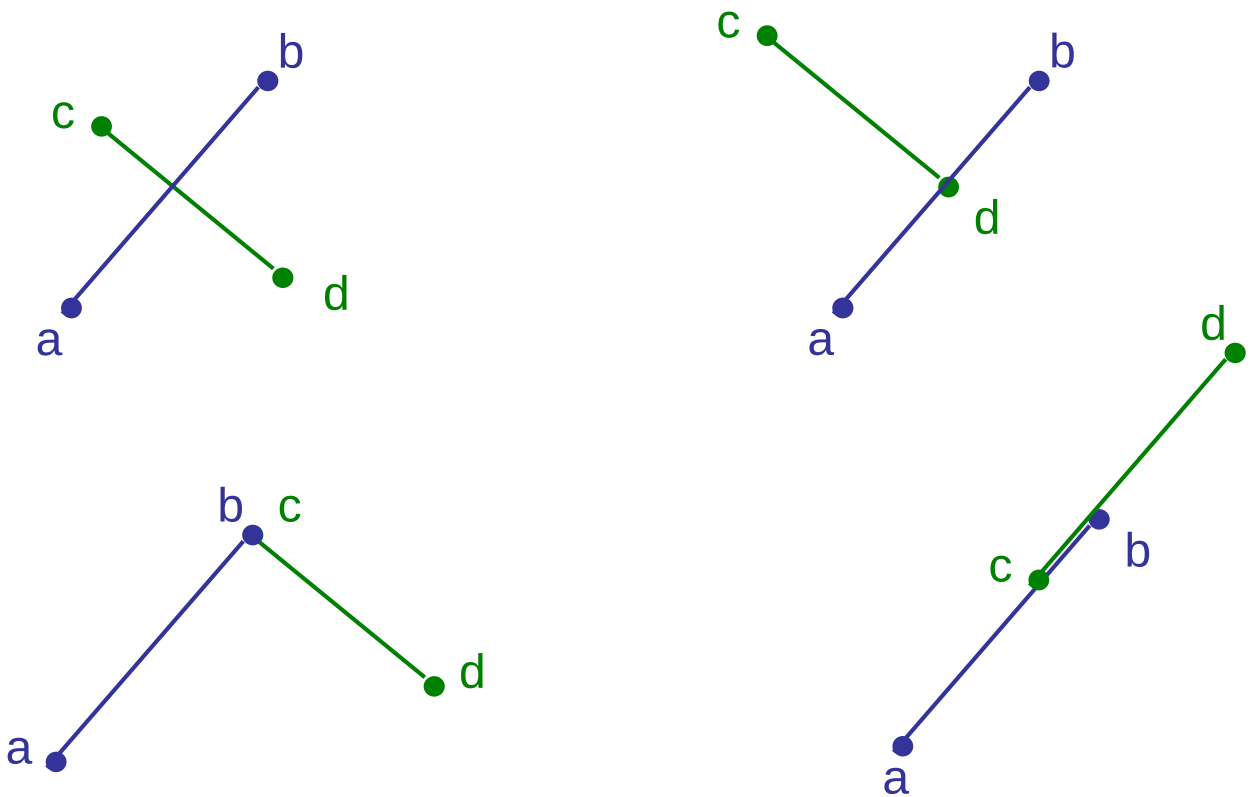
Zagadnienia

Zadany zbiór odcinków $S = \{s_1, s_2, \ldots , s_n\}$ w R^2



- Czy istnieje para (s_i, s_j) taka, że $i \neq j$ i $s_i \cap s_j \neq \emptyset$?
- Znaleźć wszystkie pary (s_i, s_j) takie, że $i \neq j$ i $s_i \cap s_j \neq \emptyset$.
- Znaleźć wszystkie pary (s_i, s_j) takie, że $i \neq j$ i $s_i \cap s_j \neq \emptyset$ wraz z punktami przecięć.

Czy dwa odcinki się przecinają?



$$s_1(t) = (1-t)a + tb \qquad 0 \le t \le 1$$

$$s_2(r) = (1-r)c + rd \qquad 0 \le r \le 1$$

Pierwsze podejście – algorytm „brutalny”

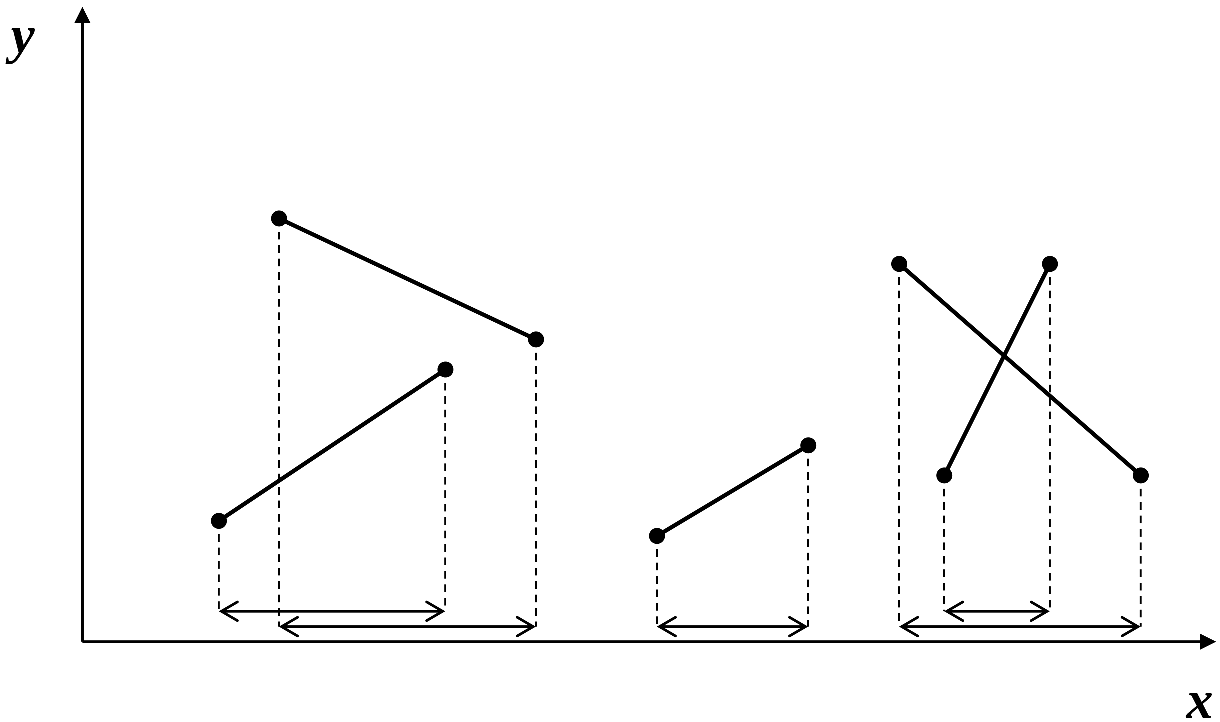
- Sprawdzamy wszystkie pary odcinków
- Wymaga czasu $O(n^2)$

Czy można lepiej?

Gdy każda para odcinków się przecina – $\Omega(n^2)$

W praktyce liczba przecięć jest znacznie mniejsza niż kwadratowa.

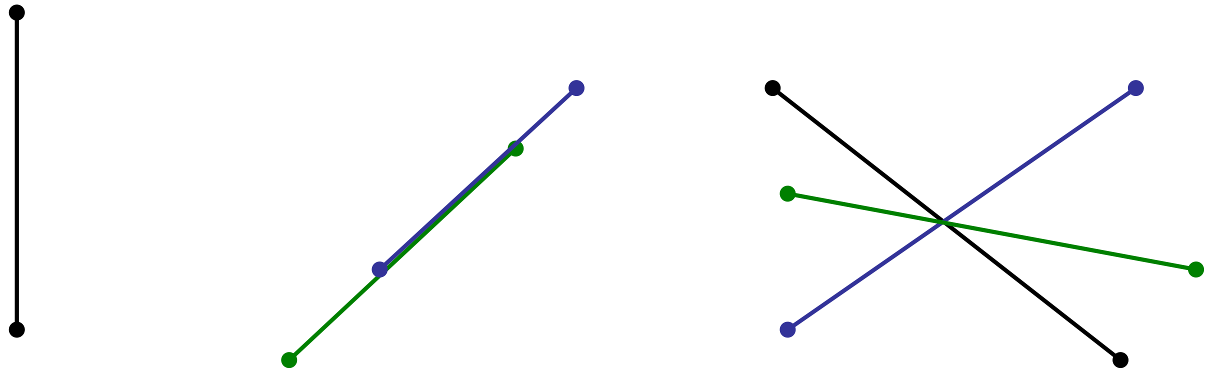
Poszukiwany algorytm, którego czas zależy nie tylko od liczby odcinków, ale też od liczby przecięć – im mniejsza liczba przecięć tym czas krótszy $\rightarrow$ *algorytm wrażliwy na przecięcia*



Sprawdzać tylko te pary odcinków,
których *x*-przedziały na siebie nachodzą

Założenia

- Żaden odcinek nie jest pionowy
- Dwa odcinki przecinają się w co najwyżej jednym punkcie
- Żadne trzy odcinki nie przecinają się w jednym punkcie



Przypadki wykluczone

Algorytm zamiatania (*sweeping*)

- Ustalamy pewną hiperpłaszczyznę (np. prostą w R2, płaszczyznę w R3). Nazywamy ją ***miotłą***.
- Przesuwamy miotłę w wyznaczonym kierunku (kierunku zamiatania).
- Pozycje, w których miotła zatrzymuje się nazywamy ***zdarzeniami***. Informacje o nich przechowujemy w ***strukturze zdarzeń***.
- Informacje potrzebne do obliczeń przechowujemy w ***strukturze stanu***. Struktura stanu jest aktualizowana w każdym zdarzeniu.

Na „zamiecionym” obszarze znane jest rozwiązanie badanego problemu dla zdarzeń należących do tego obszaru.

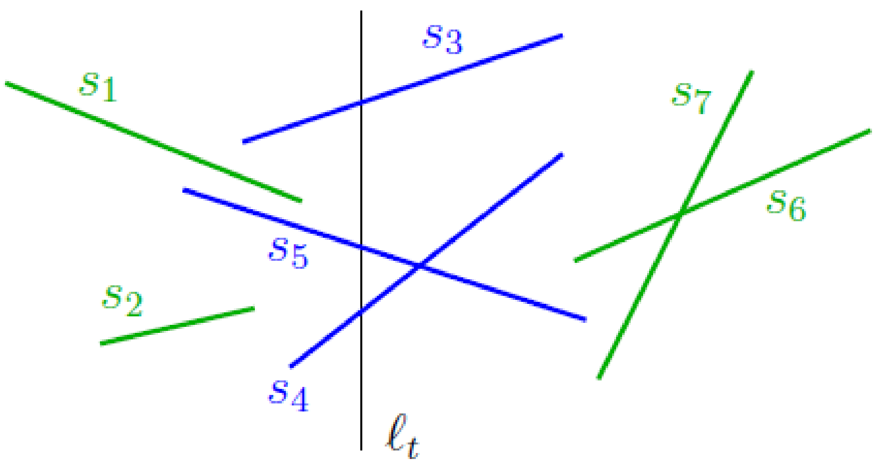
Struktura zdarzeń jak i stanu może być różnie zaimplementowana.
Ale: powinny zapewniać łatwość i efektywność pewnych operacji
(np. włączanie elementu, usuwanie …)

Metoda zamiatania

Miotła będzie zamiatać wzdłuż osi *x*-ów.

W każdym położeniu miotły:

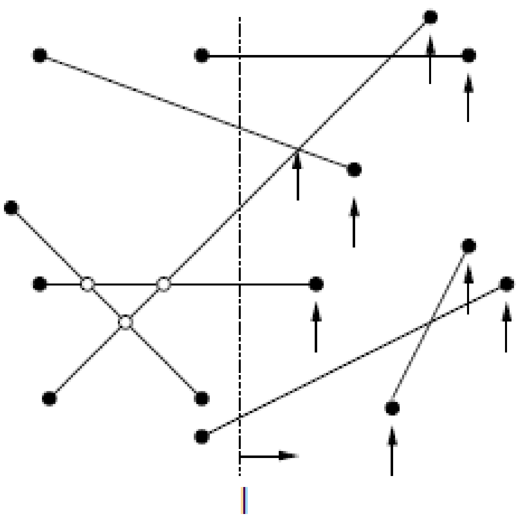
- odcinki ***przetworzone*** – odcinki, których końce znajdują się na lewo od miotły,
- odcinki ***aktywne*** – odcinki aktualnie przecinające miotłę,
- odcinki ***oczekujące*** – odcinki o obu końcach na prawo od miotły.



Miotła

Stan miotły – zbiór odcinków przecinających ją

Miotła zatrzymuje się w **punktach zdarzeń**, którymi są końce odcinków i punkty przecięć.



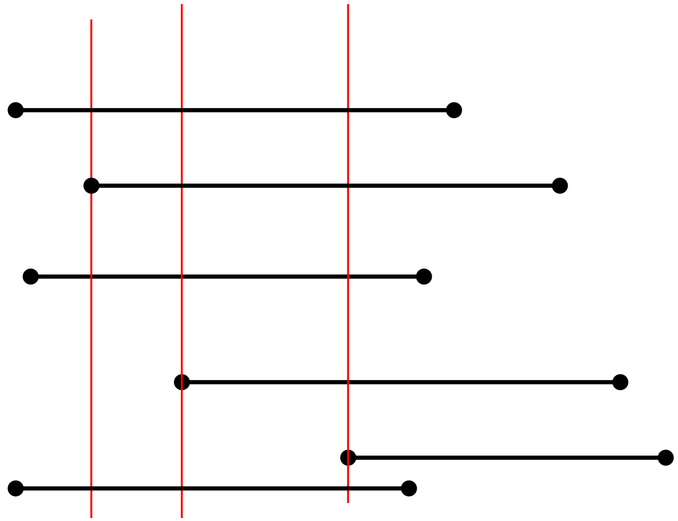
∘ wykryte przecięcia

↑ przyszłe zdarzenia

Wszystkie przecięcia na lewo od miotły są znane.
Przecięcia na prawo od miotły są nieznane.

W punktach zdarzeń – aktualizacja stanu miotły, testy przecięć

Wciąż za dużo testów →



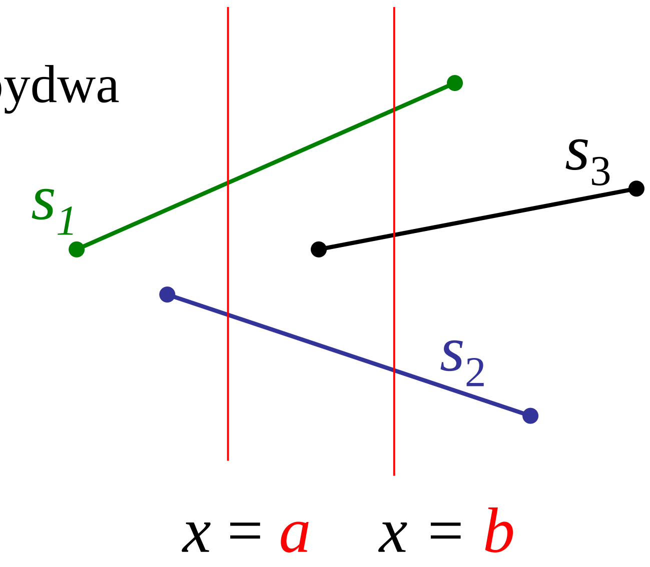
Uporządkowanie odcinków

Żaden z odcinków nie jest pionowy →
każdy z odcinków może przecinać miotłę w co najwyżej jednym punkcie

Dla każdego położenia miotły możemy uporządkować odcinki przecinające ją zgodnie z kolejnością przecięć – ze względu na współrzędne *y*

dwa odcinki są ***porównywalne*** w $x=x_0$,
jeżeli miotła umieszczona w $x=x_0$ przecina je obydwa

Mówimy, że s_i jest **powyżej** s_k w x
i oznaczamy $s_i >_x s_k$,
jeżeli s_i i s_k są porównywalne w x
oraz przecięcie s_i z miotłą w x jest wyżej niż przecięcie s_k z tą miotłą.

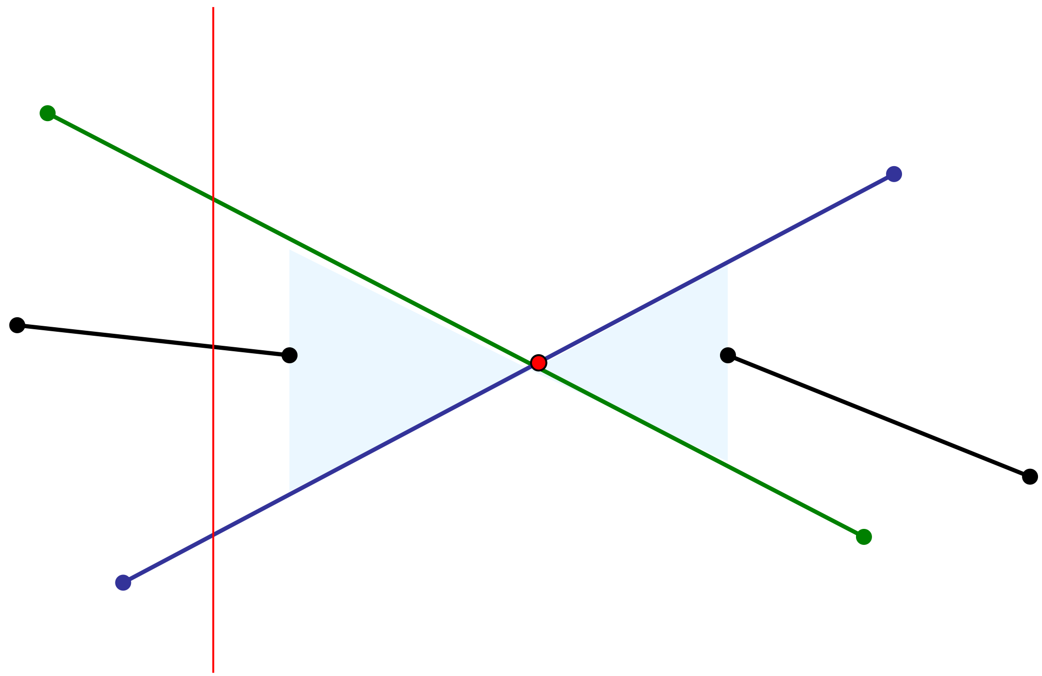


$s_1 >_a s_2 \quad s_1 >_b s_2 \quad s_1 >_b s_3 \quad s_3 >_b s_2$

Porządek ten może być różny dla różnych *x*, ponieważ:

• dla różnych położeń miotły różne odcinki ją przecinają,

• jeżeli odcinki się przecinają, to ich kolejność po obydwu stronach przecięcia jest różna.



Stan miotły - uporządkowany ciąg odcinków przecinających miotłę

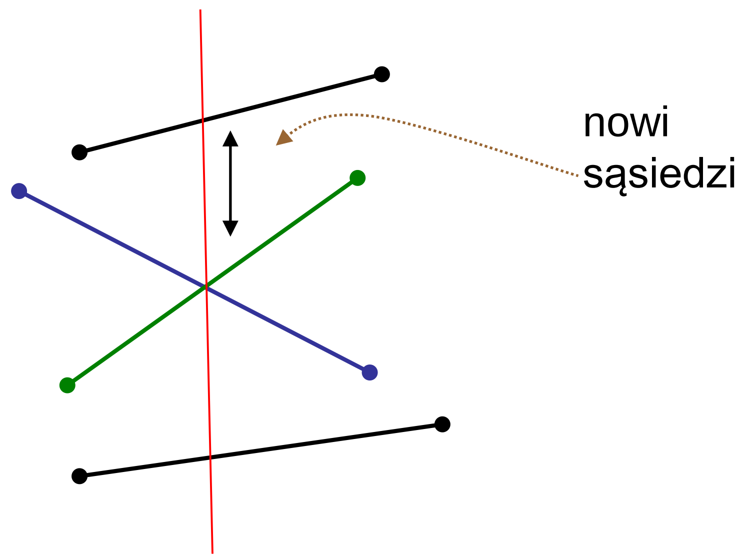
Sprawdzamy odcinki tylko wtedy,
gdy **sąsiadują ze sobą** w porządku pionowym

W punkcie przecięcia:

- zmienia się porządek odcinków
- zmieniają się sąsiedzi

$\Downarrow$

Każdy odcinek zyskuje co najwyżej jednego nowego sąsiada



Korzystamy z dwóch **struktur danych**:

Struktura zdarzeń Q

zawiera uporządkowane rosnąco względem x-ów końce odcinków oraz punkty przecięć wszystkich par odcinków aktywnych, które kiedykolwiek były sąsiadami w strukturze.

(np. zrównoważone drzewo poszukiwań binarnych)

Struktura stanu T

zbiór odcinków aktywnych
uporządkowanych względem współrzędnych y-owych.

(np. wzbogacone, zrównoważone drzewo poszukiwań)

Po dojściu miotły do kolejnego punktu zdarzenia mamy trzy możliwości.

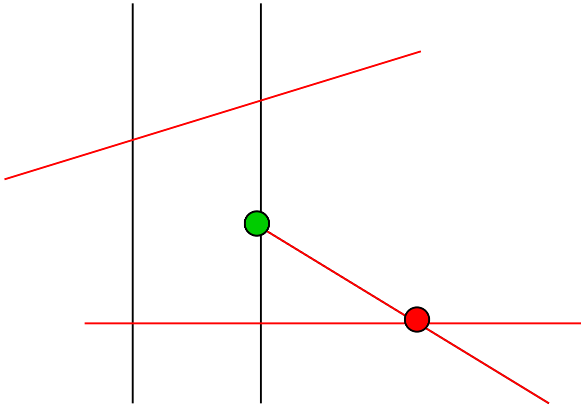
Zdarzenie jest początkiem odcinka *s*.

wstaw *s* do T; uaktualnij dowiązania;

if *s’* jest sąsiadem *s* w T

then if *s*’ przecina *s* w punkcie *p*

then wstaw *p* do Q;



Zdarzenie jest końcem odcinka *s*.

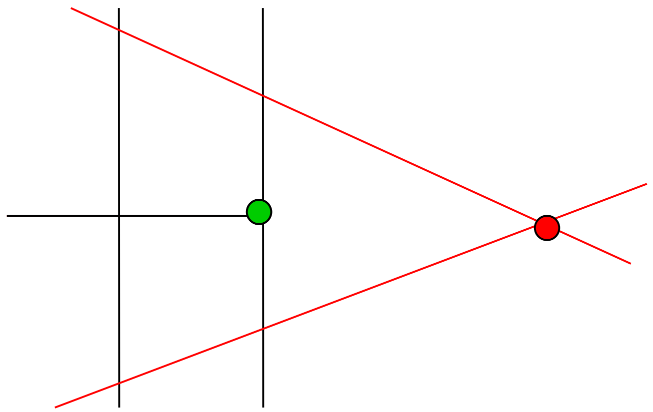
usuń *s* z T; uaktualnij dowiązania;

if s miał w T dwóch sąsiadów *s*’, *s*’’

    then if *s*’ przecina *s*’’ w punkcie *p*

        then wstaw *p* do Q, jeśli go tam

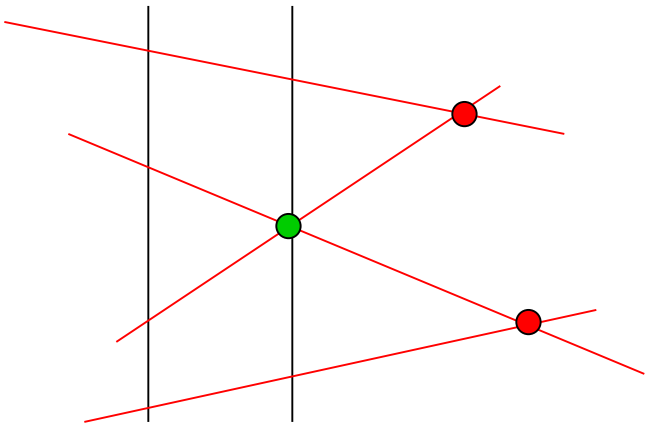
            jeszcze nie ma;

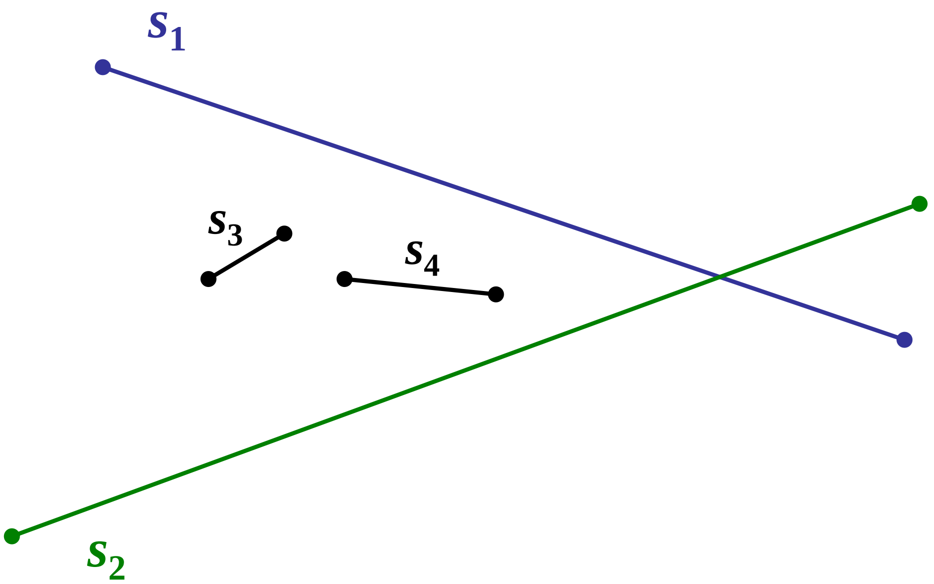


Zdarzenie jest punktem przecięcia odcinków *s*, *s*’.

Zamień kolejność *s* i *s*’ w T;

if *w* jest sąsiadem *s*
  then if *w* przecina *s* w punkcie *p*
    then wstaw *p* do Q , jeśli go tam
        jeszcze nie ma;
if *v* jest sąsiadem *s*’
  then if *v* przecina *s*’ w punkcie *q*
    then wstaw *q* do Q , jeśli go tam
        jeszcze nie ma;





Punkt przecięcia s_1 i s_2 wykrywany trzy razy:

- gdy dodajemy lewy koniec s_1
- gdy usuwamy prawy koniec s_3
- gdy usuwamy prawy koniec s_4

Utwórz pustą strukturę stanu T

Utwórz strukturę zdarzeń Q – wstaw posortowane wzdłuż *x* końce odcinków

Powtarzaj

- pobierz nowe zdarzenie *p* z Q
- zaktualizuj T:
  - jeśli *p* jest lewym końcem odcinka – dodaj odcinek do T
  - jeśli *p* jest prawym końcem odcinka – usuń odcinek z T
  - jeśli *p* jest punktem przecięcia *s* i *s*', zmień porządek *s* i *s*' w T
- zaktualizuj Q:
  - dla każdej pary *s* i *s*' z T sprawdź, czy *s* i *s*' przecinają się po prawej stronie miotły
    jeśli tak – dodaj punkt przecięcia do Q
  - usuń możliwe duplikaty z Q

aż Q będzie puste

Złożoność przy odpowiednich strukturach:

- inicjowanie Q – $O(n \log n)$
- aktualizacja T – $O((P+n) \log n)$
- aktualizacja Q – $O(P \log n)$

gdzie P – liczba przecięć

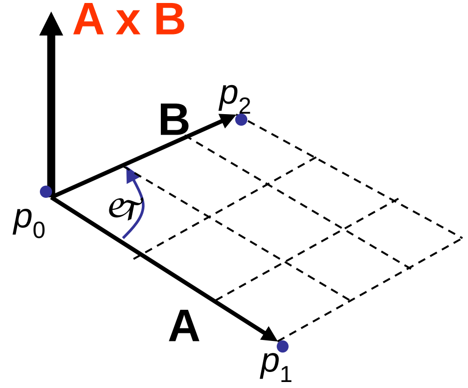
Całość: $O((P+n) \log n)$

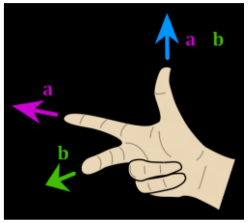
Dodatkowe uwagi

Iloczyn wektorowy

A x B – wektor prostopadły do A i B

Długość: |A||B| sin θ





Długość wektora otrzymanego
jako iloczyn wektorowy dwóch wektorów
jest równy polu równoległoboku
rozpiętego na tych wektorach:

$$\mathrm{A x B} = \begin{vmatrix} \vec{\mathrm{i}} & \vec{\mathrm{j}} & \vec{\mathrm{k}} \\ A_x & A_y & A_z \\ B_x & B_y & B_z \end{vmatrix}$$

Dla 2-wymiarowych wektorów $A_z = B_z = 0$, obliczenia redukują się:

$$(A_x B_y - A_y B_x)\vec{\mathrm{k}}$$

Pole równoległoboku rozpiętego na wektorach (p_0,p_1) i (p_0,p_2) =

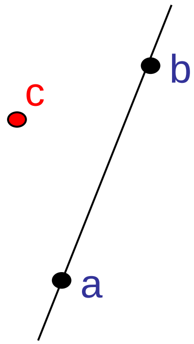
$$(p_{1x} - p_{0x})(p_{2y} - p_{0y}) - (p_{2x} - p_{0x})(p_{1y} - p_{0y})$$

lub analogicznie:

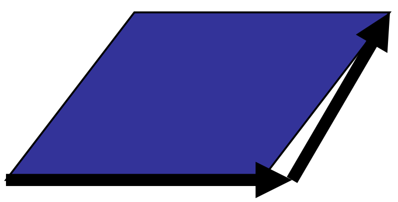
$$\det(p_0, p_1, p_2) = \begin{vmatrix} p_{0x} & p_{0y} & 1 \\ p_{1x} & p_{1y} & 1 \\ p_{2x} & p_{2y} & 1 \end{vmatrix}$$

Po której stronie (a,b) znajduje się c ?

$$\det(a,b,c)\begin{cases} <0 \\ >0 \\ =0 \end{cases}$$



<0



>0

Na płaszczyźnie

Krzywa parametryczna:

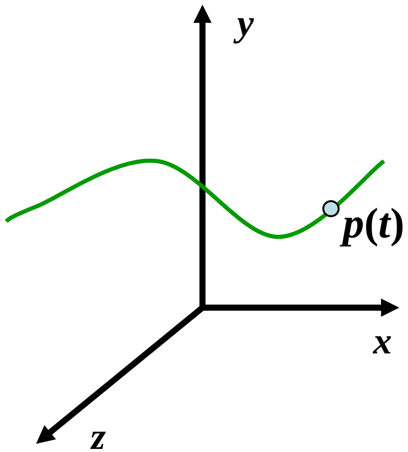
$$C(t) = (\ x(t), y(t)) \qquad \text{dla } t \in [t_0, t_1]$$

$x(t), y(t)$ – funkcje współrzędnych

- łatwe do rysowania
- trudno sprawdzić, czy punkt leży na krzywej
- mogą przedstawiać krzywe zamknięte

W przestrzeni

$$C(t) = (\ x(t), y(t), z(t)) \qquad \text{dla } t \in [t_0, t_1]$$

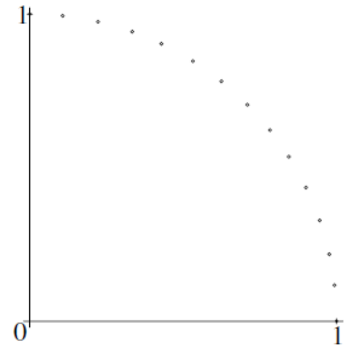


Postać parametryczna

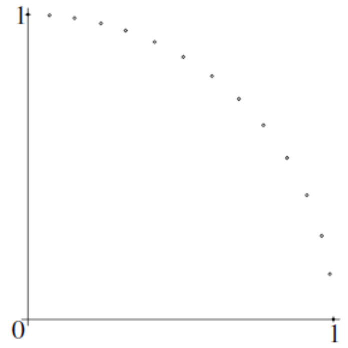
Dana krzywa może mieć różne parametryzacje.

Np. ćwiartka okręgu

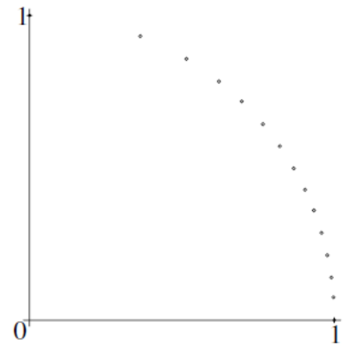
$$C(t) = (\cos\frac{\pi}{2}t, \sin\frac{\pi}{2}t), \quad t \in [0,1]$$



$$C(t) = (\frac{1-t^2}{1+t^2}, \frac{2t}{1+t^2}) \quad t \in [0,1]$$



$$C(t) = (\sqrt{1-t^2}, t) \quad t \in [0,1]$$



Ciągłość krzywych parametrycznych

Niech $\mathbf{C}(t) = (x(t), y(t))$ dla $t \in (t_0, t_1)$

Krzywa ma **ciągłość** (***parametryczną***) klasy C^k,
jeżeli istnieją ciągłe pochodne rzędu od 0 do k funkcji $x(t)$ oraz $y(t)$.

Jeżeli krzywa jest C^1 na przedziale I, to funkcja $v(t) = \sqrt{(x'(t))^2 + (y'(t))^2}$
jest nazywana ***prędkością*** krzywej $C(t)$.

Jeżeli $v(t) \neq 0$ dla wszystkich $t \in I$, to krzywa jest nazywana ***regularną***.

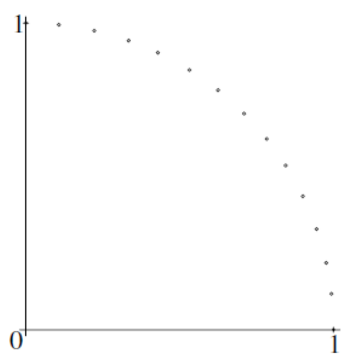
$$L(C) = \int_a^b \sqrt{(x'(t))^2 + (y(t))^2}\, dt = \int_a^b v(t)\, dt$$ ***długość*** łuku krzywej

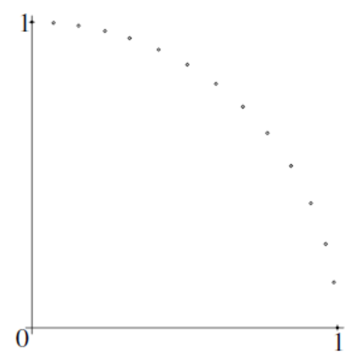
Prędkość krzywej dla trzech różnych parametryzacji ćwiartki okręgu

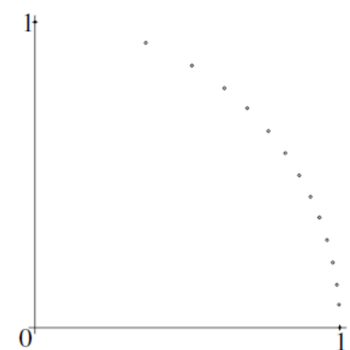
$$C(t) = (\cos\frac{\pi}{2}t, \sin\frac{\pi}{2}t), \quad t \in [0,1] \qquad v(t) = \sqrt{\left(-\frac{\pi}{2}\sin\frac{\pi}{2}t\right)^2 + \left(\frac{\pi}{2}\cos\frac{\pi}{2}t\right)^2} = \frac{\pi}{2}$$

$$C(t) = (\frac{1-t^2}{1+t^2}, \frac{2t}{1+t^2}) \quad t \in [0,1] \qquad v(t) = \sqrt{\left(\frac{-4t^2}{(1+t^2)^2}\right)^2 + \left(\frac{2(1-t^2)}{(1+t^2)^2}\right)^2} = \frac{2}{1+t^2}$$

$$C(t) = (\sqrt{1-t^2}, t) \quad t \in [0,1] \qquad v(t) = \sqrt{(-t(1-t^2)^{-1/2})^2 + 1} = (1-t^2)^{-1/2}$$







Funkcja długości łuku dla trzech parametryzacji ćwiartki okręgu