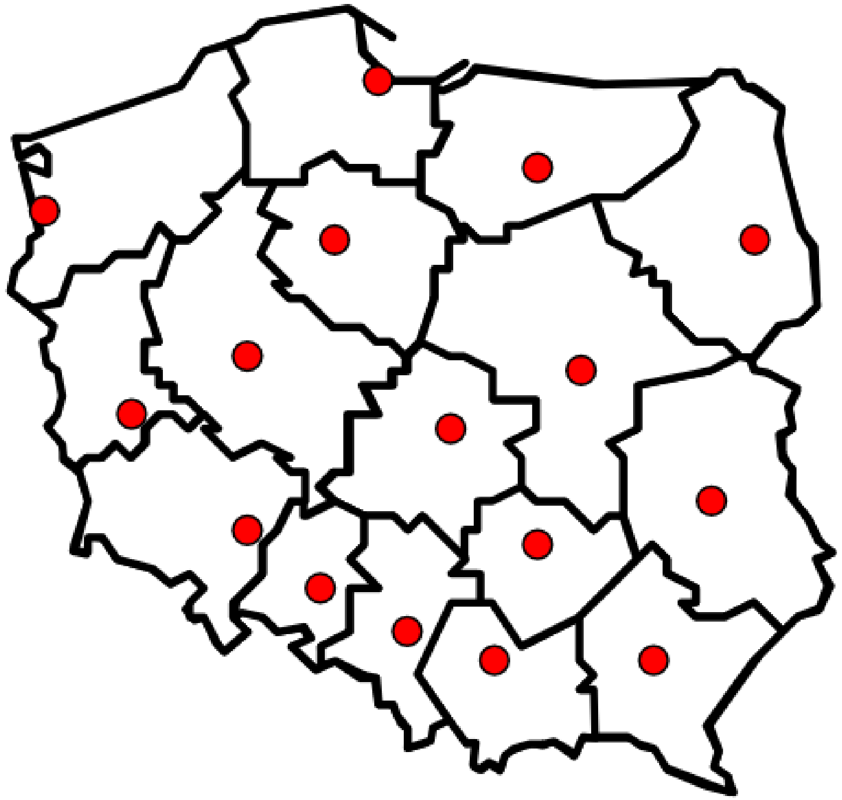
Lokalizacja punktu na płaszczyźnie
Metoda Trapezowa

Autorzy:

Maciej Wiśniewski

Oskar Blajsz



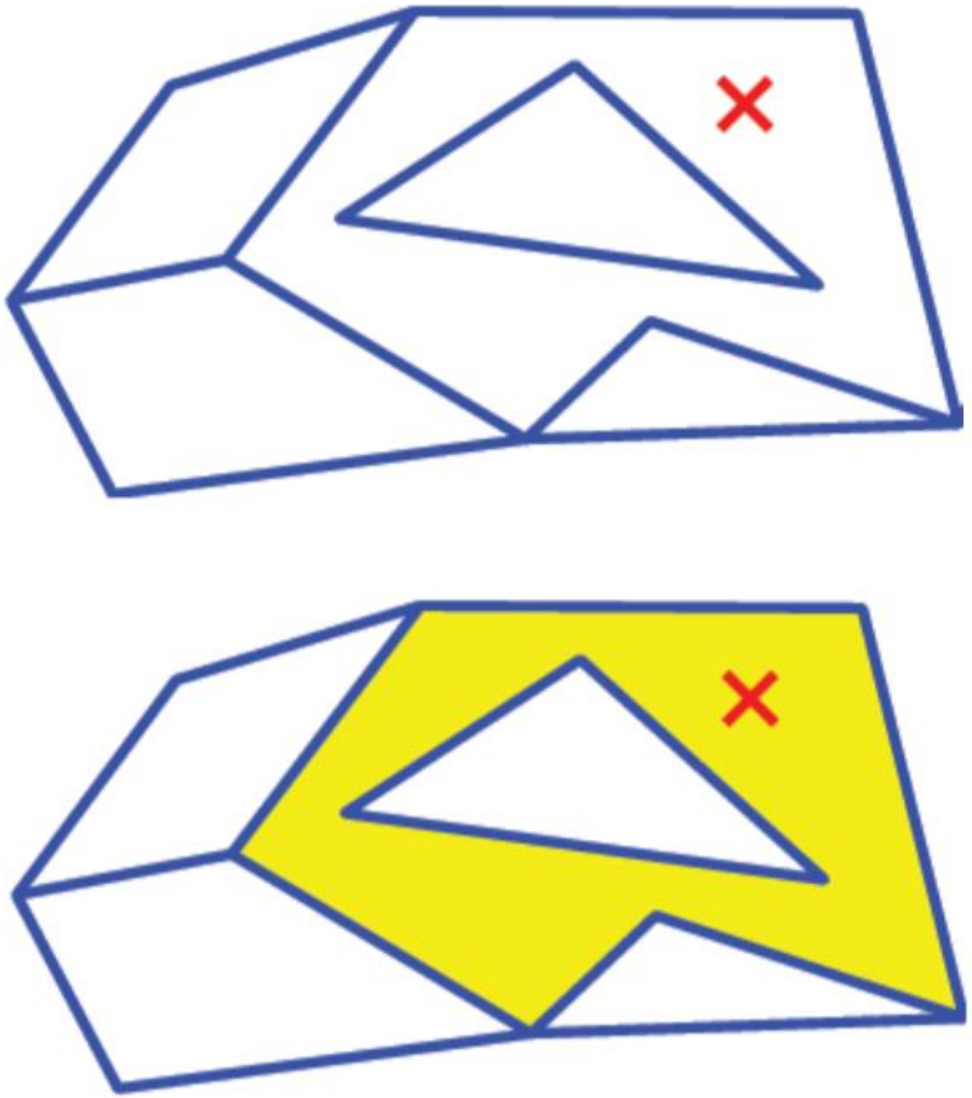
Przykład problemu

Dane:

- Mapa regionu podzielona na segmenty (kraje, województwa)
- Współrzędne punktu geograficznego na tej mapie

Szukane:

- Segment w którym znajduje się punkt



Ogólna definicja problemu

Dane:

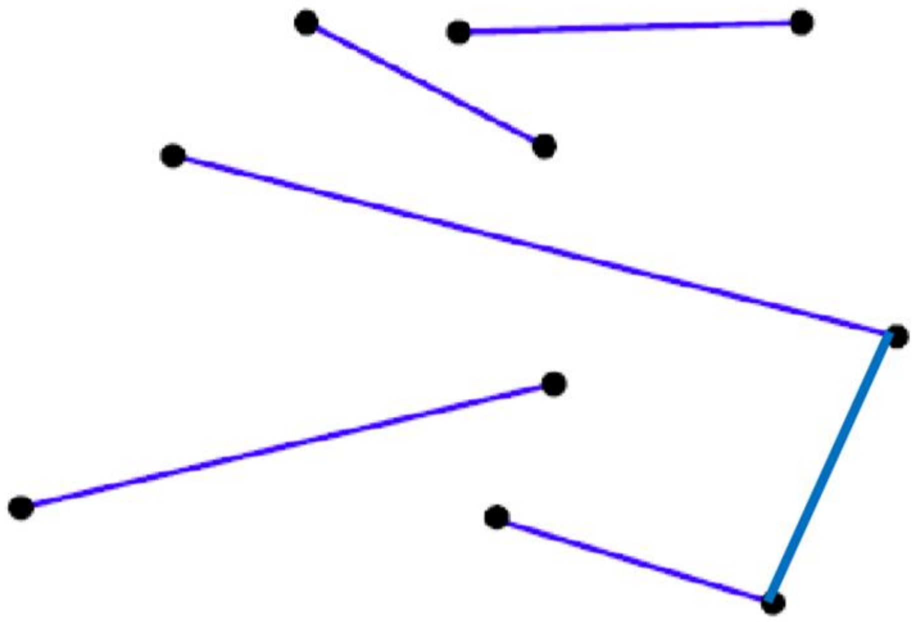
- Poligonowy podział płaszczyzny (podział planarny)
- Punkt na tej płaszczyźnie

Szukane:

- Wielokąt (ściana) zawierająca zadany punkt

Oczekiwane złożoności operacji:

- Pamięciowa O(n)
- Czasowa lokalizacji O(log n)
- Czasowa konstrukcji O(n log n)

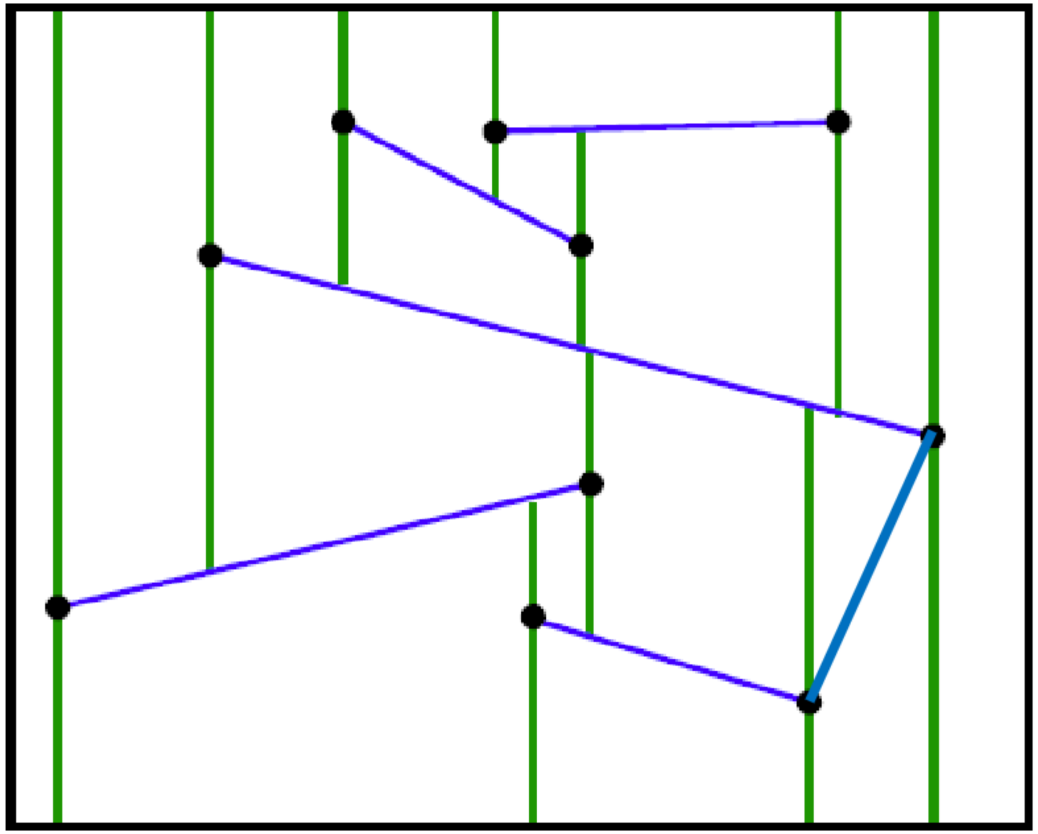


Takie położenie odcinków zakładamy w celu uproszczenia algorytmu oraz jego implementacji

Położenie ogólne odcinków

Założenia:

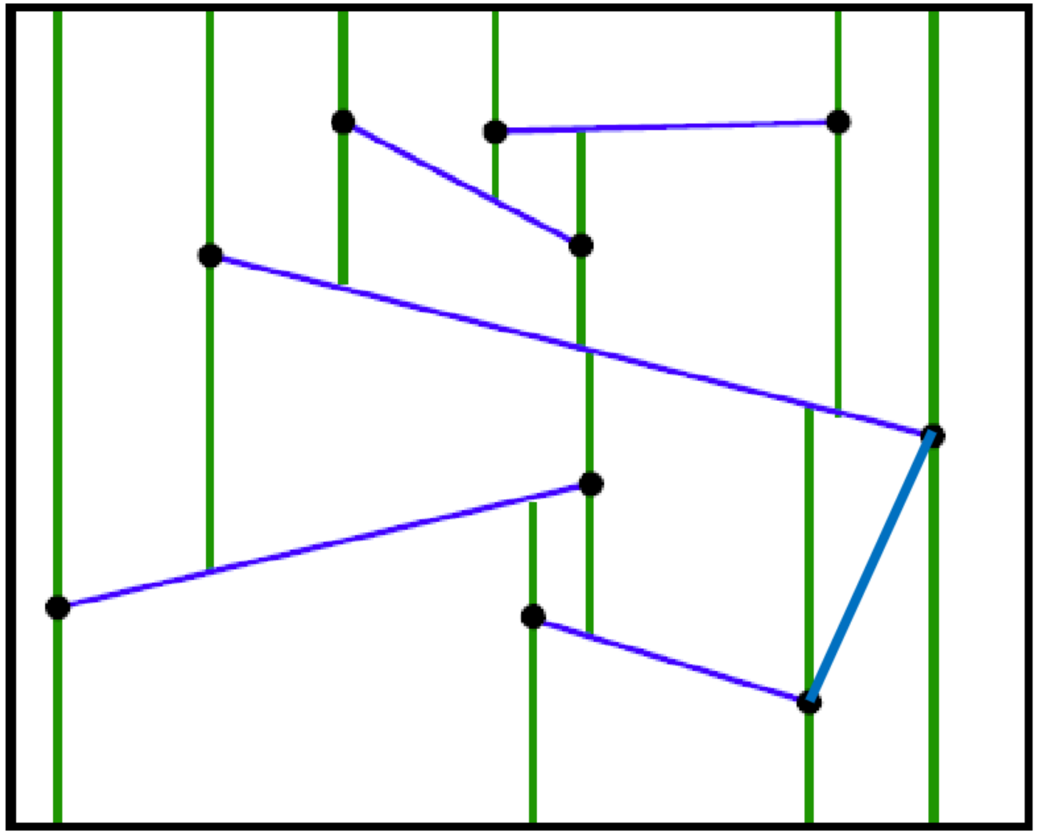
- Żaden odcinek nie jest pionowy
- Wierzchołki żadnych dwóch odcinków nie mają takiej samej współrzędnej X (poza końcami połączonych odcinków)
- Odcinki przecinają się tylko w wierzchołkach



Mapa trapezowa

Mapa trapezowa powstaje poprzez otoczenie zbioru odcinków prostokątem zawierającym je wszystkie, a następnie poprowadzenie pionowych linii od każdego końca odcinka.

Pionowe linie kończą się, gdy napotkają inny odcinek lub brzeg zewnętrznego prostokąta.



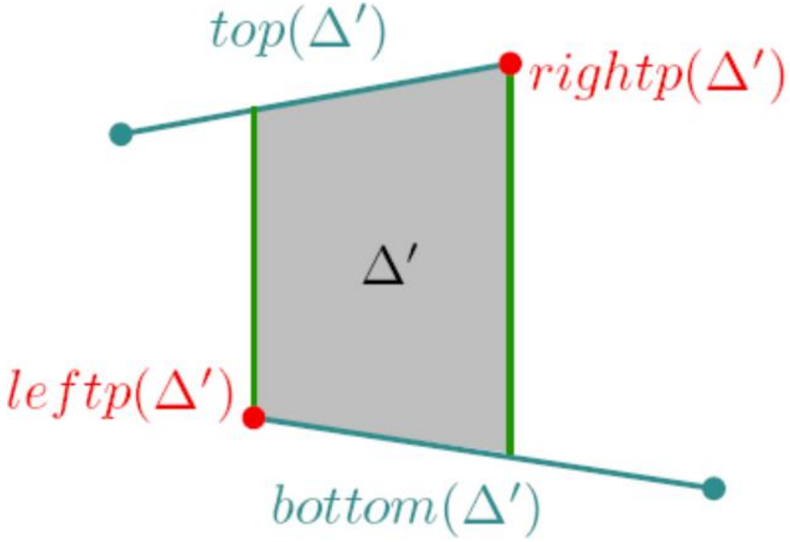
Elementy mapy trapezowej

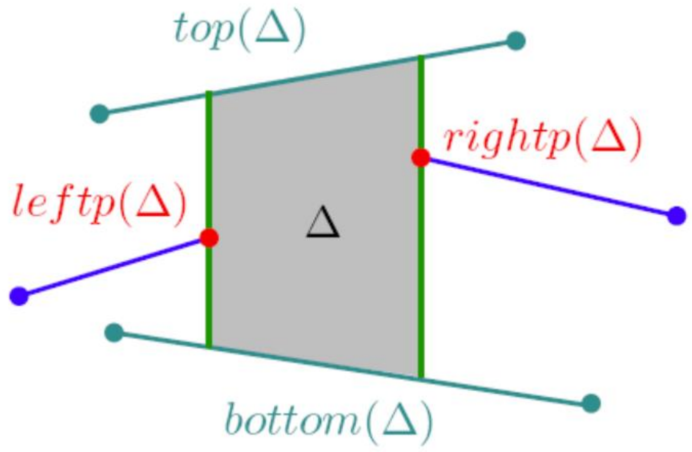
Elementy mapy trapezowej mogą być trapezami lub trójkątami. Jest to zależne od liczby ich pionowych boków, która wynosi jeden lub dwa.

Każdy element mapy posiada zawsze dokładnie dwa „niepionowe” boki.

Zakładając zbiór n odcinków w położeniu ogólnym, możemy ograniczyć liczbę:

- Wierzchołków do maksymalnie 6n+4
- Trapezów do maksymalnie 3n+1



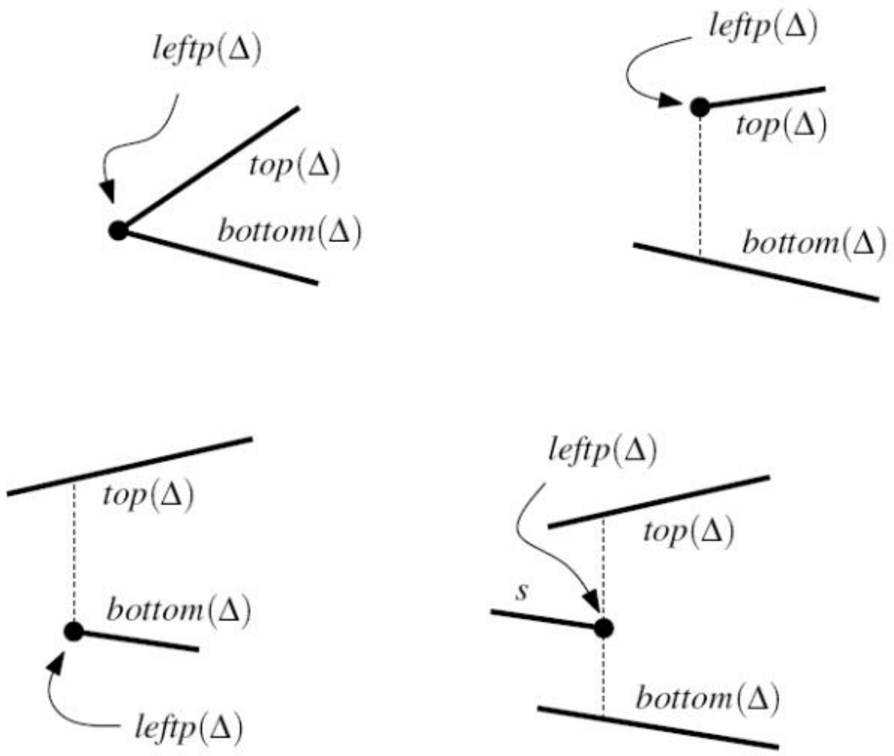


Struktura trapezu

Boczne boki każdego elementu mapy trapezowej zawsze będą pionowymi odcinkami.

Struktura, którą wykorzystujemy do przechowywania trapezu przechowuje dwa odcinki: górny (top(**Δ**)) i dolny (bottom(**Δ**)) oraz dwa wierzchołki trapezu w postaci punktów: lewy (leftp(**Δ**)) i prawy (rightp(**Δ**)).

Przechowywane wierzchołki trapezu mogą znajdować się w 4 różnych pozycjach względem górnego i dolnego odcinka.



Możliwe pozycje wierzchołków

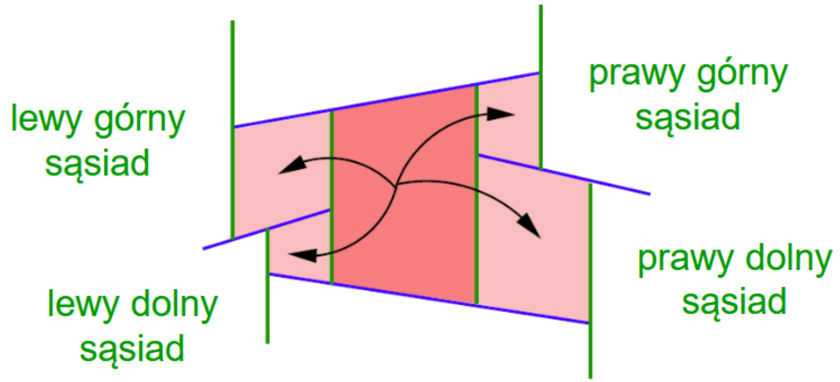
Na przykładzie lewego wierzchołka. Dla prawego sytuacja wygląda analogicznie.

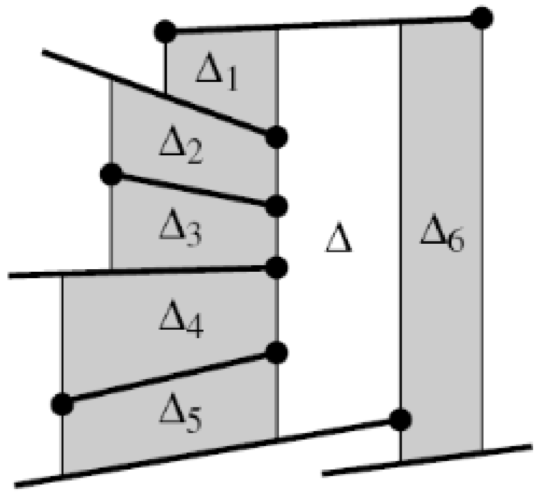
Jeśli wierzchołek leży na lewym końcu jednego z odcinków to obliczamy wartość drugiego odcinka w punkcie o takiej samej współrzędnej X, a następnie prowadzimy pionową prostą.

Jeśli wierzchołek leży pomiędzy odcinkami to obliczamy ich wartości w punktach o współrzędnej X wierzchołka oraz prowadzimy między nimi prostą pionową.

Jeśli wierzchołek leży na lewym końcu obydwu odcinków to nie prowadzimy żadnej prostej.

Sąsiedztwo trapezów – reprezentacja mapy





Jeśli odcinki nie są w położeniu ogólnym to trapez może posiadać więcej niż 4 sąsiadów

Trapezy sąsiadują ze sobą, tylko jeśli mają wspólną krawędź pionową. Warto zauważyć, że jeśli rozpatrujemy zbiór odcinków w położeniu ogólnym to każdy trapez będzie miał co najwyżej 4 sąsiadów.

Aby reprezentować mapę trapezową wykorzystamy ten fakt i będziemy ją reprezentować jako strukturę powiązań między sąsiadami.

Każdy trapez posiada wskaźniki do czterech swoich sąsiadów: lewego górnego oraz dolnego i prawego górnego oraz dolnego.

Randomizowany algorytm przyrostowy konstrukcji T(S)

Dane wejściowe:

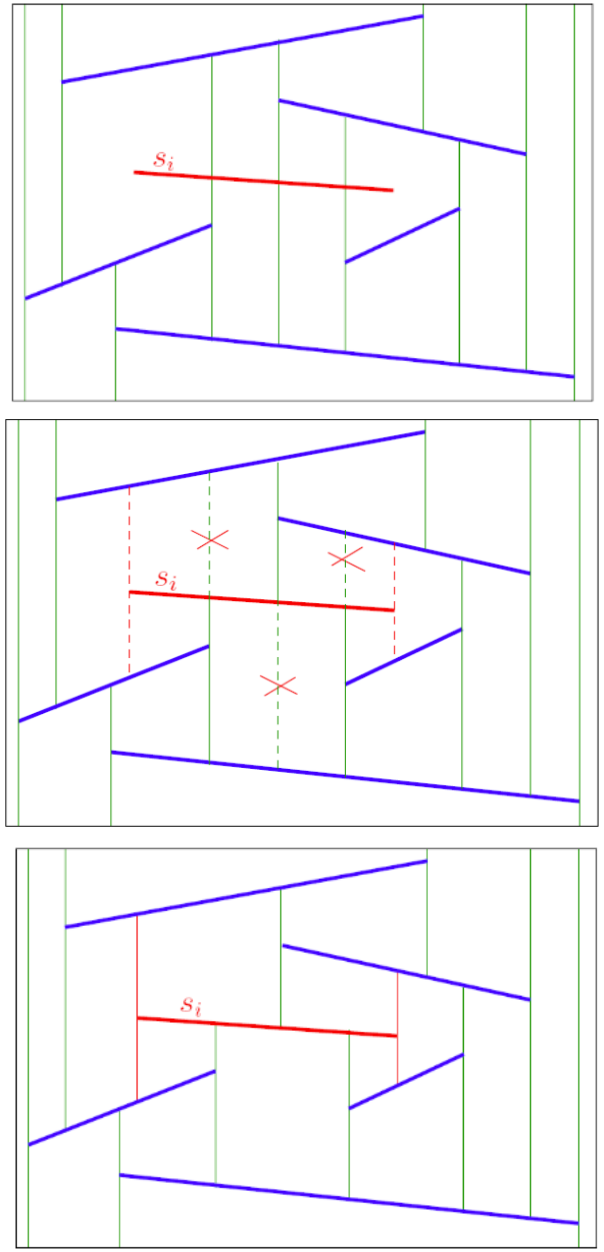
- Zbiór odcinków $S = \{s_1, s_2, s_3, \dots, s_n\}$ położonych na płaszczyźnie dwuwymiarowej w położeniu ogólnym

Wynik:

- Mapa trapezowa T(S) oraz struktura przeszukiwań D dla T(S) w postaci grafu przeszukiwań

Przebieg algorytmu

1. Wyznaczamy prostokąt zawierający w sobie wszystkie odcinki z S.
2. Inicjalizujemy mapę trapezową T(S) oraz strukturę przeszukiwań D
3. Dla każdego s_i:
    1. Znajdujemy zbiór trapezów $\Delta_0, \Delta_1, \ldots, \Delta_k$ przeciętych przez s_i
    2. Usuwamy znalezione trapezy $\Delta_0, \Delta_1, \ldots, \Delta_k$ z mapy trapezowej T(S)
    3. Dodajemy do T(S) nowo utworzone trapezy, które stworzyły się po dodaniu odcinka s_i
    4. Usuwamy liście zawierające trapezy $\Delta_0, \Delta_1, \ldots, \Delta_k$ ze struktury przeszukiwań D
    5. W strukturze przeszukiwań D tworzymy liście zawierające nowo utworzone trapezy i łączymy je z istniejącymi węzłami, w razie potrzeby dodając nowe węzły



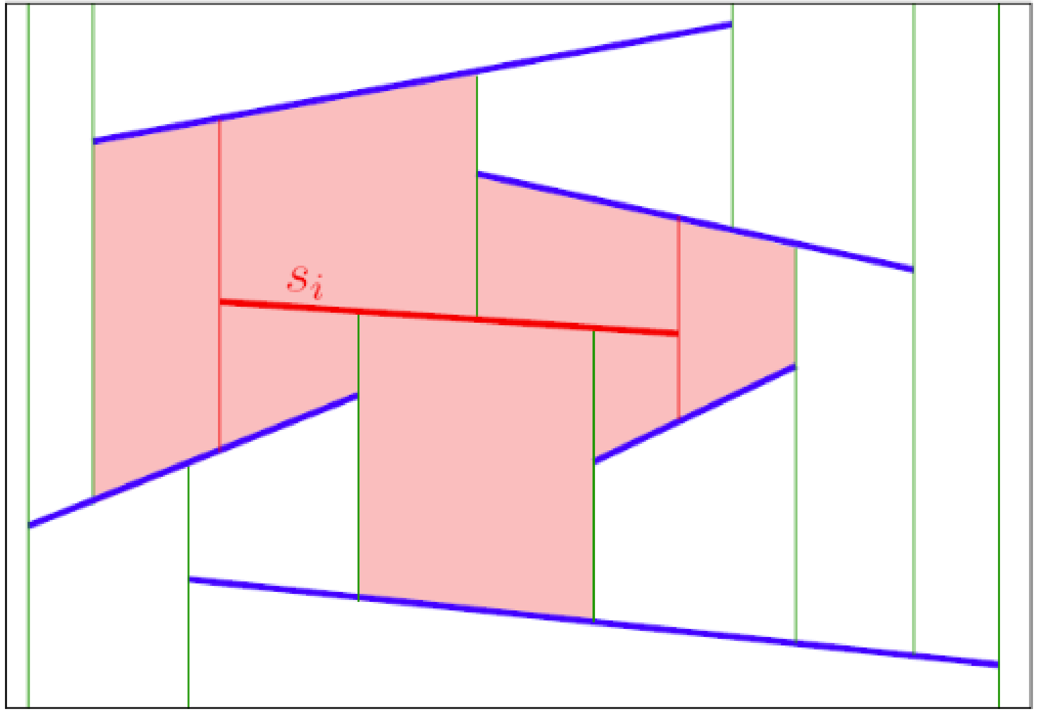
Wstawianie odcinka s_i

Odcinek s_i wstawiamy do struktury $T(S_{i-1})$.

Nowo wstawiany odcinek może przecinać kilka trapezów.

Każdy z trapezów, które są przecinane przez obecnie wstawiany odcinek s_i może zostać podzielony na maksymalnie 4 nowe trapezy.

Podczas wstawiania nowego odcinka, niektóre trapezy mogą zostać połączone wskutek skracania pionowych linii.



Strefa s_i

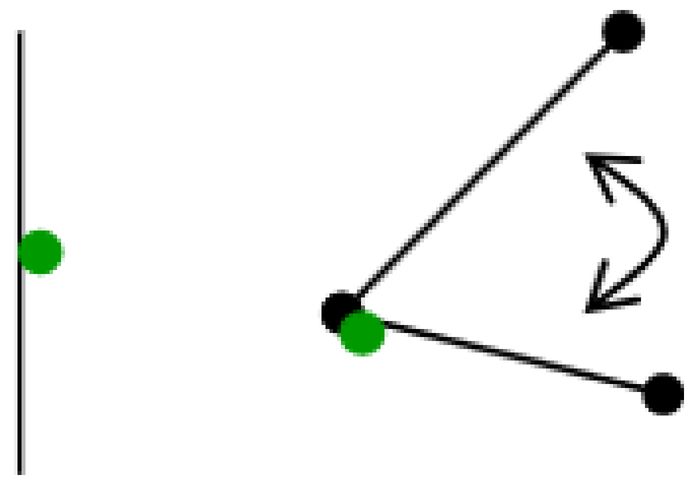
Strefę dla odcinka si w mapach T(Si) i T(Si-1) tworzą wszystkie trapezy przecinane przed odcinek si

Dla T(Si) strefą jest suma wszystkich nowo utworzonych trapezów

Dla T(Si-1) strefą jest suma wszystkich trapezów, koniecznych do usunięcia

Strefy dla obu map T(Si) i T(Si-1) są jednakowe pod względem powierzchni i kształtu. Różnią się jedynie wewnętrznym podziałem na trapezy

Wyznaczenie strefy dla s_i



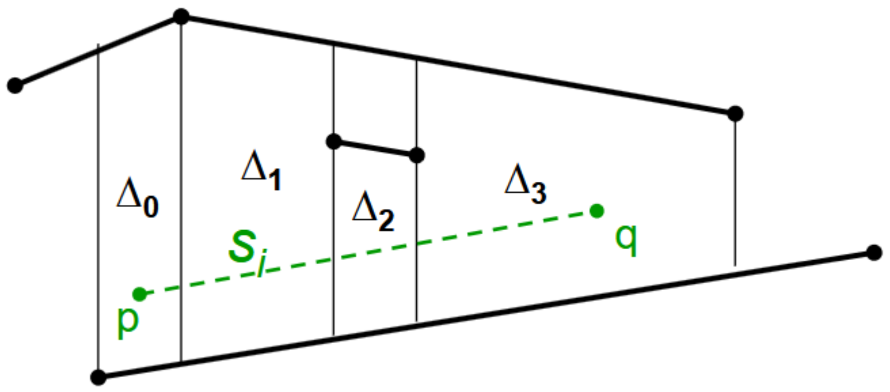
Zaczynamy od przeszukania struktury D aż do znalezienia trapezu Δ_0, który zawiera lewy koniec p odcinka s_i.

Sposoby rozwiązywania wątpliwych sytuacji mogących wystąpić podczas przeszukiwania:

- Jeśli okaże się, że punkt p leży na prostej pionowej to przyjmujemy, że leży po jej prawej stronie
- Jeśli okaże się, że punkt p jest wspólnym punktem z innym odcinkiem z S, wtedy porównujemy nachylenie obu odcinków. Jeśli nachylenie s_i będzie mniejsze to przyjmujemy, że leży on poniżej drugiego odcinka.

Mając to na uwadze pobieramy ze struktury D trapez Δ_0 i rozpoczynamy wyznaczanie strefy dla odcinka s_i

Wyznaczenie strefy dla s_i



j = 0

while q znajduje się na prawo od rightp(Δ_j) do:

jeśli rightp(Δ_j) znajduje się powyżej s_i, to:

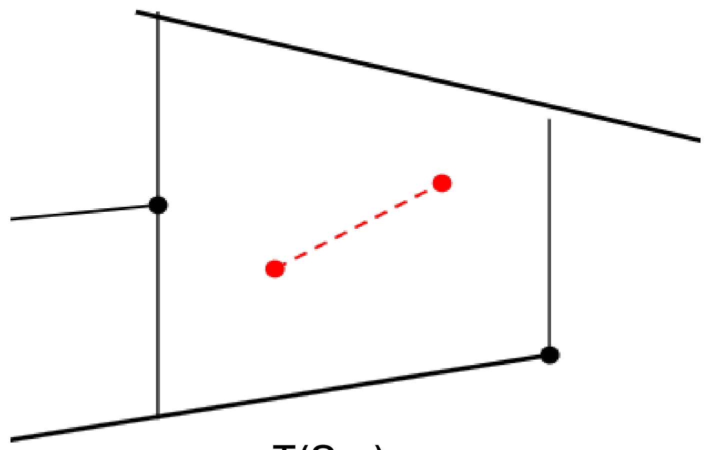
$\Delta_{j+1} \leftarrow$ dolny prawy sąsiad Δ_j

w przeciwnym razie:

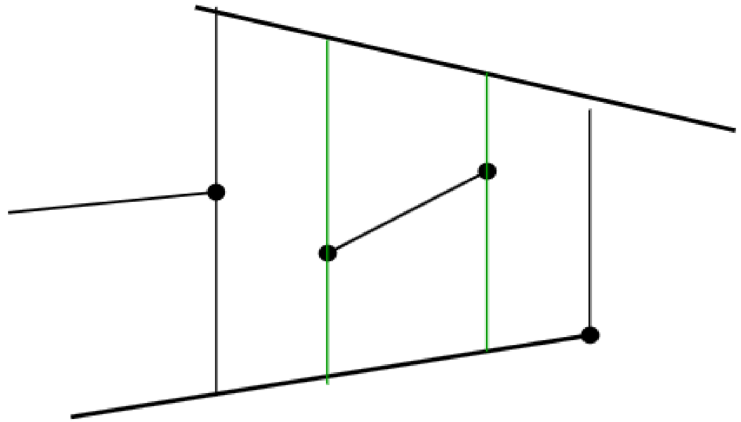
$\Delta_{j+1} \leftarrow$ górny prawy sąsiad Δ_j

$j \leftarrow j + 1$

zwróć $\Delta_0, \Delta_1, \ldots, \Delta_k$



$T(S_{i-1})$

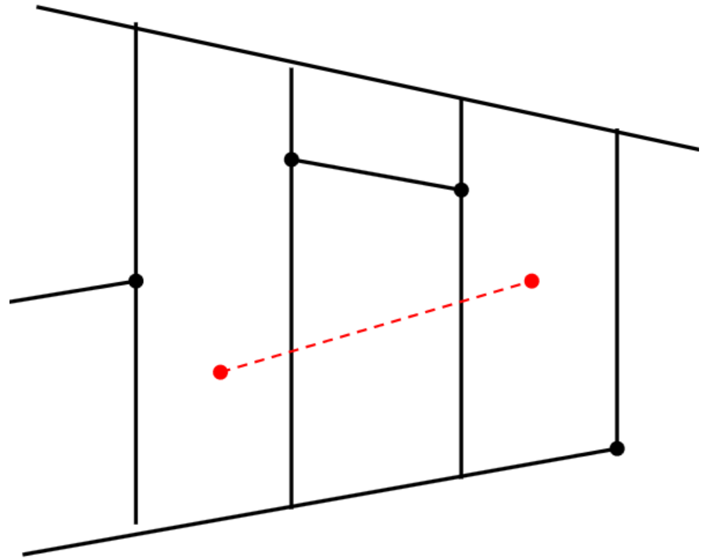


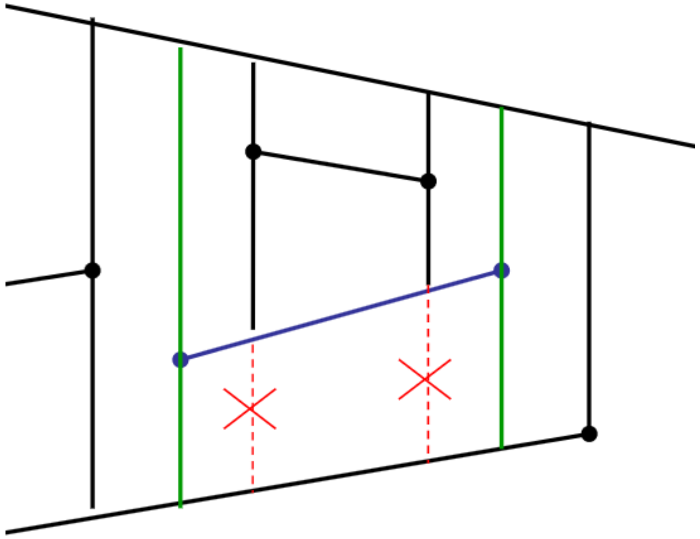
$T(S_i)$

Aktualizacja mapy trapezowej

Jeśli nowo dodawany odcinek s_i w całości zawiera się w jednym trapezie to możemy zaktualizować mapę w następujących krokach:

1. Usuwamy trapez Δ zawierający odcinek s_i z mapy
2. Zastępujemy go przez odpowiednią ilość nowych trapezów (maksymalnie 4)
3. Aktualizujemy informacje dla trapezów o sąsiadach, bottom(Δ), top(Δ), leftp(Δ), rightp(Δ)





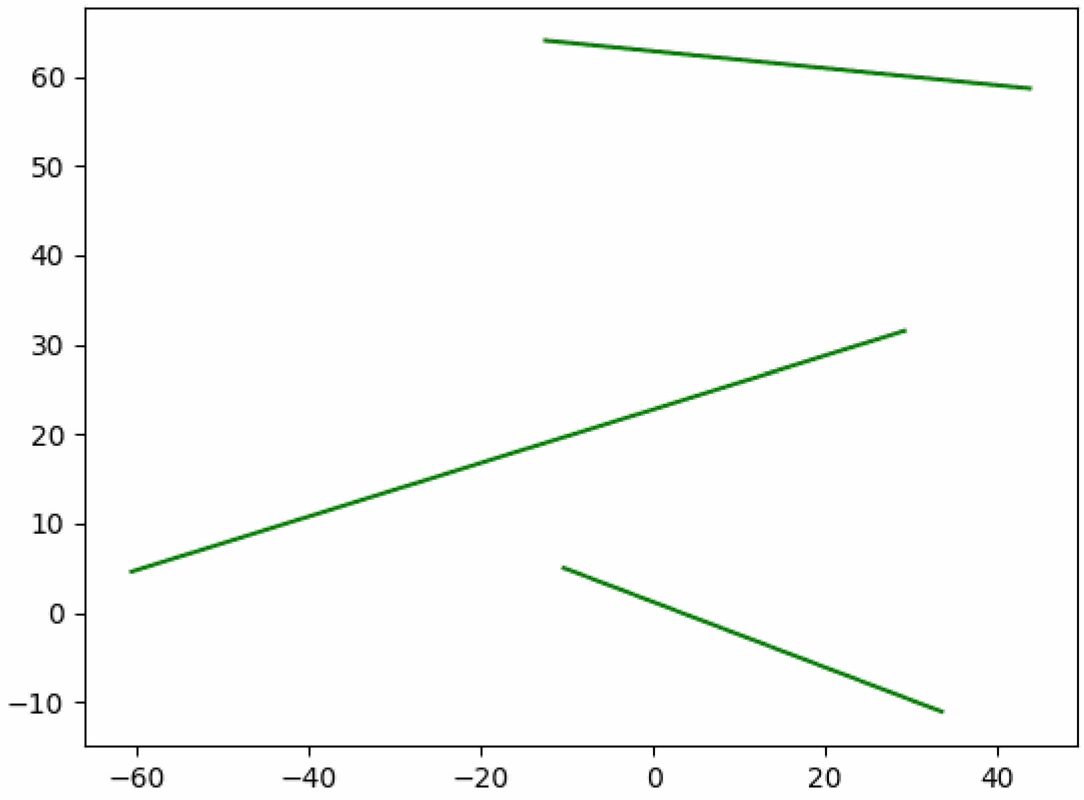
Aktualizacja mapy trapezowej

W przypadku, gdy nowo dodawany odcinek s_i zawiera się w więcej niż jednym trapezie to w celu aktualizacji mapy trapezowej musimy wykonać następujące kroki:

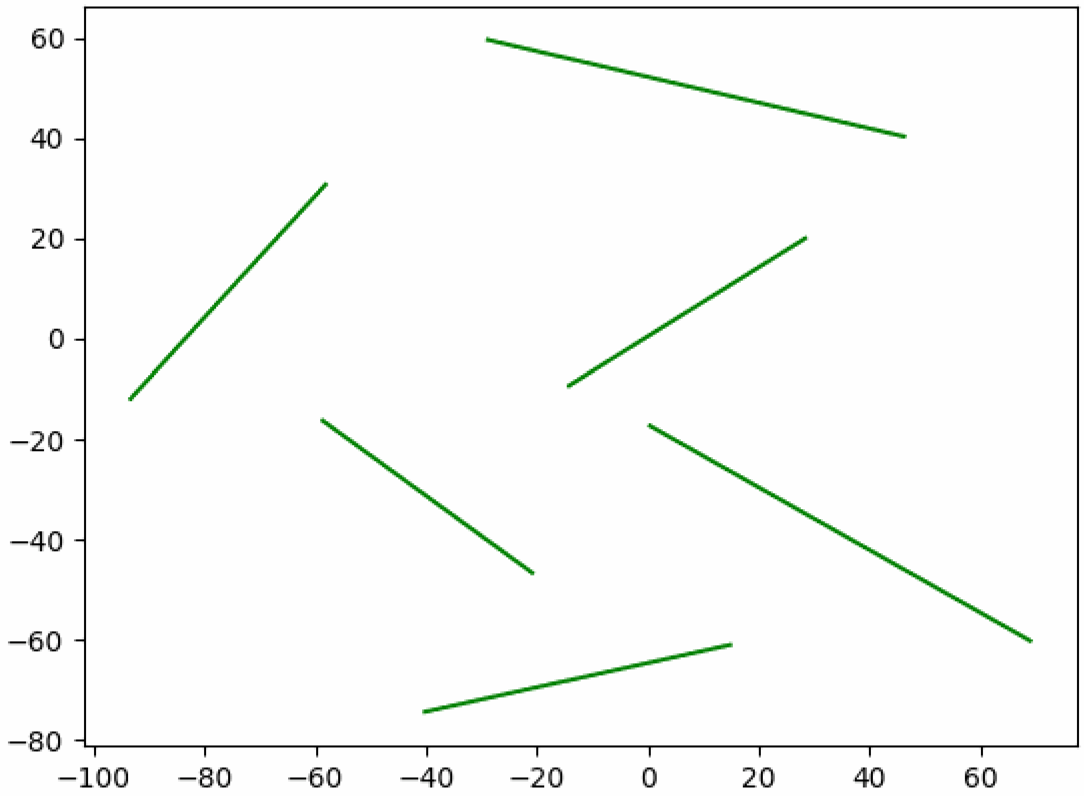
1. Wstawiamy rozszerzenia pionowe przechodzące przez oba końce odcinka s_i dzieląc trapezy Δ_0 i Δ_k.
2. Skracamy pionowe linie, które przecinają odcinek s_i, tak aby się z nim stykały.
3. Aktualizujemy informacje o sąsiadach dla zmienionych i utworzonych trapezów.

Oczekiwany czas konstrukcji mapy trapezowej - O(n log n)

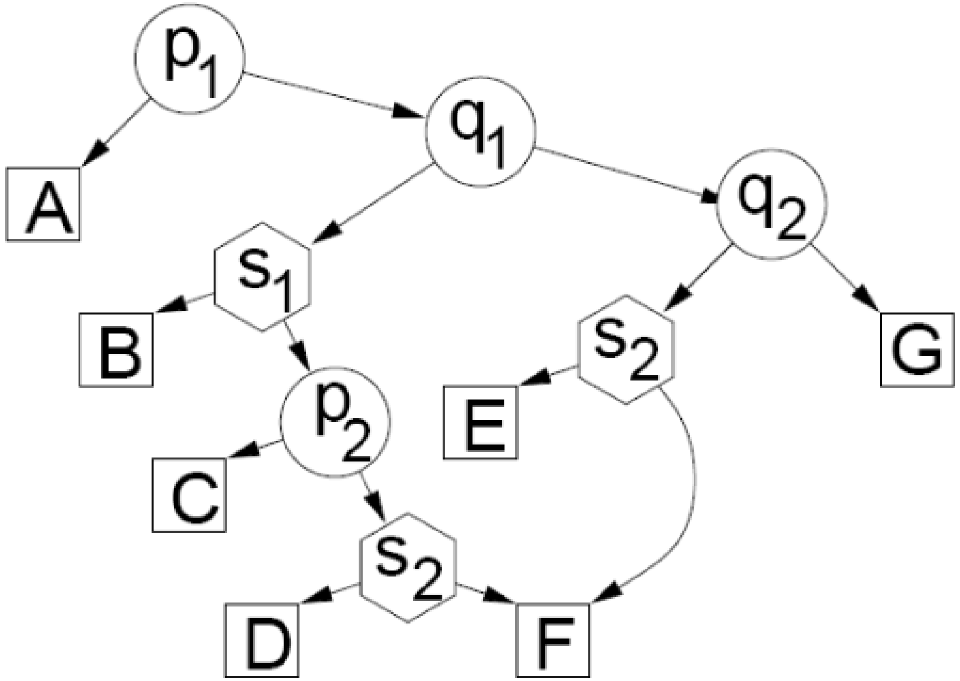
Przykład 1 tworzenia mapy trapezowej

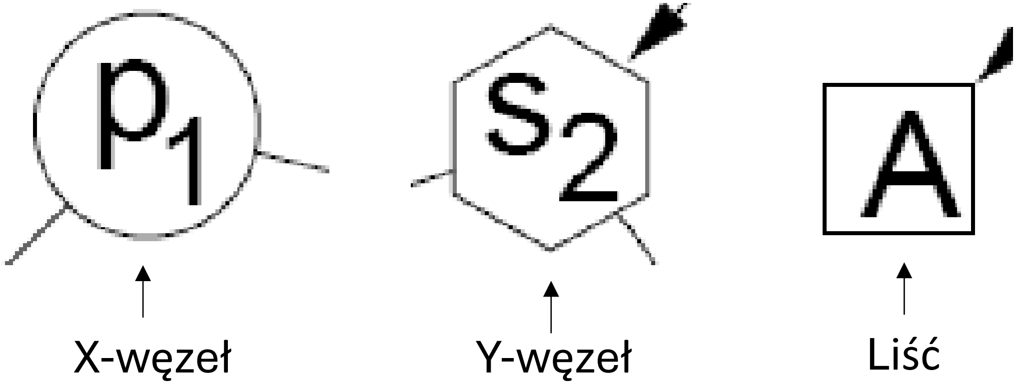


Przykład 2 tworzenia mapy trapezowej



Graf wyszukiwania





Jest to struktura, w której każdy trapez z T(S) ma wskaźnik do odpowiadającego mu liści, a liść ma wskaźnik do odpowiadającemu mu trapezu w T(S)

Posiada on węzły wewnętrzne dwóch kategorii:

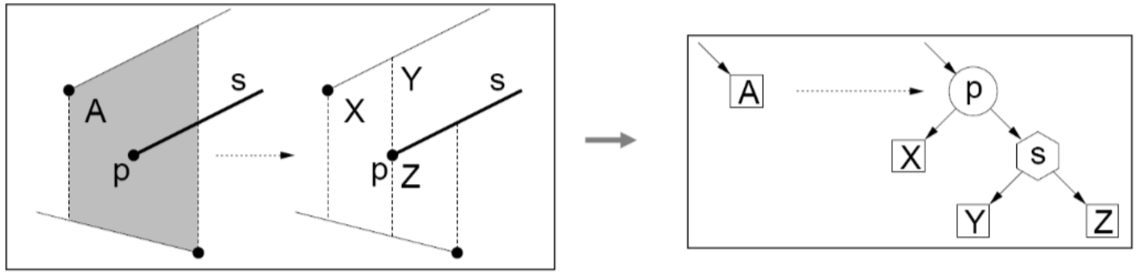
- x-węzły – przechowują współrzędne wierzchołka
- y-węzły – przechowują wskaźnik do odcinka

Schemat poruszania się po grafie wyszukiwania:

- Jeśli znajdujemy się w x-węźle sprawdzamy, po której stronie pionowej prostej przechodzącej przez punkt znajdujący się w tym węźle leży punkt p. Jeśli leży po lewej to kierujemy się do lewego dziecka obecnego węzła.
- Jeśli znajdujemy się w y-węźle sprawdzamy, czy punkt p leży nad czy pod odcinkiem znajdującym się w węźle. Jeśli nad to kierujemy się do lewego dziecka obecnego węzła

Przyrostowa konstrukcja grafu

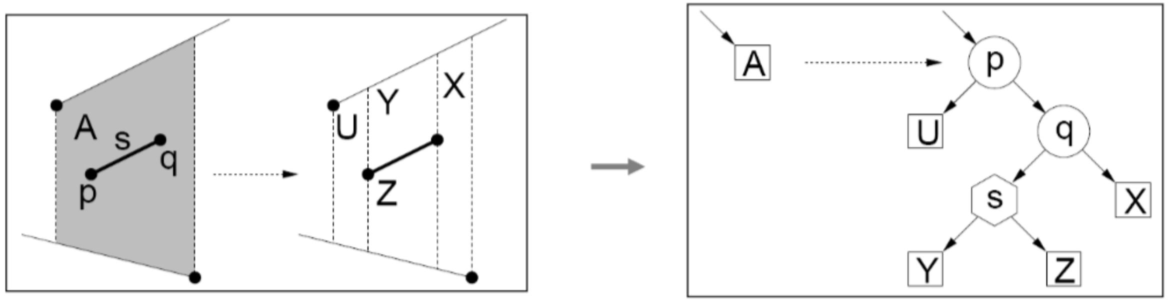
Graf wyszukiwania ma przyrostową konstrukcję, oznacza to, że usuwany trapez jest zastępowany fragmentem struktury wyszukiwania kierującym do jednego z nowo stworzonych trapezów.



Przypadek 1

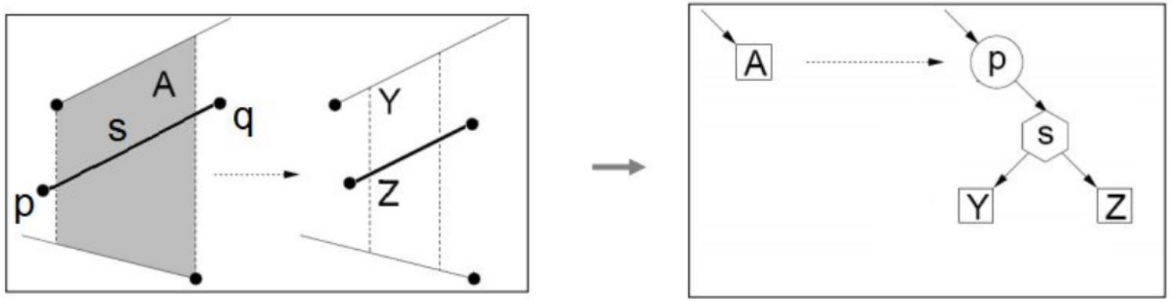
Pojedynczy trapez A (zawierający jeden wierzchołek odcinka) jest zastępowany przez trzy trapezy X, Y i Z

Przyrostowa konstrukcja grafu



Przypadek 2

Pojedynczy trapez A (przecięty całkowicie) jest zastępowany przez dwa trapezy Y i Z



Przypadek 3

Pojedynczy trapez zawiera cały odcinek i zostaje podzielony na cztery trapezy U, X, Y i Z

Algorytm wyszukiwania punktu

Dane: Punkt q zadany w postaci współrzędnych x i y

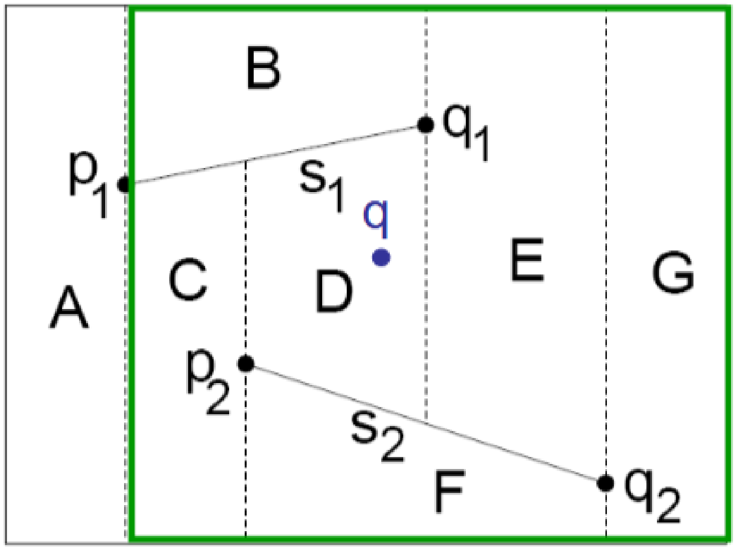
Wynik: Trapez Δ zawierający punkt q

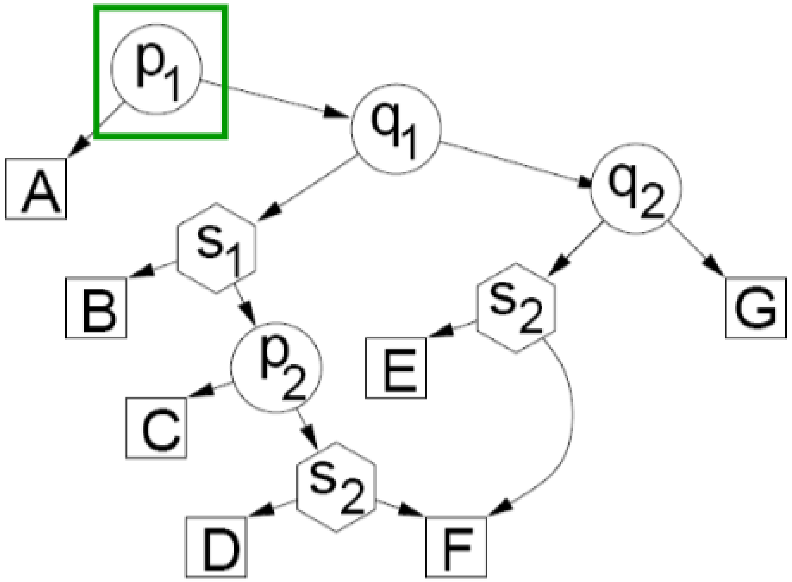
Algorytm:

- Jeśli bieżący element jest x-węzłem:
  - Jeśli szukany punkt znajduje się po **lewej stronie pionowej linii** przechodzącej przez punkt w x-węźle to przejdź do **lewego potomka**.
  - W przeciwnym przypadku przejdź do **prawego potomka**.
- Jeśli bieżący element jest y-węzłem:
  - Jeśli szukany punkt znajduje się **poniżej poziomej linii** w y-węźle to przejdź do **lewego potomka**.
  - W przeciwnym przypadku przejdź do **prawego potomka**.
- Jeśli bieżący element jest liściem to zakończ algorytm (znaleziono trapez, w którym znajduje się punkt).

Przykład wyszukiwania punktu

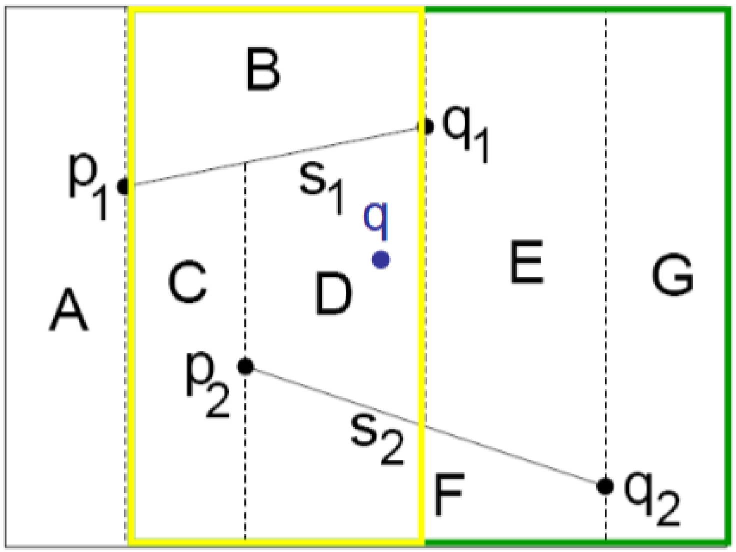
Znajdujemy się w x-węźle z punktem p1. Sprawdzamy po której stronie tego punktu leży dany punkt q. Punkt q leży po prawej stronie, zatem udajemy się do prawego potomka

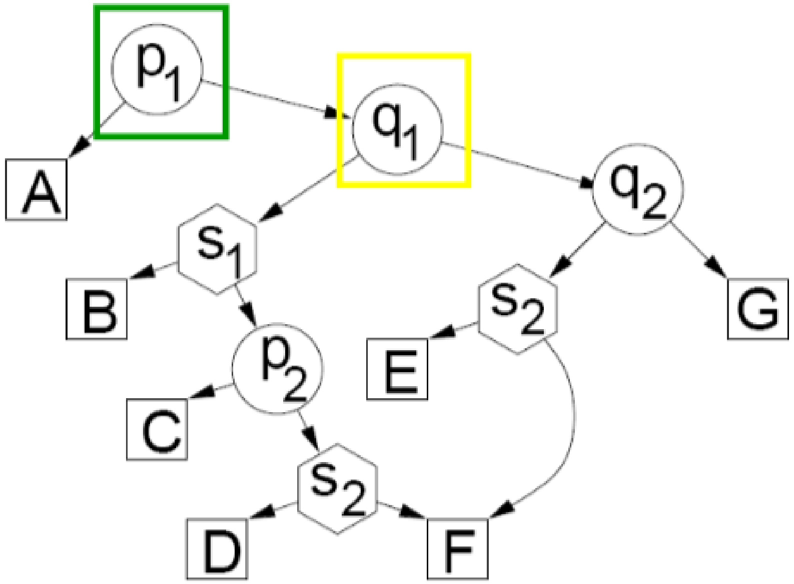




Przykład wyszukiwania punktu

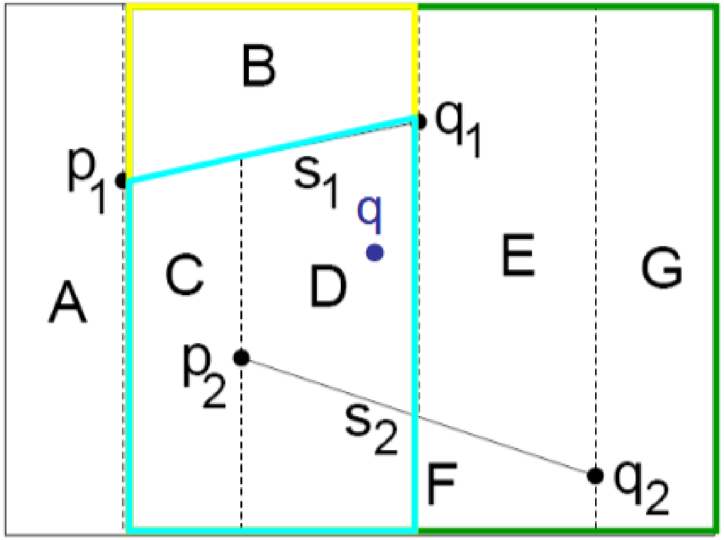
Znajdujemy się w x-węźle z punktem q1. Sprawdzamy po której stronie tego punktu leży dany punkt q. Punkt q leży po lewej stronie, zatem udajemy się do lewego potomka

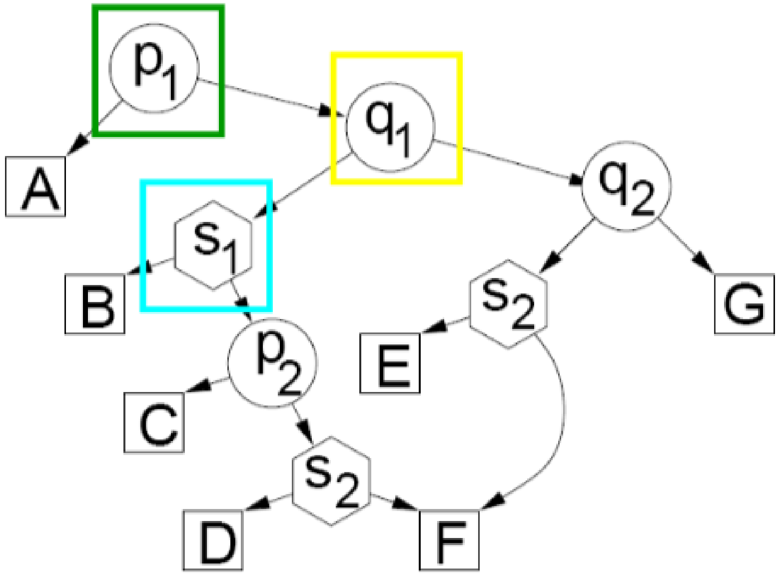




Przykład wyszukiwania punktu

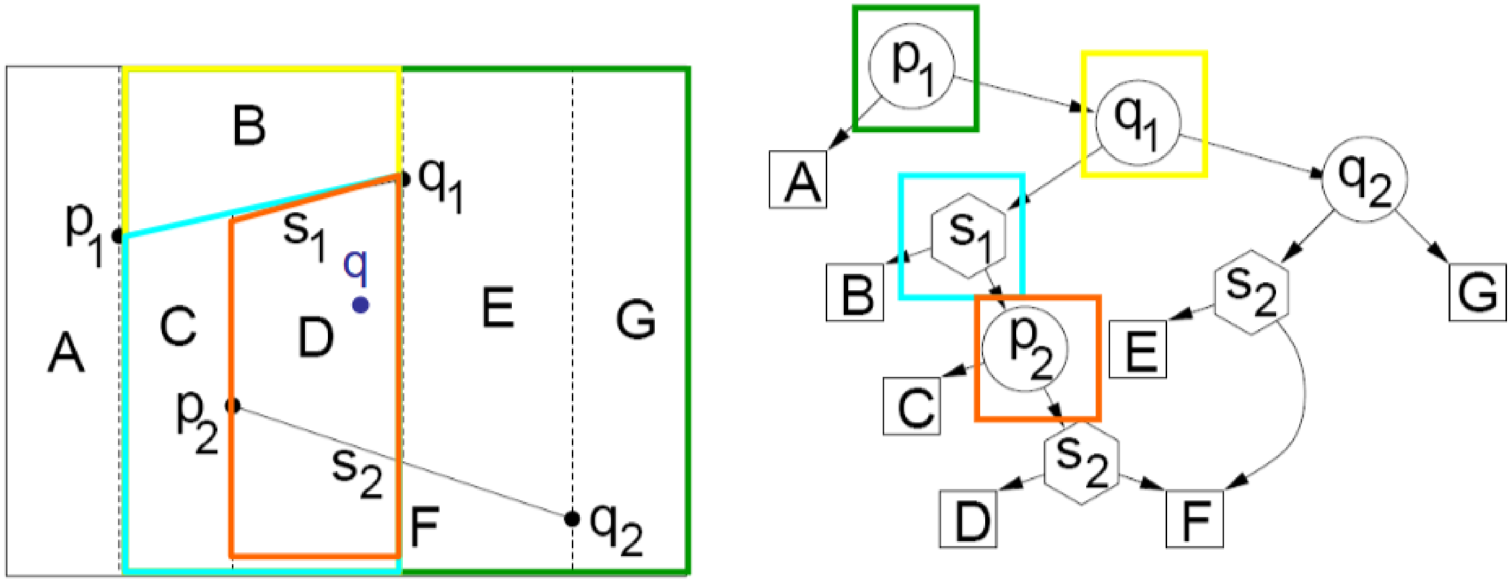
Znajdujemy się w y-węźle z odcinkiem s1. Sprawdzamy, czy punkt q leży powyżej czy poniżej niego. Punkt q leży poniżej niego, więc udajemy się do prawego potomka





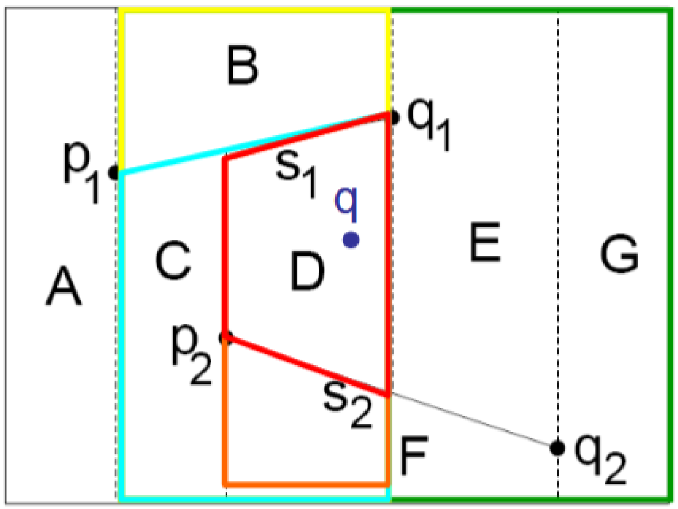
Przykład wyszukiwania punktu

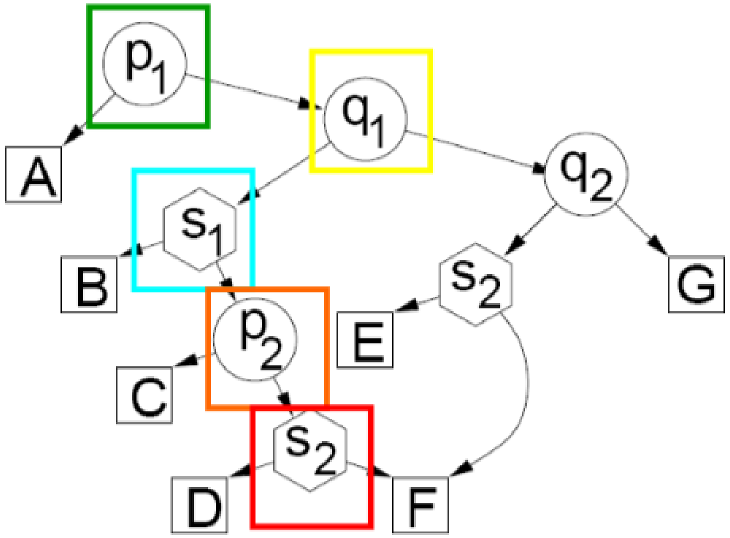
Znajdujemy się w x-węźle z punktem p2. Sprawdzamy po której stronie tego punktu leży dany punkt q. Punkt q leży po prawej stronie, zatem udajemy się do prawego potomka



Przykład wyszukiwania punktu

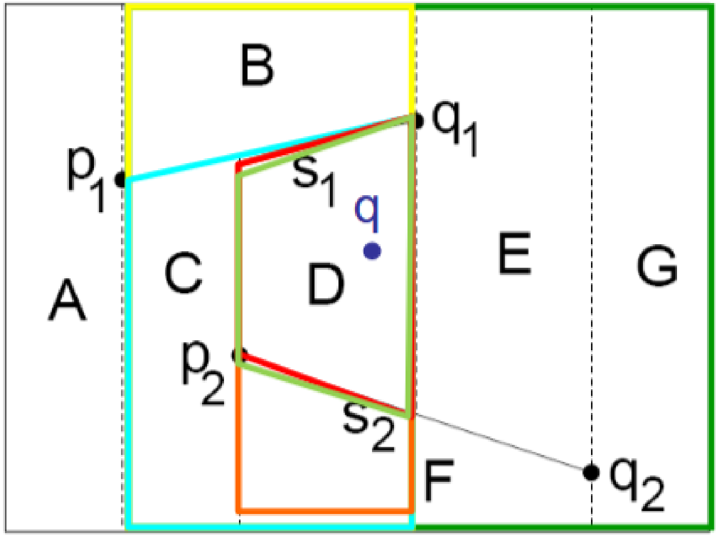
Znajdujemy się w y-węźle z odcinkiem s1. Sprawdzamy, czy punkt q leży powyżej czy poniżej niego. Punkt q leży powyżej niego, więc udajemy się do lewego potomka

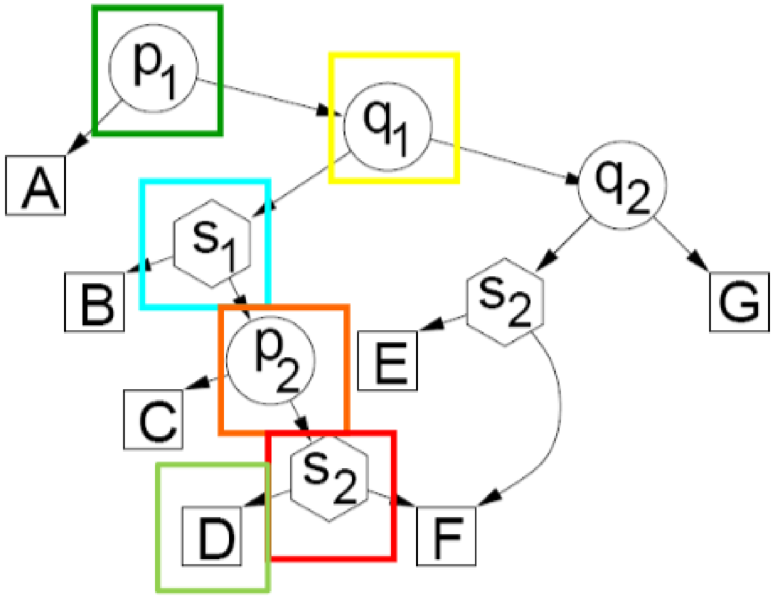




Przykład wyszukiwania punktu

Dotarliśmy do liścia, zatem kończymy algorytm i zwracamy wskaźnik na trapez D, w którym znajduje się dany punkt q.

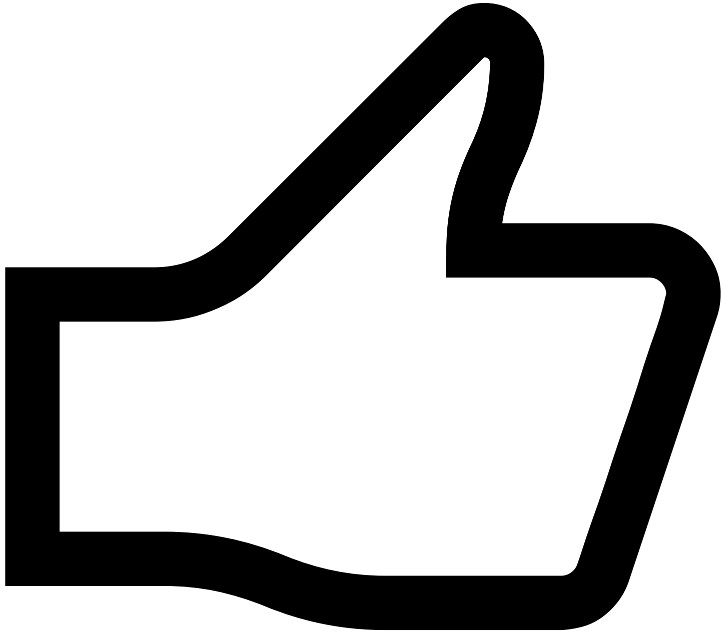




Bibliografia

- https://upel.agh.edu.pl/pluginfile.php/433098/mod_resource/content/1/wyklad_lokpkt_m.pdf
- https://users.dimi.uniud.it/~claudio.mirolo/teaching/geom_comput/presentations/trapezoidal_map.pdf
- http://cglab.ca/~cdillaba/comp5008/trapezoid.html
- https://algo.uni-trier.de/lectures/algeo/slides/ag-ws20-vl06-point-location.pdf

KONIEC



Dziękujemy za uwagę!