Lokalizacja punktu na płaszczyźnie 2D metodą trapezową

Prezentacja przygotowana przez:
Krzysztof Kwiecień
Paweł Żurawski

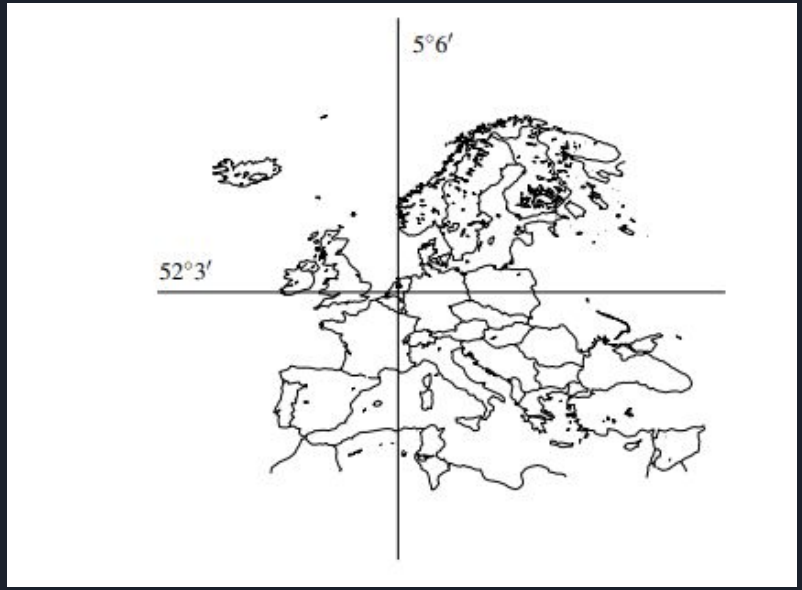
Ogólna definicja problemu

Dane:

- Mapa świata
- Koordynaty współrzędnych geograficznych punktu (5°6’ E,52°3’ N)

Szukane:

- Region w którym znajduje się zaznaczony punkt

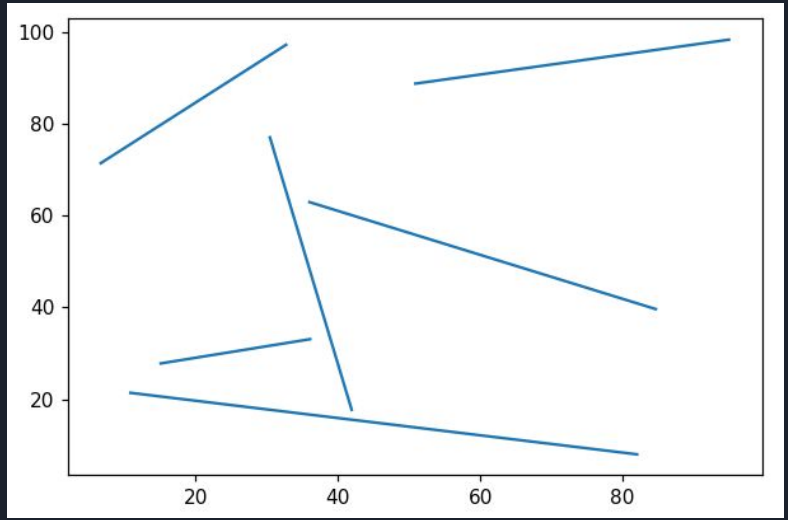


Położenie ogólne odcinków

O położeniu ogólnym odcinków $S = \{s_1, s_2, ..., s_n\}$ na płaszczyźnie dwuwymiarowej mówimy wtedy, gdy:

- żaden odcinek nie jest pionowy,
- odcinki przecinają się tylko w wierzchołkach,
- wierzchołki nie mają takich samych współrzędnych x (nie dotyczy to końców połączonych odcinków)

Położenie ogólne odcinków pozwoli na łatwiejsze przedstawienie algorytmu wyznaczania mapy trapezowej oraz pozwoli usprawnić jego implementacje .



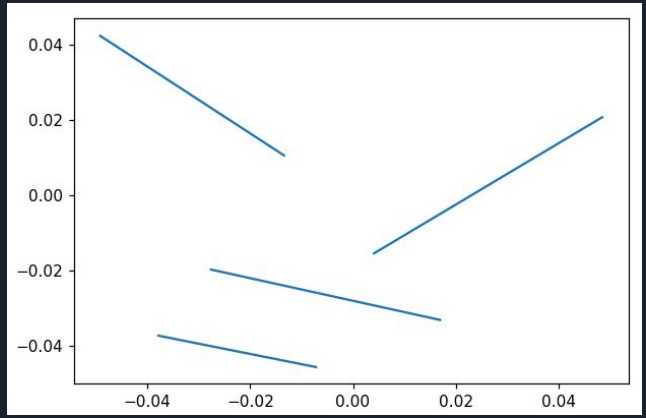
Mapa trapezowa

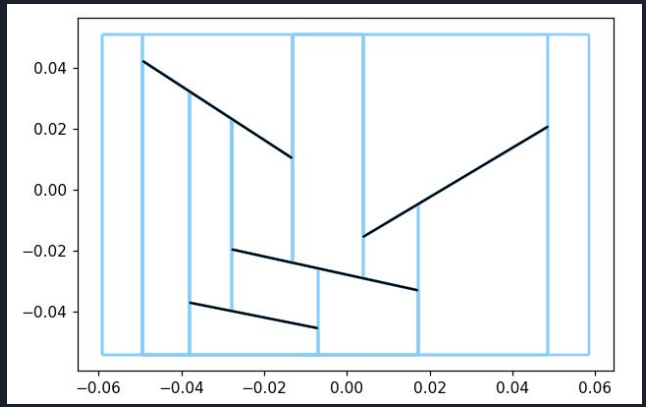
Mapa trapezowa T(S) jest podziałem zbioru odcinków S na trapezy lub trójkąty otrzymanym poprzez poprowadzenie pionowych rozszerzeń każdego z końców odcinka w S.

Pionowe rozszerzenia kończą się, gdy napotkają inny odcinek w S lub brzeg prostokąta.

Każdy element mapy ma dokładnie dwa boki poziome oraz jeden lub dwa boki pionowe.

Gdy n odcinków znajduje się w położeniu ogólnym, można wtedy ograniczyć liczbę wierzchołków do co najwyżej 6n+4 oraz liczbę trapezów do 3n+1,



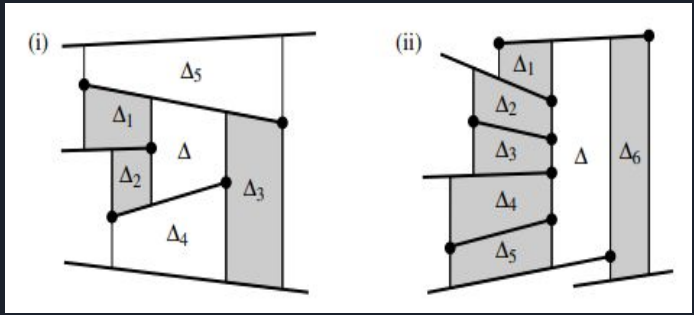


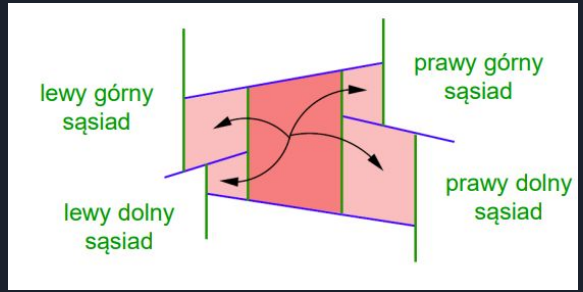
Reprezentacja mapy trapezowej

Do reprezentacji mapy trapezowej będziemy używać struktury powiązań sąsiedzkich

Dwa trapezy sąsiadują ze sobą wtedy i tylko wtedy gdy posiadają wspólną krawędź pionową.

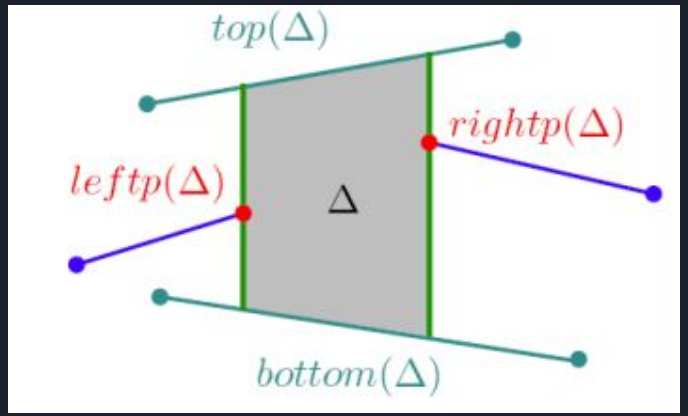
Dany trapez może posiadać wiele sąsiadów (rys. ii), jednak gdy posiadamy zbiór odcinków w położeniu ogólnym to dany trapez posiada co najwyżej czterech sąsiadów (rys. i). Są to lewy górny i dolny sąsiad oraz prawy dolny i górny sąsiad. W takim przypadku możemy przechowywać dla trapezu wskaźniki do co najwyżej 4 sąsiadów.





Struktura przechowująca Trapez

Każdy trapez posiada w sobie dwa odcinki: górny - top(Δ) i dolny - bottom(Δ), z których każdy jest reprezentowany przez dwa wierzchołki. Oprócz tego struktura przechowuje lewy - leftp(Δ) i prawy - rightp(Δ) wierzchołek trapezu.



Wierzchołki leftp(Δ) i rightp(Δ) mogą znajdować się w różnych pozycjach względem top(Δ) i bottom(Δ)

Pozycje leftp(Δ) względem top(Δ) i bottom(Δ)

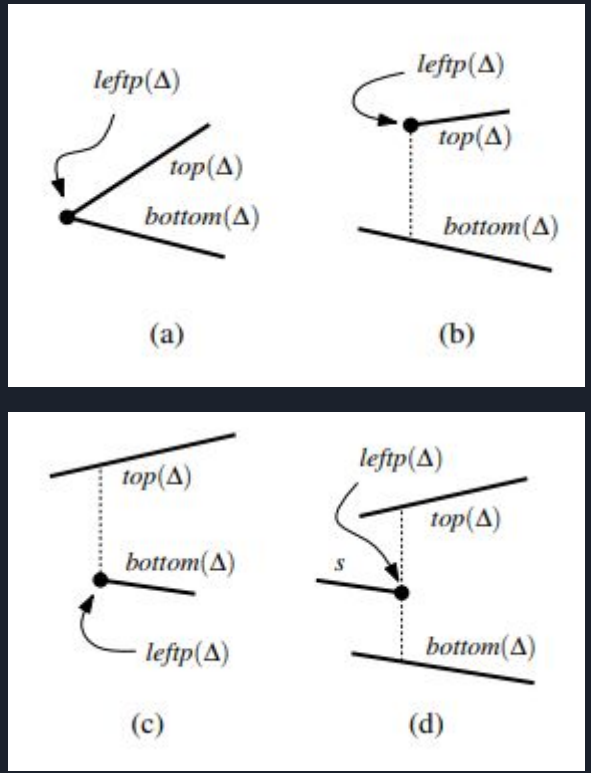
Mamy 4 możliwości wystąpienia lewego wierzchołka. Może się on pojawić pomiędzy odcinkami top i bottom lub znajdować się w tych samych miejscach co odcinki.

Jeśli dany wierzchołek leży pomiędzy odcinkami top i bottom musimy poprowadzić prostą, które ma swoje końce w momencie przecięcia z tymi odcinkami.

Gdy dany wierzchołek leży na lewym końcu odcinka top (bottom), wystarczy obliczyć wartość odcinka top(bottom) w punkcie o współrzędnej x lewego wierzchołka oraz poprowadzić w tym samym punkcie prostą pionową. Prosta ta ma końce na odcinku top i bottom.

Gdy odcinek znajduje się na końcach obydwu odcinków, nie prowadzimy wtedy żadnej linii

Dla prawego wierzchołka sytuacja jest analogiczna jak dla lewego



Randomizowany algorytm przyrostowy konstrukcji T(S)

1. Dane wejściowe algorytmu

   Aby algorytm działał poprawnie musi otrzymać prawidłowe dane wejściowe. Danymi przyjmowanymi przez algorytm jest zbiór odcinków $S = \{s_1, s_2, ..., s_n\}$ na płaszczyźnie dwuwymiarowej w położeniu ogólnym.

2. Wynik

   Algorytm jako wynik działania tworzy mapę trapezową T(S) oraz zwraca strukturę przeszukiwań D dla T(S) w postaci grafu przeszukiwań.

Pseudokod

Na początku przyjmujemy losową permutację odcinków $S = \{s_1, s_2, ..., s_n\}$ w położeniu ogólnym.

1. Stwórz strukturę danych dla prostokąta zewnętrznego, który zawiera w sobie wszystkie odcinki z S. Zainicjalizuj prostokąt do struktury mapy trapezowej T i struktury przeszukiwań D.
2. **For** i ← 1 **to** n **do**
3. Znajdź zbiór trapezów $\Delta_0, \Delta_1, ..., \Delta_k$ z struktury T, które przecinają odcinek s_i
4. Usuń $\Delta_0, \Delta_1, ..., \Delta_k$ z struktury T i zastąp je przez nowo utworzone trapezy, które pojawiły się po dodaniu odcinka s_i
5. Usuń liście $\Delta_0, \Delta_1, ..., \Delta_k$ z struktury D oraz stwórz liście dla nowych trapezów. Połącz nowe liście z istniejącymi węzłami wewnętrznymi, dodając kilka nowych węzłów

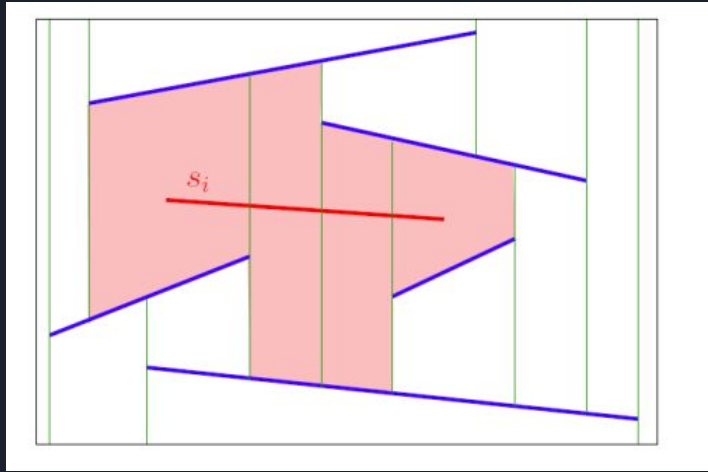
Strefa Si

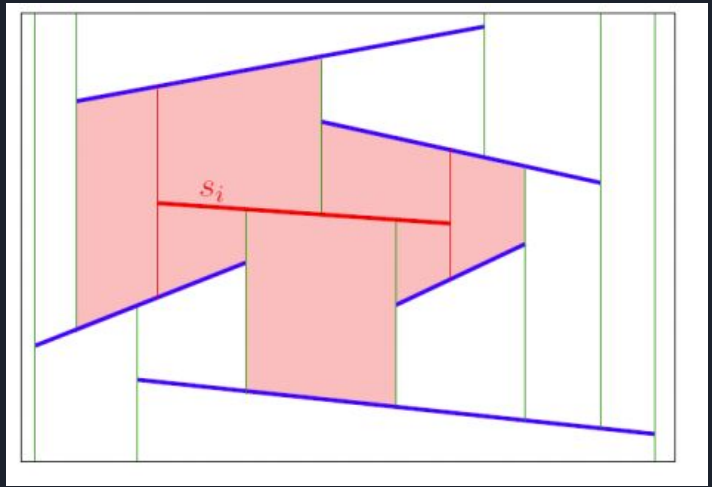
Strefę dla odcinka s_i w $T(S_{i-1})$ i w $T(S_i)$ tworzą wszystkie trapezy przecinające s_i.

Dla $T(S_{i-1})$ jest to suma wszystkich trapezów które zostaną usunięte.

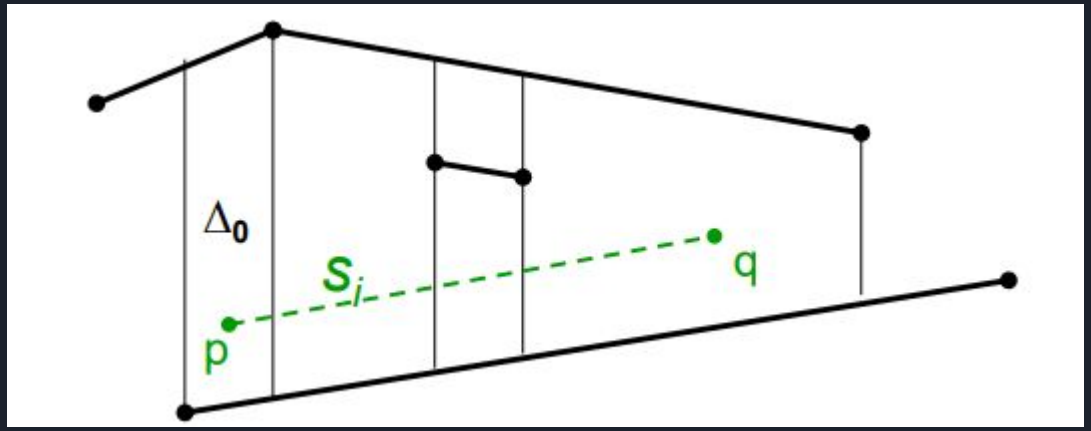
Dla $T(S_i)$ jest to suma wszystkich trapezów które zostaną stworzone.

Strefy dla $T(S_{i-1})$ i w $T(S_i)$ są jednakowe pod względem kształtu i powierzchni. Różnią się jedynie ułożeniem i ilością trapezów zawierających w sobie.





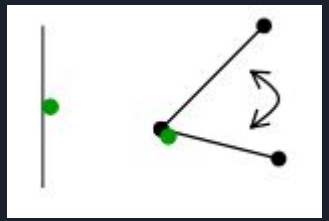
Algorytm wyznaczania strefy dla s_i



Przeszukaj dla p strukturę D aż do znalezienia trapezu Δ_0 w którym znajduje się początek odcinka s_i.

Jeśli w trakcie przeszukiwania znajdziemy punkt p w strukturze (grafie poszukiwań) D. Musimy wtedy uwzględnić dwie rzeczy:

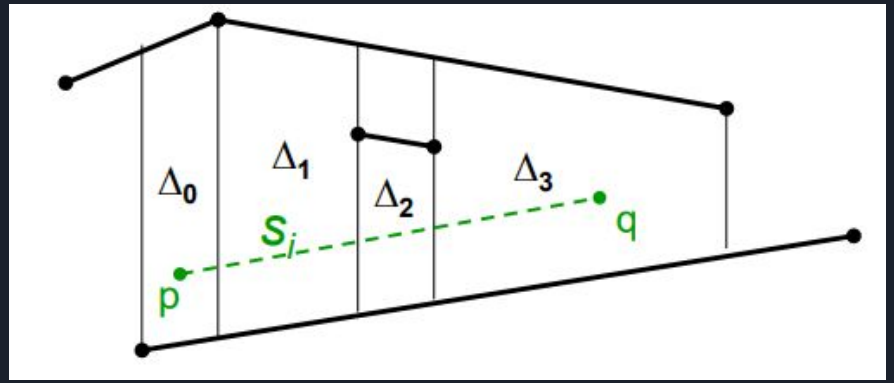
- Jeżeli punkt p leży na prostej pionowej, wtedy przyjmujemy, że punkt p leży po prawej stronie punktu znajdującego się w strukturze D
- Jeżeli punkt p jest początkiem z innym odcinkiem s, wtedy porównujemy ich nachylenie. Jeśli nachylenie s_i jest mniejsze od nachylenia odcinka s, wtedy punkt p leży poniżej s



Mając na uwadzę te dwie rzeczy pobierzmy trapez Δ_0 z struktury D

Po wykonaniu powyższych kroków możemy rozpocząć algorytm szukania strefy dla odcinka s_i

Algorytm wyznaczania strefy dla s_i



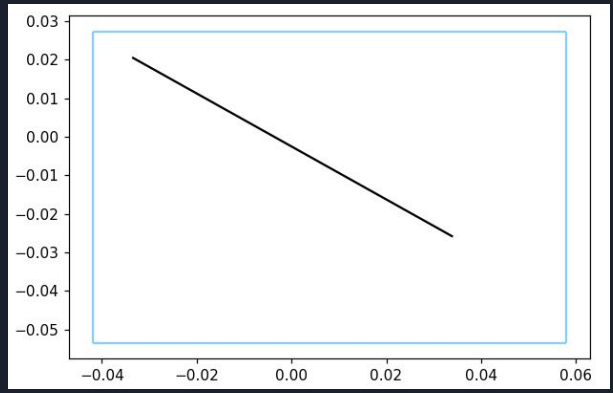
j ← 0
while q leży na prawo od rightp(Δ_j)
    do if rightp(Δ_j) leży powyżej s_i
        then niech Δ_{j+1} będzie dolnym prawym sąsiadem Δ_j
        else niech Δ_{j+1} będzie górnym prawym sąsiadem Δ_j
    j ← j + 1
return $\Delta_0, \Delta_1, \ldots, \Delta_k$

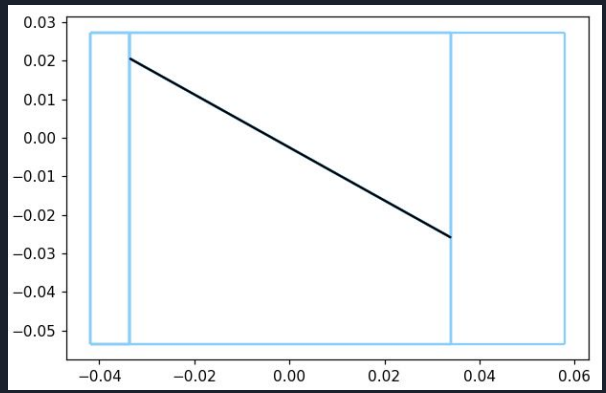
Zastępowanie trapezu w T(S)

1. Usuwamy Δ z T

2. Zastępujemy przez odpowiednią ilość nowych trapezów

3. Aktualizujemy informacje dla trapezów o sąsiadach, bottom(Δ), top(Δ), leftp(Δ), rightp(Δ)

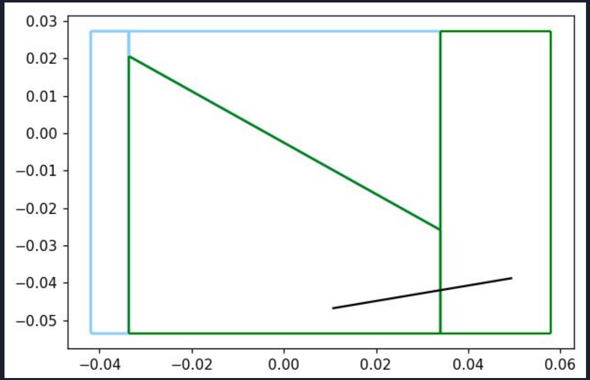


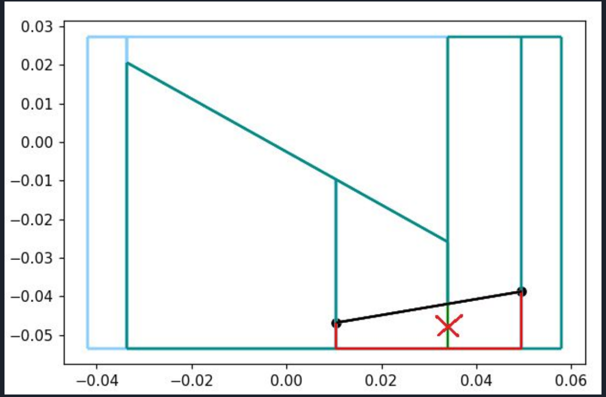


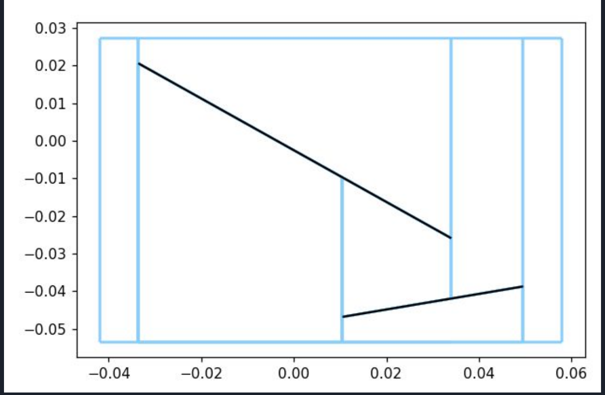
Wstawianie rozszerzeń pionowych

1. Wstawiamy rozszerzenia pionowe przechodzące przez końce s
2. Skracamy rozszerzenia pionowe, aby stykały się z S
3. Na koniec aktualizujemy informacje o sąsiadach dla zmienionych trapezów

Oczekiwany czas konstrukcji mapy trapezowej to O(*n log n*)



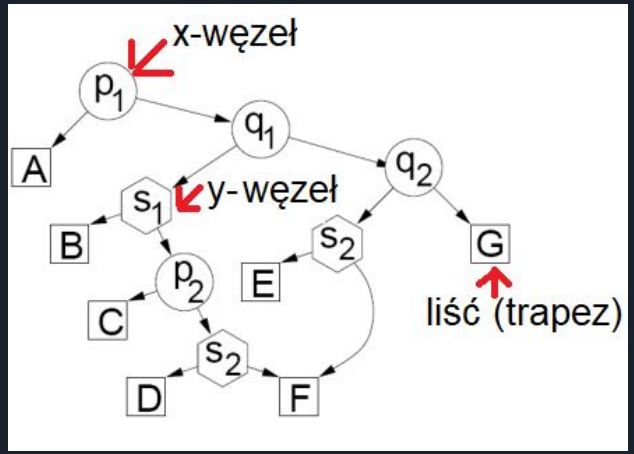




Graf wyszukiwania

Graf wyszukiwań to struktura która w liściach zawiera wyłącznie wskaźnik do odpowiadającemu mu trapezu w T(S). Węzły wewnętrzne są podzielone na dwa rodzaje:

- x-węzeł - przechowuje współrzędne wierzchołka
- y-węzeł - przechowuje wskaźnik do odcinka



Każdy trapez z mapy trapezowej T(S) ma wskaźnik do liścia w strukturze D, który jest jego odpowiednikiem.

Model poruszania się w drzewie:

- Gdy punkt leży na lewo od prostej pionowej przechodzącej przez wierzchołek w x-węźle, wtedy kierujemy go na lewo.
- W przeciwnym przypadku kierujemy wierzchołek na prawo
- Gdy punkt q leży powyżej odcinka w y-węźle wtedy kierujemy go na lewo. W przeciwnym przypadku kierujemy wierzchołek na prawo.

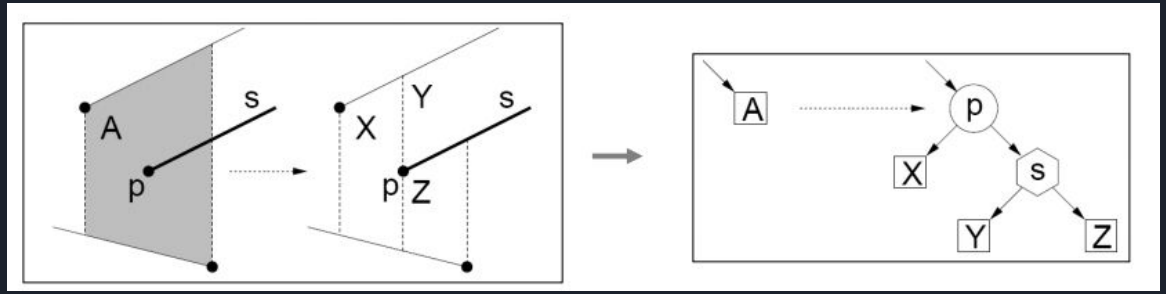
Oczekiwany rozmiar grafie przeszukiwań to O(n), a oczekiwany czas konstrukcji grafu to - O(n log n) (wliczając w tym konstrukcje mapy trapezowej) .

Konstrukcja grafu wyszukiwania

Gdy usuwamy trapez z grafu to zastępujemy liść reprezentujący trapez kolejną częścią drzewa

W zależności od ilości końców odcinka s zawierających się w usuwanym trapezie, postępujemy następująco:

a) Gdy usuwany trapez zawiera jeden koniec odcinka s

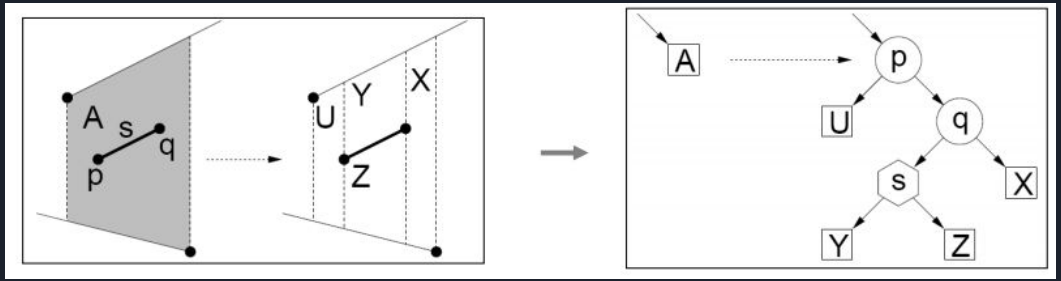


Pojedynczy trapez A zastępujemy przez trzy trapezy X,Y,Z. Wstawiamy je według modelu poruszania się w drzewie, opisanego na wcześniejszym slajdzie, dodając odpowiednio x-węzły i y-węzły reprezentujące początek i koniec odcinka oraz cały odcinek do struktury.

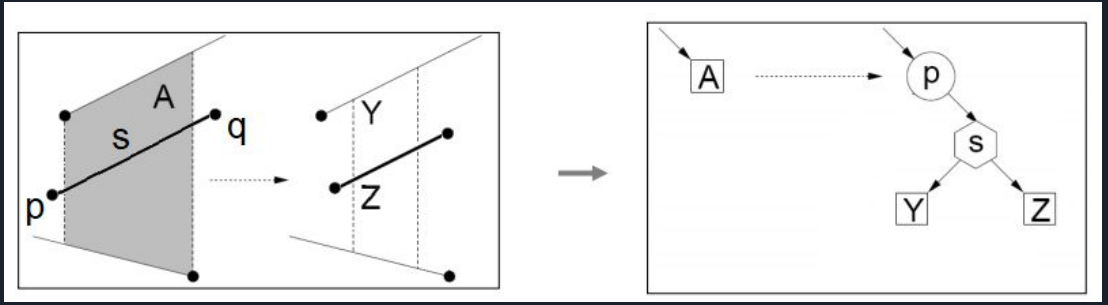
Konstrukcja grafu wyszukiwania

b) Gdy usuwany trapez zawiera dwa końce odcinka s

Pojedynczy trapez A jest zastępowany przez cztery trapezy U,X,Y i Z



c) Gdy odcinek przecina trapez, ale ten nie zawiera jego końców

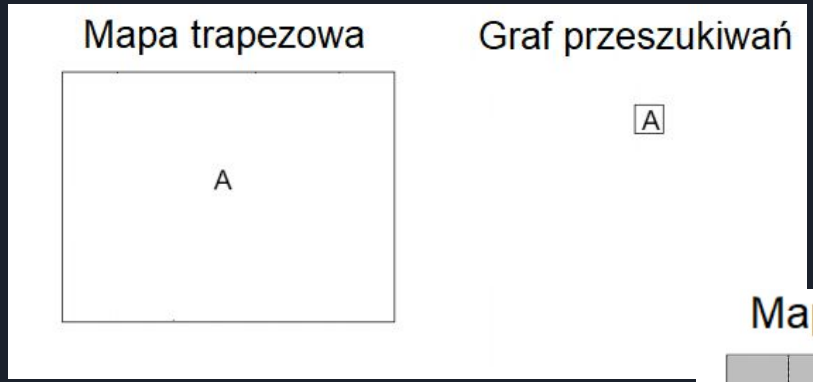


Pojedynczy trapez A jest zastępowany przez dwa trapezy Y i Z

We wszystkich przypadkach metoda dodawania trapezów do grafu, jest taka sama jak metoda opisana na slajdzie wyżej.

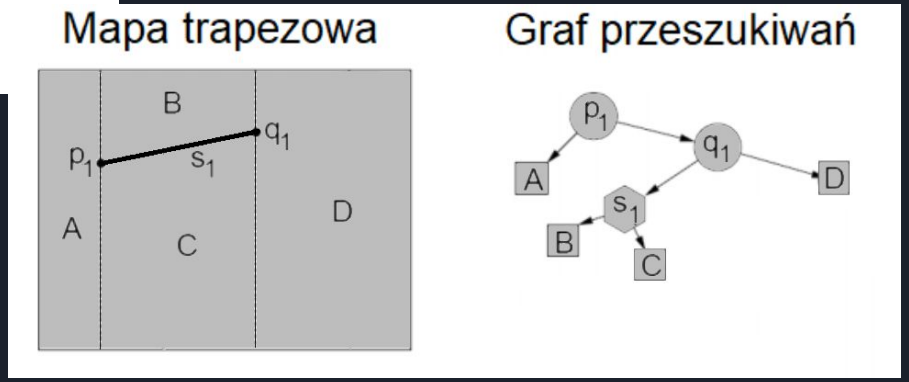
Przykład konstrukcji

Na początku mamy jeden trapez obejmujący wszystkie linie . Jest to zewnętrzny prostokąt



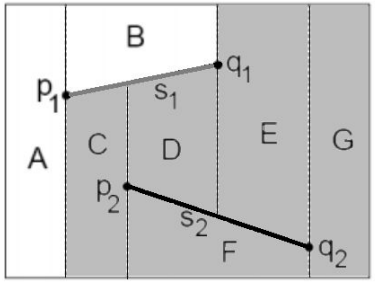
W kolejnych krokach modyfikujemy struktury

na skutek dodania kolejnych odcinków

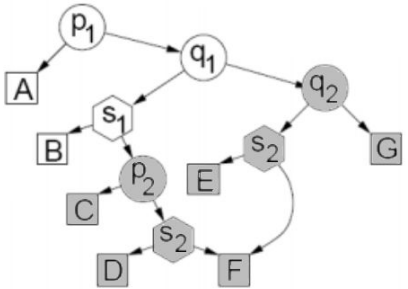


Przykład konstrukcji (cd.)

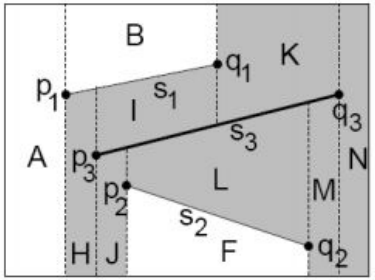
Mapa trapezowa



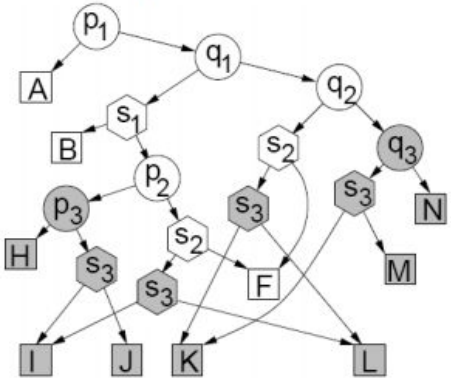
Graf przeszukiwań



Mapa trapezowa



Graf przeszukiwań



Wyszukanie punktu w mapie

Dane: zadany punkt q zdefiniowany poprzez swoje współrzędne

Szukane: lokalizacja punkt q na mapie trapezowej

Algorytm:

Jeśli element jest x-węzłem:

- Jeśli szukany punkt leży po lewej stronie prostej pionowej przechodzącej przez punkt w x-węźle: przejdź w dół do lewego potomka
- w przeciwnym przypadku przejdź w dół do prawego potomka

Jeśli element jest y-węzłem:

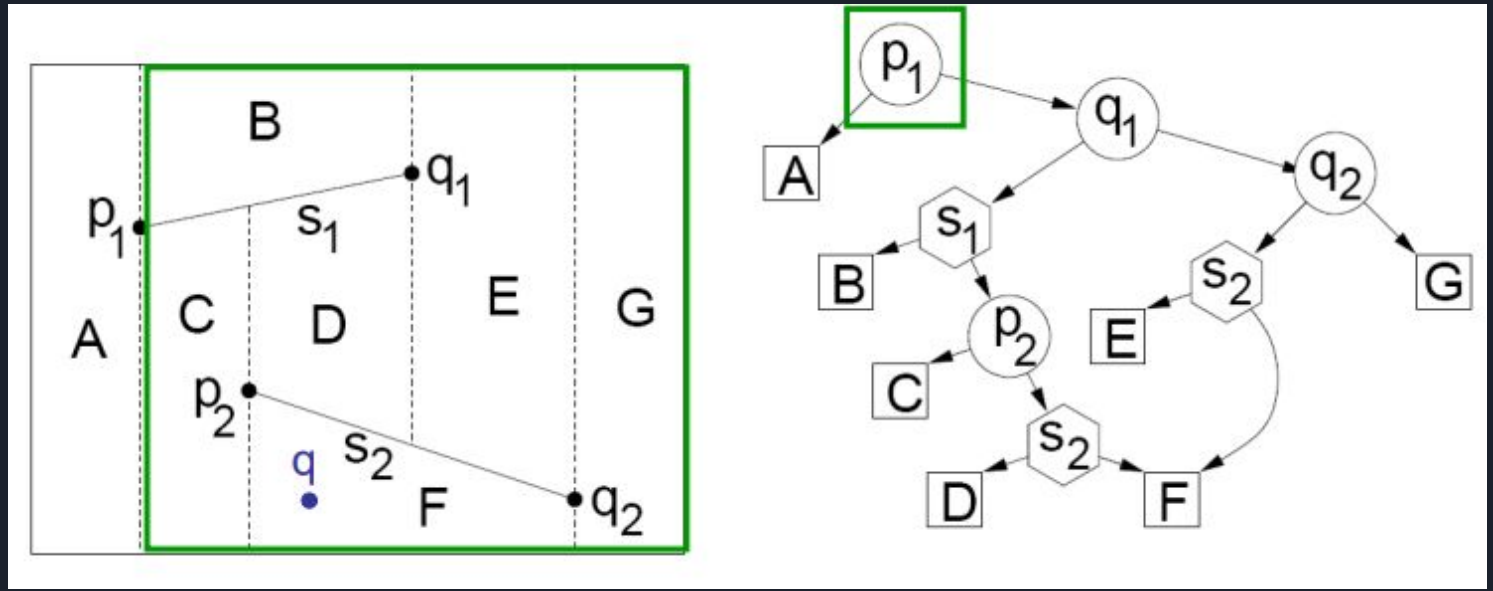
- Jeśli szukany punkt leży poniżej prostej w y-węźle: przejdź w dół do lewego potomka
- w przeciwnym przypadku przejdź w dół do prawego potomka

Jeśli bieżący element jest liściem, zakończ algorytm

Oczekiwany czas lokalizacji położenie dowolnego punktu na mapie trapezowej w czasie to O(log n).

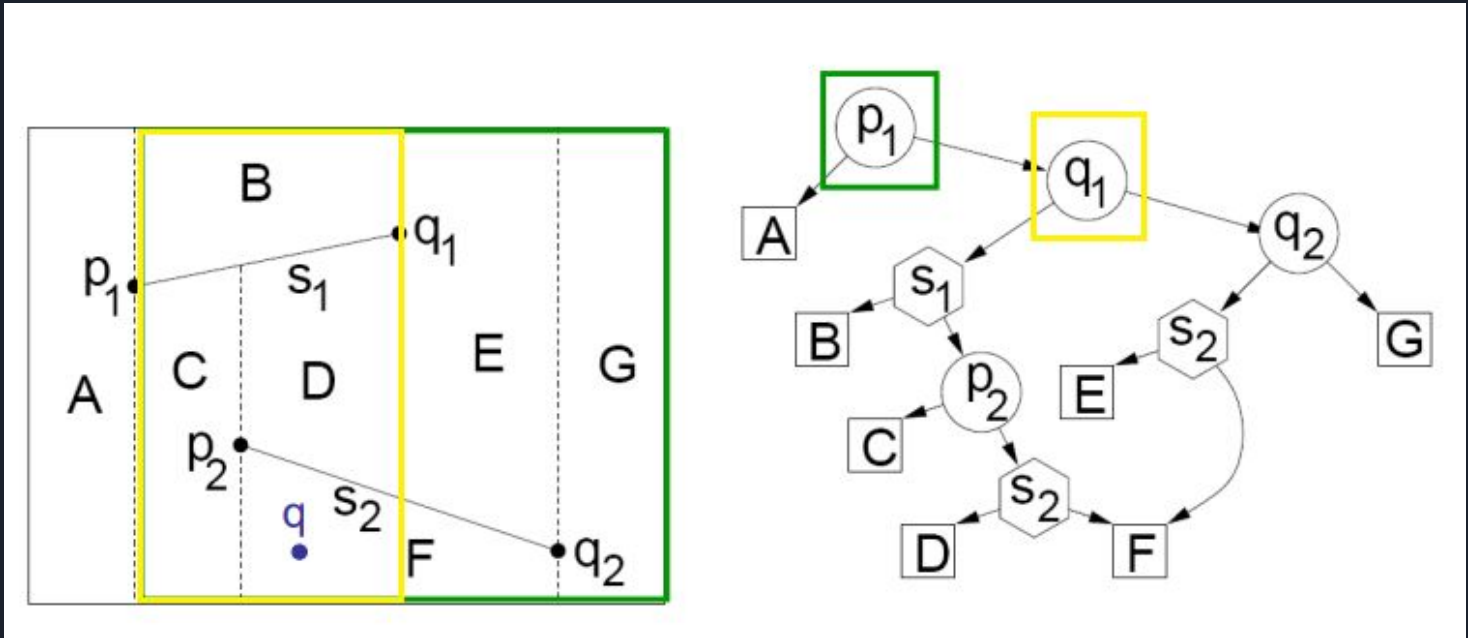
Wyszukanie punktu w grafie wyszukiwań

Pytamy punkt p1 czy szukany punkt q leży po jego lewej stronie. Dostajemy odpowiedź negatywną.Algorytm kieruje nas do swojego prawnego węzła



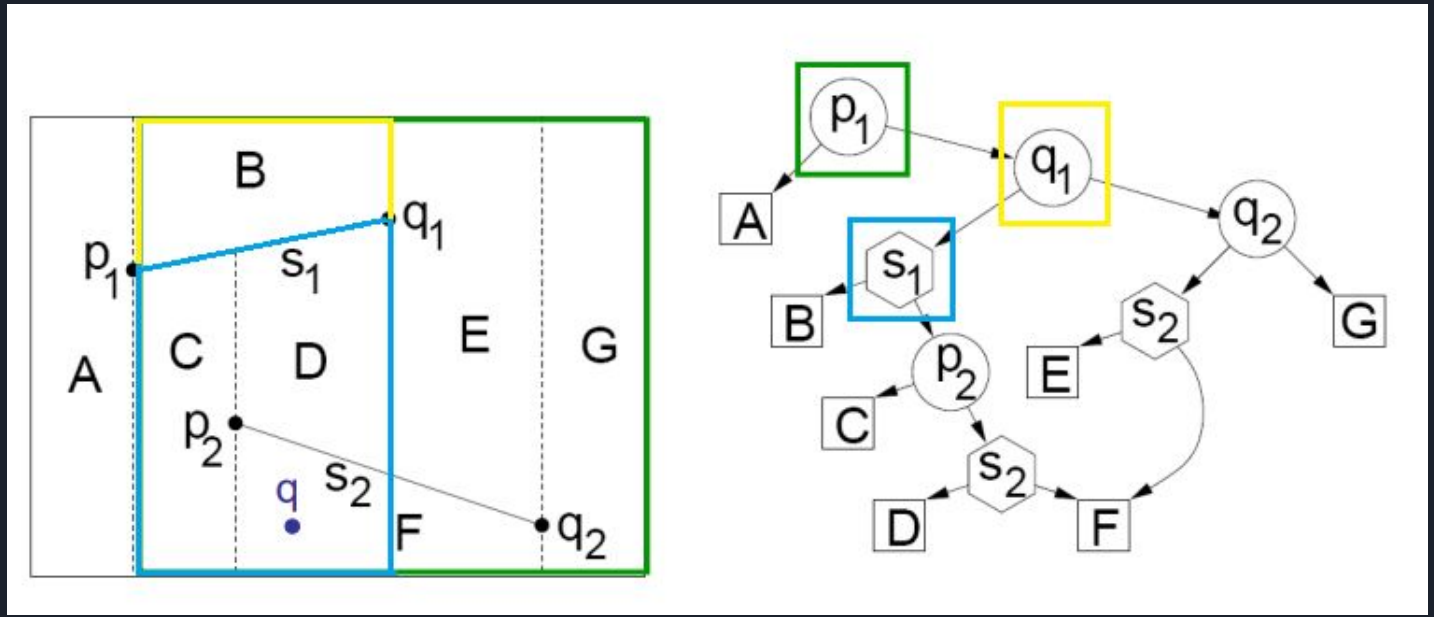
Wyszukanie punktu w grafie wyszukiwań

W kolejnym kroku pytamy punkt q1 czy szukany punkt q leży po jego lewej stronie. Dostajemy odpowiedź pozytywną .Algorytm kieruje nas do swojego lewego węzła



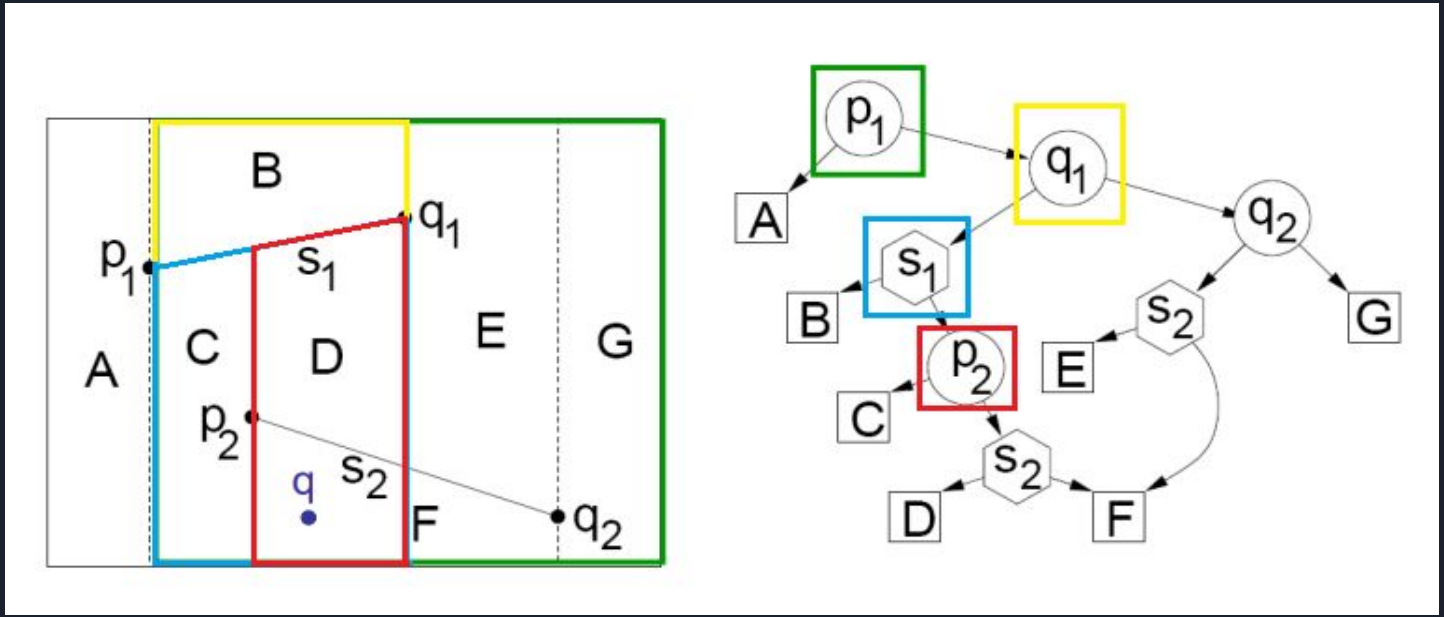
Wyszukanie punktu w grafie wyszukiwań

Następnie pytamy odcinek s1 czy szukany punkt q leży powyżej niego. Dostajemy odpowiedź negatywną.Algorytm kieruje nas do swojego prawego węzła



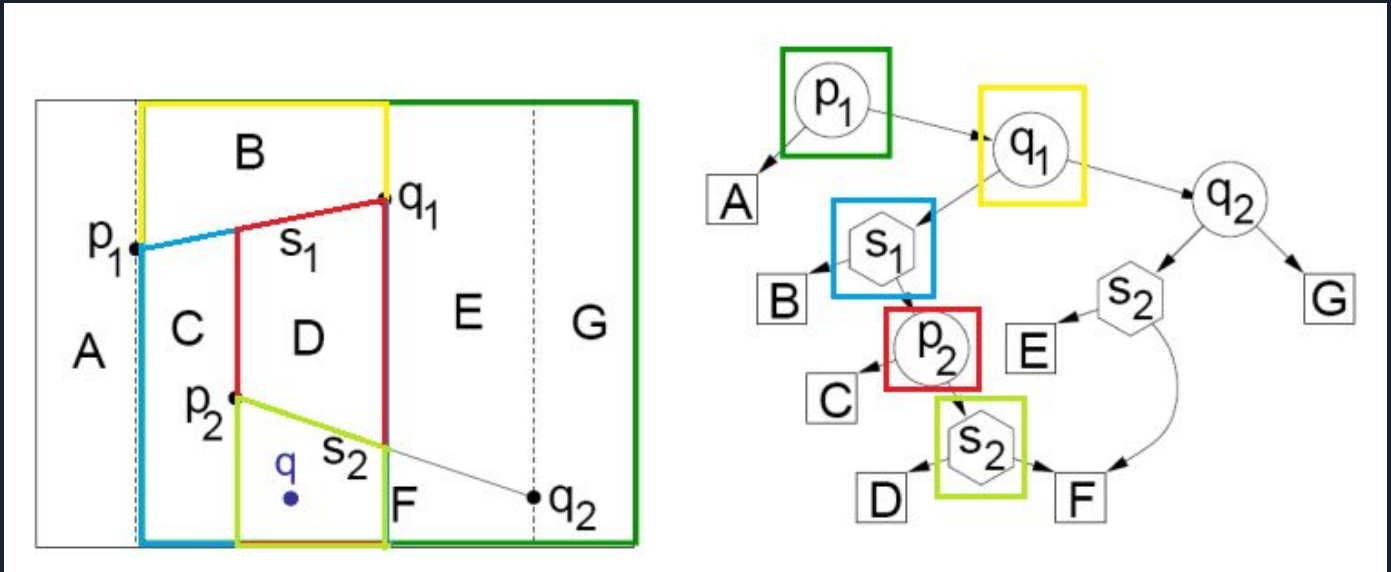
Wyszukanie punktu w grafie wyszukiwań

W kolejnym kroku pytamy punkt p2 czy szukany punkt q leży na lewo od niego. Dostajemy odpowiedź negatywną. Algorytm kieruje nas do swojego prawego węzła



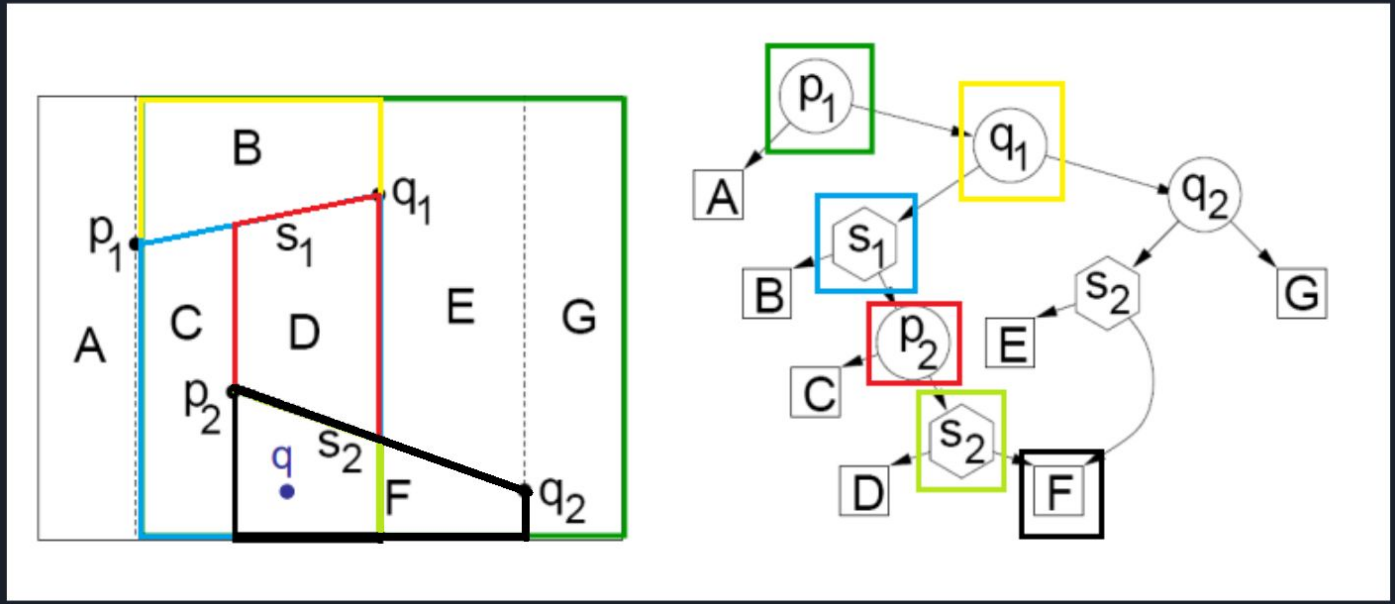
Wyszukanie punktu w grafie wyszukiwań

Następnie pytamy odcinek s2 czy szukany punkt q leży nad nim. Dostajemy odpowiedź negatywną. Algorytm kieruje nas do swojego prawego węzła



Wyszukanie punktu w grafie wyszukiwań

Gdy dojdziemy do liścia, pobieramy nasz trapez i zwracamy odpowiedź.



Bibliografia

https://upel2.cel.agh.edu.pl/wiet/pluginfile.php/82098/mod_resource/content/1/wyklad6_lokpkt.pdf

Mark de Berg, Otfried Cheong, Marc van Kreveld, Mark Overmars - Computational Geometry - Algorithms and Applications Third Edition

https://www.ti.inf.ethz.ch/ew/lehre/CG12/lecture/Chapter%209.pdf?fbclid=IwAR2zMABGfnhie063BRSw0EWnCDkHZ3uWHCo36jja54xd3dELgmcdpGVjAwI

https://janrollmann.de/projects/thesis/

https://github.com/ad8454/TrapezoidalMaps

https://github.com/fenderglass/trapezoid_map

http://cglab.ca/~cdillaba/comp5008/trapezoid.html

http://www.cs.umd.edu/class/spring2020/cmsc754/Lects/lect08-trap-map.pdf

https://users.dimi.uniud.it/~claudio.mirolo/teaching/geom_comput/presentations/trapezoidal_map.pdf

Koniec

Dziękujemy za uwagę