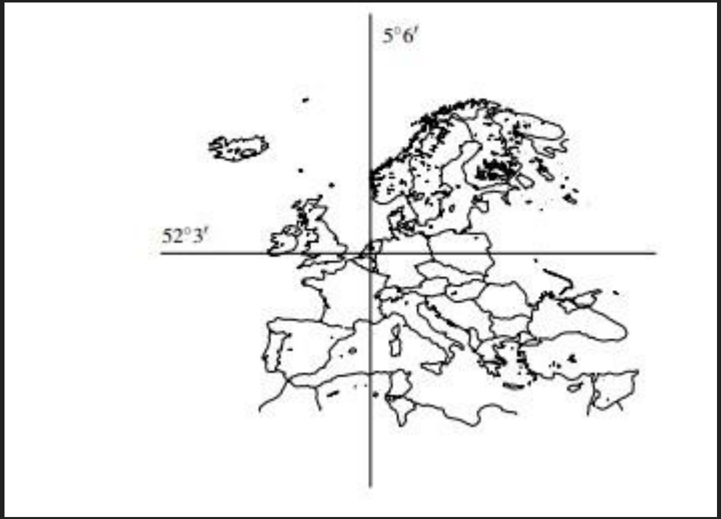
Lokalizacja punktu metodą trapezową

Łukasz Powęska
Samuel Hełdak

Cel algorytmu



Efektywne znalezienie wielokąta w którym znajduje się zadany punkt.

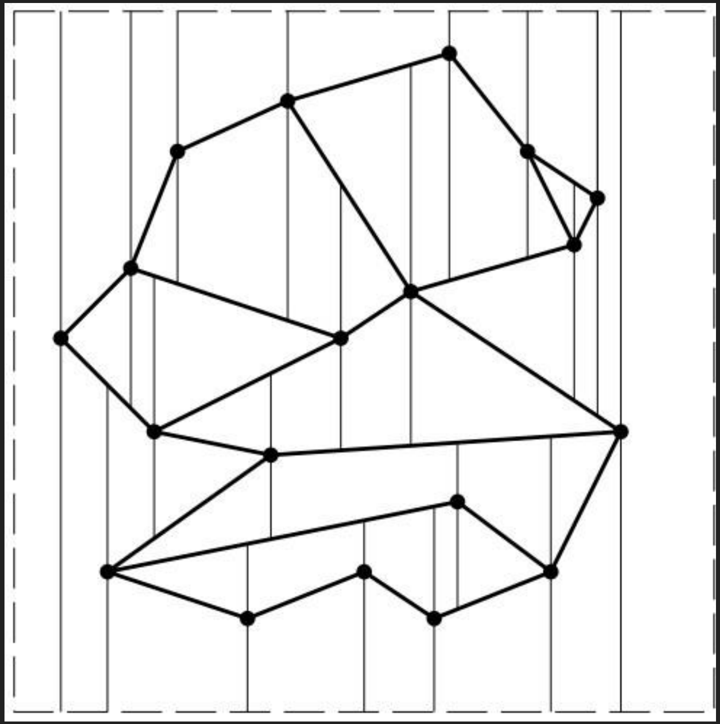
Jak to zrobić?

- Tworzymy odpowiednie struktury, które pozwolą na lokalizację punktu w czasie proporcjonalnym do O(logN), gdzie N jest liczbą wielokątów tworzących podział poligonowy przestrzeni

- Używamy funkcji wyszukującej w strukturach wielokąt zawierający zadany punkt

Struktury

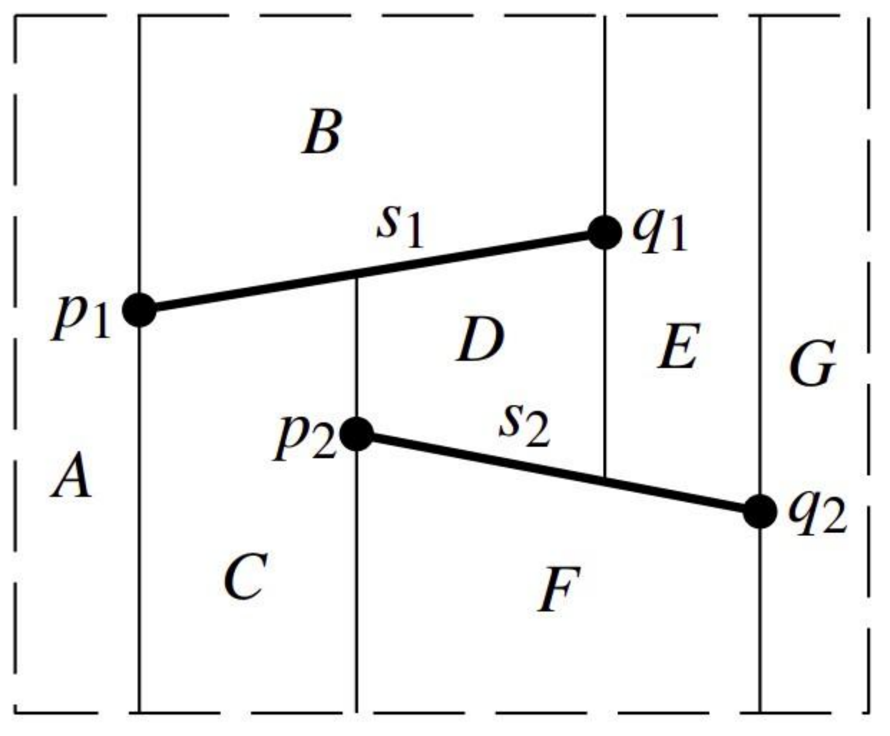
- Dążymy do stworzenia podziału przestrzeni na trapezy (i trójkąty) jak na rysunku obok.
- Każdy z trapezów jest ograniczony 2 niepionowymi odcinkami oraz 1 lub 2 pionowymi odcinkami (dodanymi tam gdzie są wierzchołki wielokątów i poprowadzone do najbliższych 2 odcinków).
- Potrzebujemy w tym celu 2 struktur: mapy trapezowej oraz struktury przeszukiwań



Mapa trapezowa (T)

- Od każdego trapezu możemy się dostać do jego lewego (lub lewego górnego i lewego dolnego) i prawego (lub prawego górnej i prawego dolnego) sąsiada. Np. trapez E ma prawego sąsiada G, lewego górnego sąsiada B i lewego dolnego sąsiada D.

- Dodatkowo każdy trapez posiada informację o niepionowych odcinkach między którymi jest zawarty oraz o punktach które wyznaczają pionowe odcinki go ograniczające.

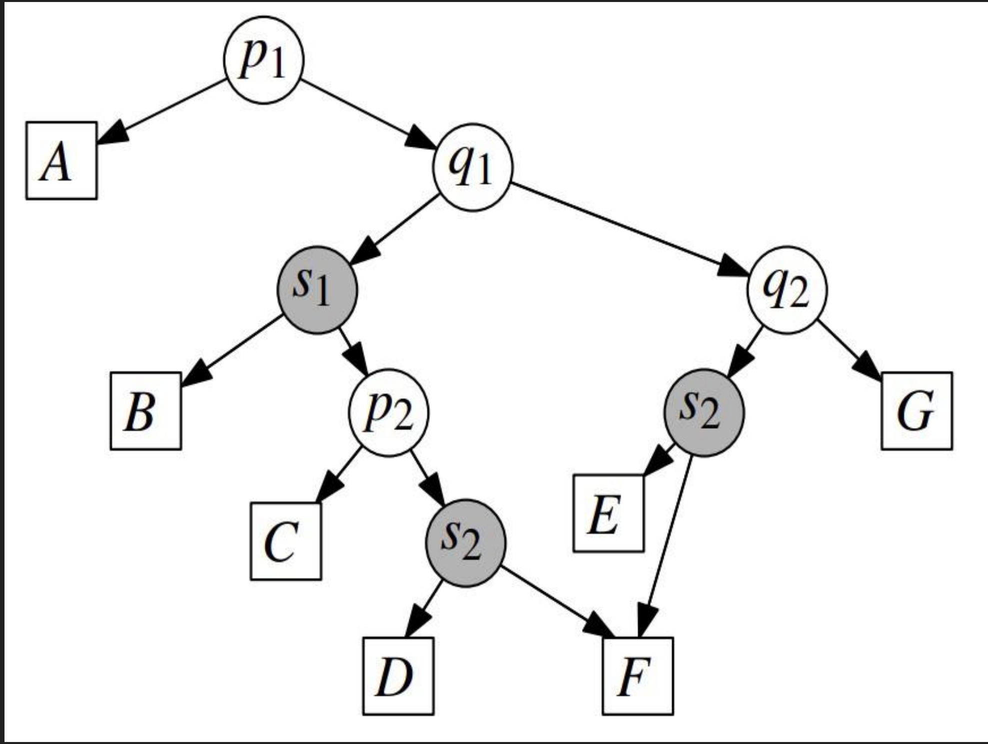


Struktura przeszukiwań (D)

- Struktura przeszukiwań to acykliczny graf skierowany zawierający 3 typy wierzchołków: punkt (białe koło), odcinek (szare koło) oraz trapez (kwadrat)

- Każdy wierzchołek poza trapezami ma 2 krawędzie wychodzące

- Istnieje dokładnie jeden wierzchołek do którego nie wchodzi żadna krawędź (wierzchołek początkowy). Do pozostałych wierzchołków nie będących trapezami wchodzi dokładnie jedna krawędź, a do trapezów co najmniej jedna.



Algorytm

TrapezoidalMap(S)

1. Determine a bounding box R that contains all segments of S, and initialize the trapezoidal map structure T and search structure D for it
2. Compute a random permutation s1, s2, …, sn of the elements of S.
3. **For** i <- 1 **to** n
4.   **do** Find the set Δ0, Δ1, …, Δk of trapezoids in T properly intersected by si
5.   Remove Δ0, Δ1, …, Δk from T and replace them by the new trapezoids that appear because of the insertion of si
6.   Remove the leaves for Δ0, Δ1, …, Δk from D, and create leaves for the new trapezoids. Link the new leaves to the existing inner nodes by adding some new inner nodes.

Budowanie Struktury Poszukiwań na podstawie Mapy Trapezowej

Legenda :

- Węzeł okrągły : punkt
- Węzeł kwadratowy : obszar (trapez, trójkąt) -> zawsze jest liściem w drzewie
- Węzeł trójkątny : odcinek

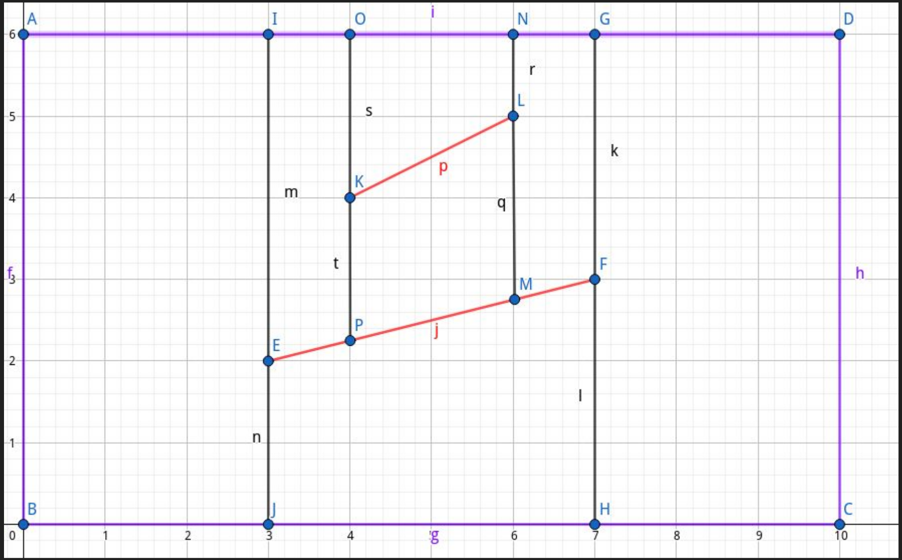
Dla punktu :

- Na lewo : obszar, linia znajdują się po lewej stronie punktu
- Na prawo : obszar, linia znajdują się po prawej stronie punktu

Dla odcinka:

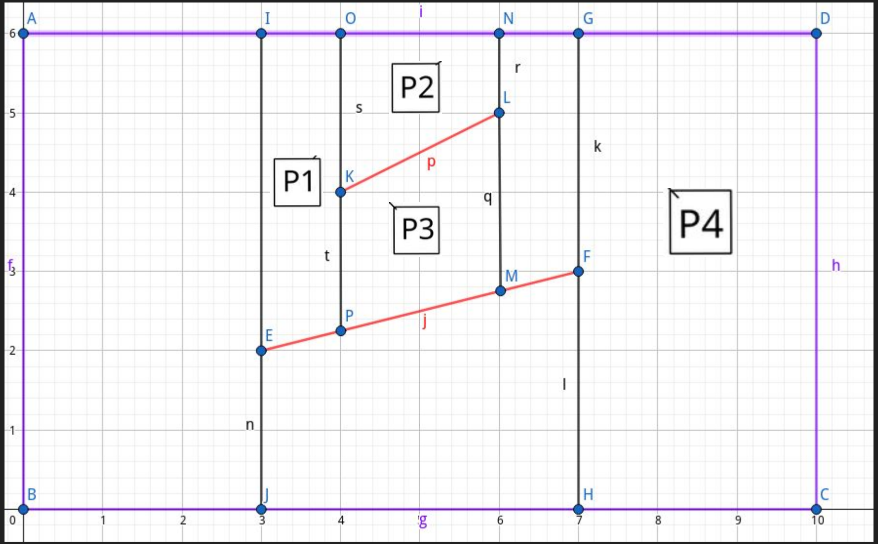
- Na lewo : znajduje się nad odcinkiem
- Na prawo : znajduję się pod odcinkiem

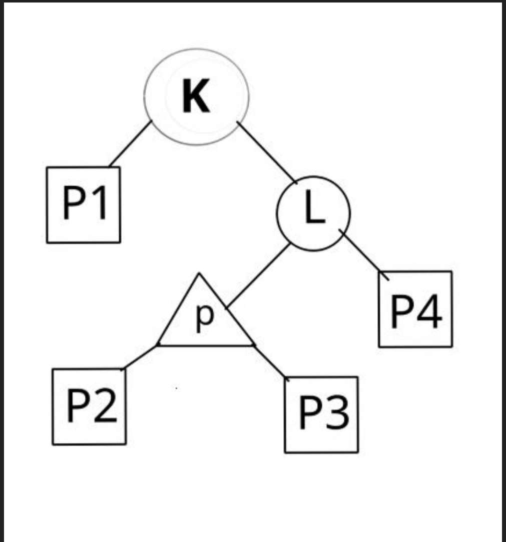
Sytuacja 1 : W wyniku dodania nowego odcinka do mapy stary obszar został podzielony na 4 nowe obszary.



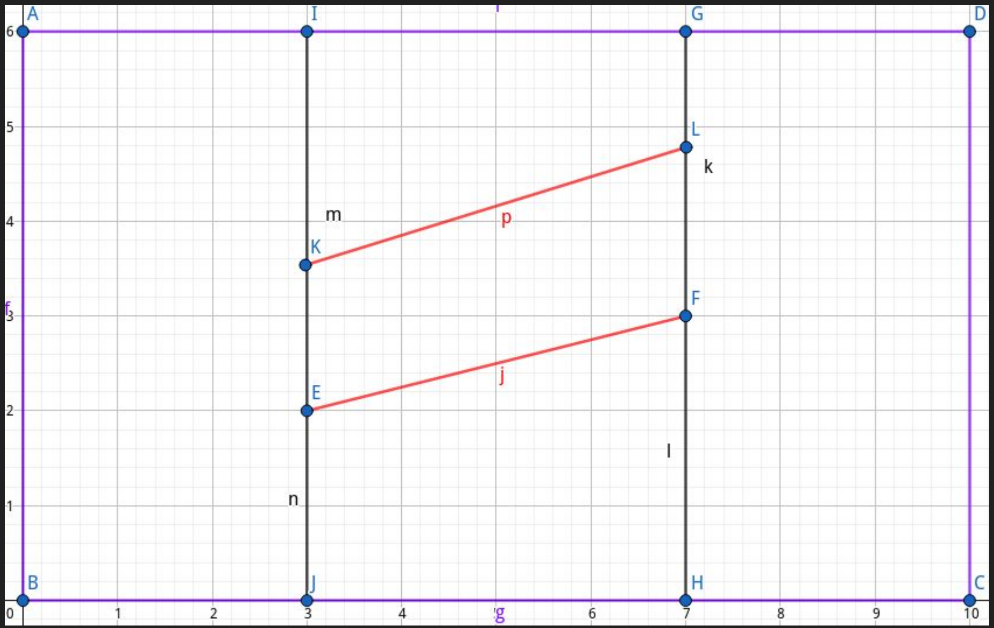
W wyniku dodania prostej “p” obszar znajdujący się ponad prostą “j” został podzielony na 4 nowe obszary.

W wyniku tego podziału obszar znajdujący się ponad prostą “j” zostanie zastąpiony przez poniższe poddrzewo



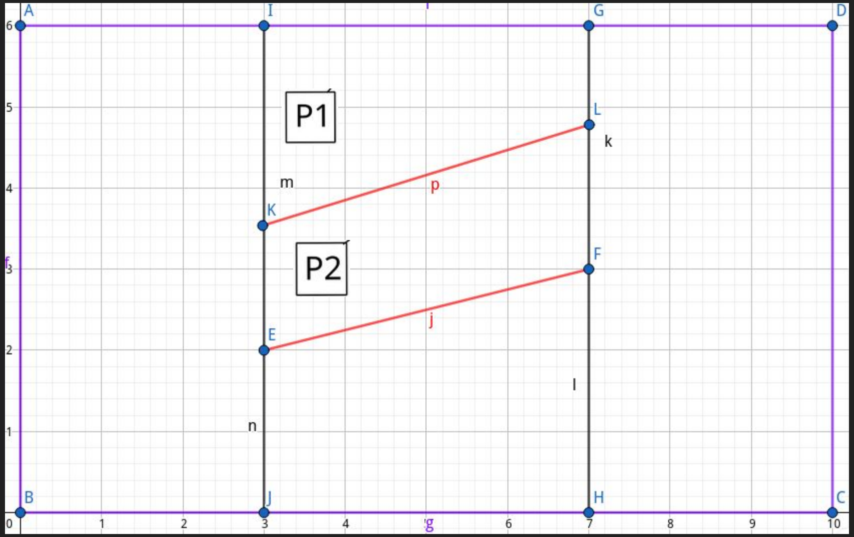


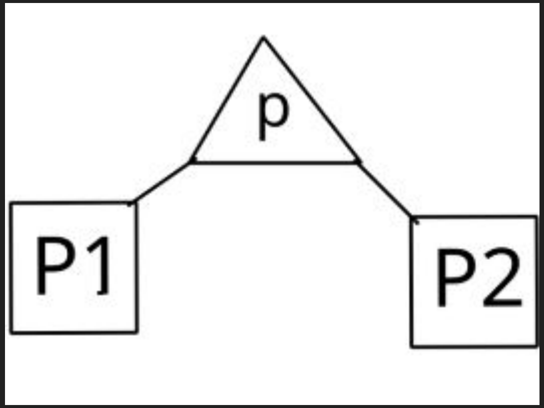
Sytuacja 2 : W wyniku dodania nowego odcinka do mapy stary obszar został podzielony na 2 nowe obszary.



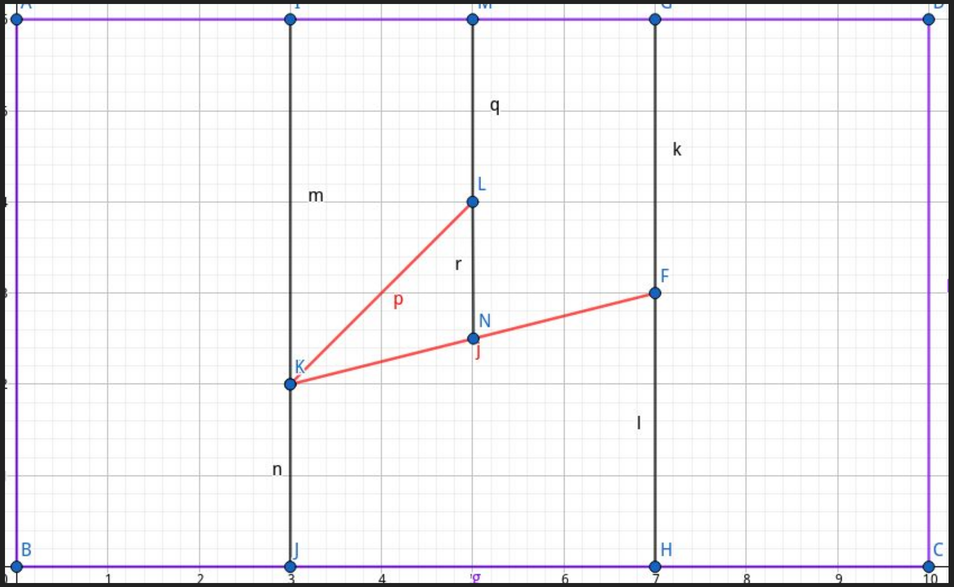
W wyniku dodania prostej “p” obszar znajdujący się ponad prostą “j” został podzielony na 2 nowe obszary.

W wyniku tego podziału obszar znajdujący się ponad prostą "j" zostanie zastąpiony przez poniższe poddrzewo



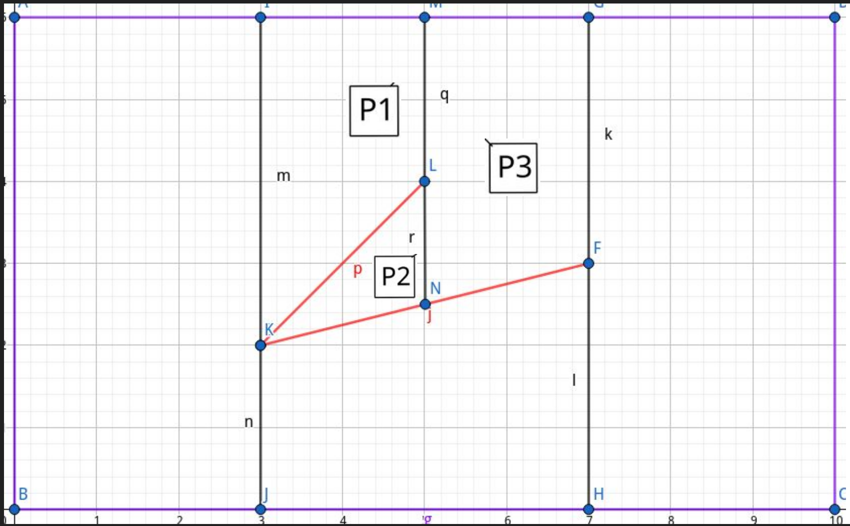


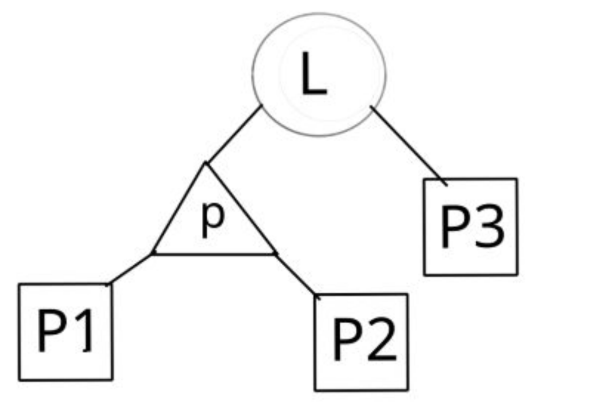
Sytuacja 3 : W wyniku dodania nowego odcinka do mapy stary obszar został podzielony na 3 nowe obszary.

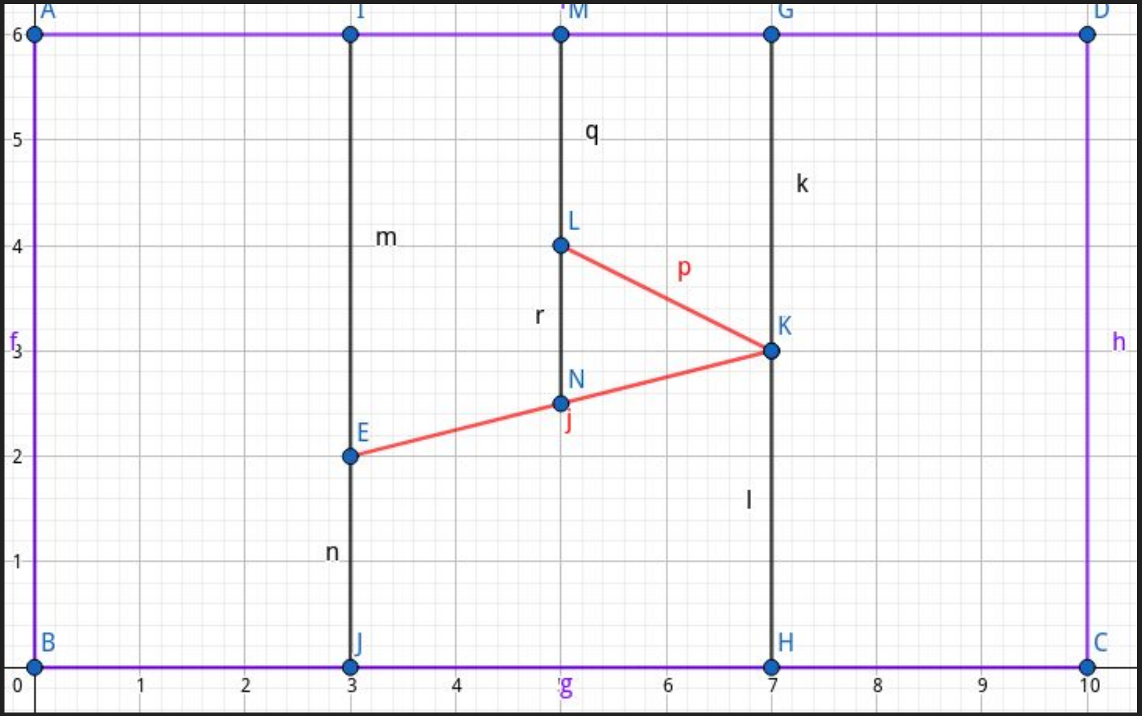


Przypadek 1 : odcinki stykają się w lewym końcem

W wyniku tego podziału obszar znajdujący się ponad prostą “j” zostanie zastąpiony przez poniższe poddrzewo

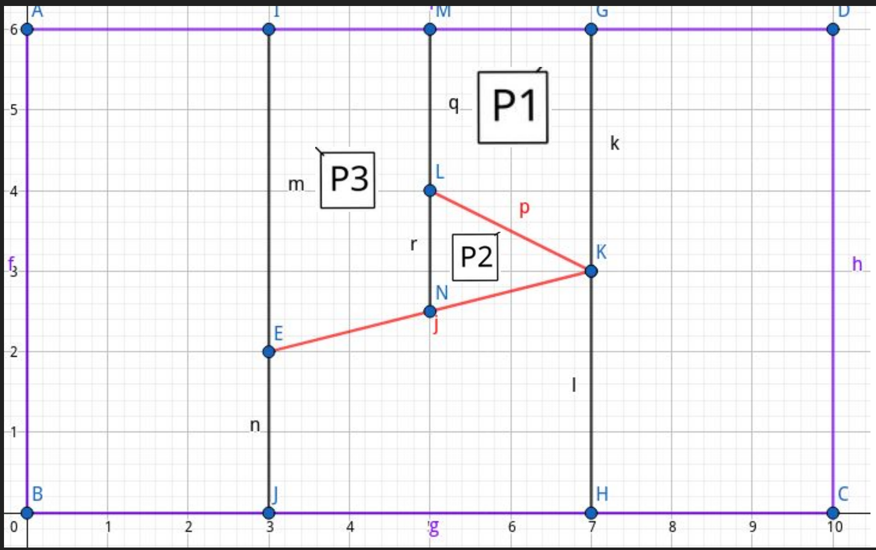


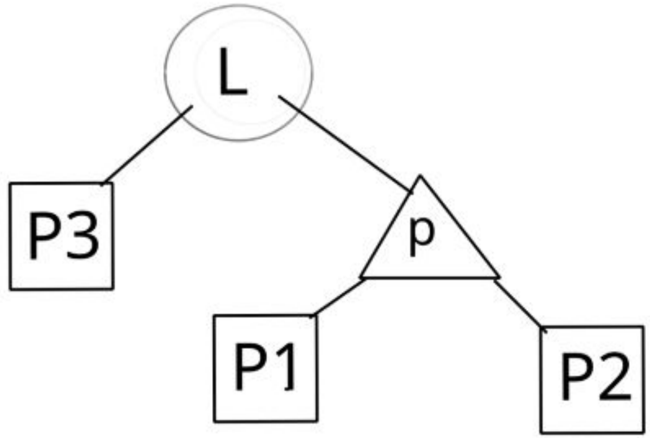


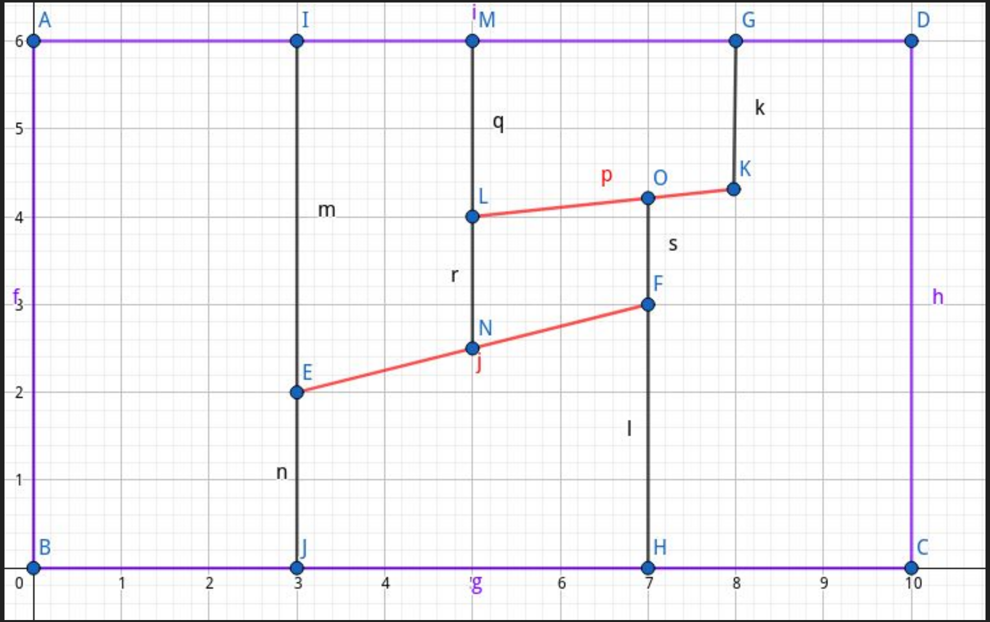


Przypadek 2 : odcinki stykają się w prawym końcem

W wyniku tego podziału obszar znajdujący się ponad prostą “j” zostanie zastąpiony przez poniższe poddrzewo

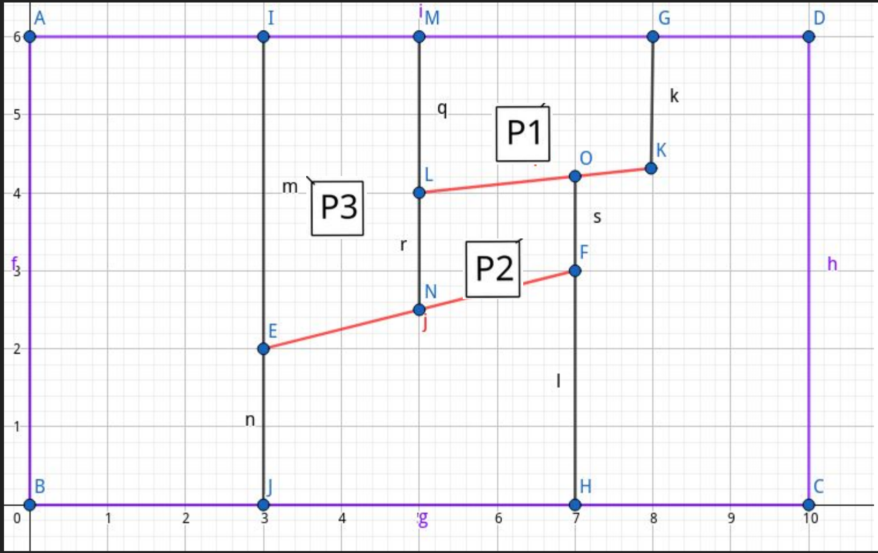


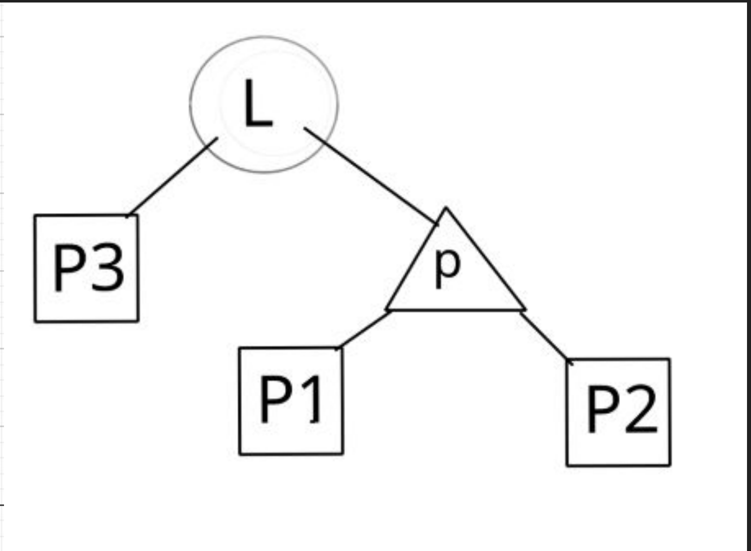


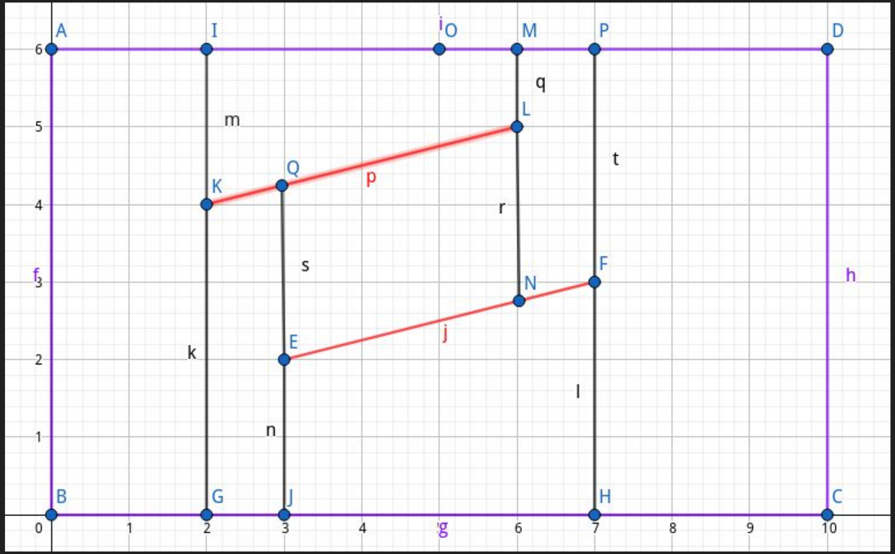


Przypadek 3 : koniec odcinka “p” (punkt K) znajduję się dalej niż punkt ograniczający “stary” obszar ze strony prawej

W wyniku tego podziału obszar znajdujący się ponad prostą "j" zostanie zastąpiony przez poniższe poddrzewo

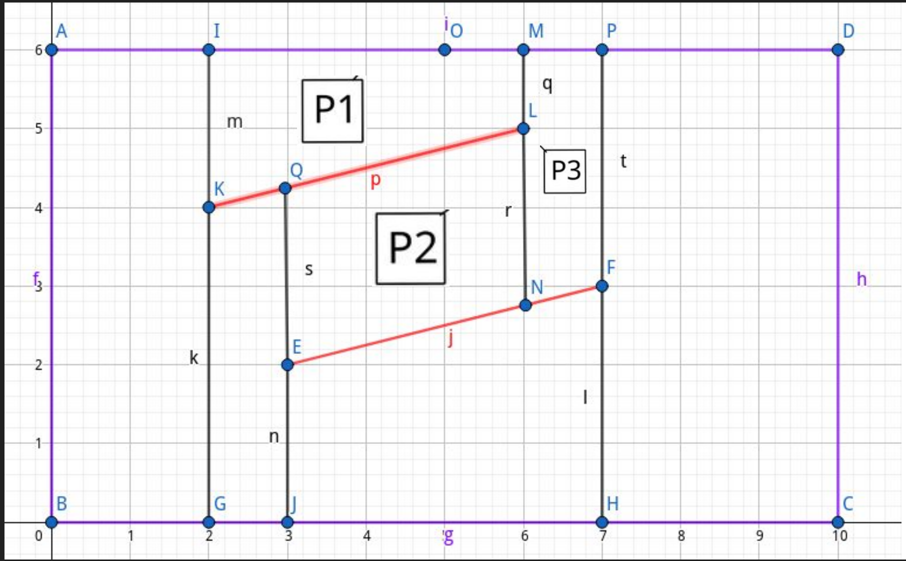


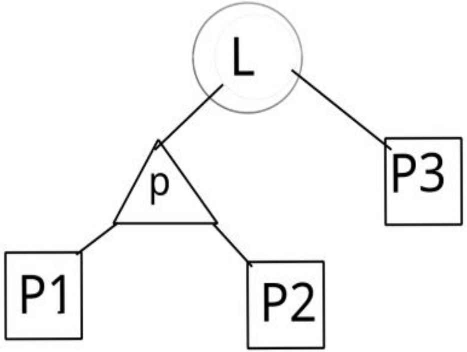




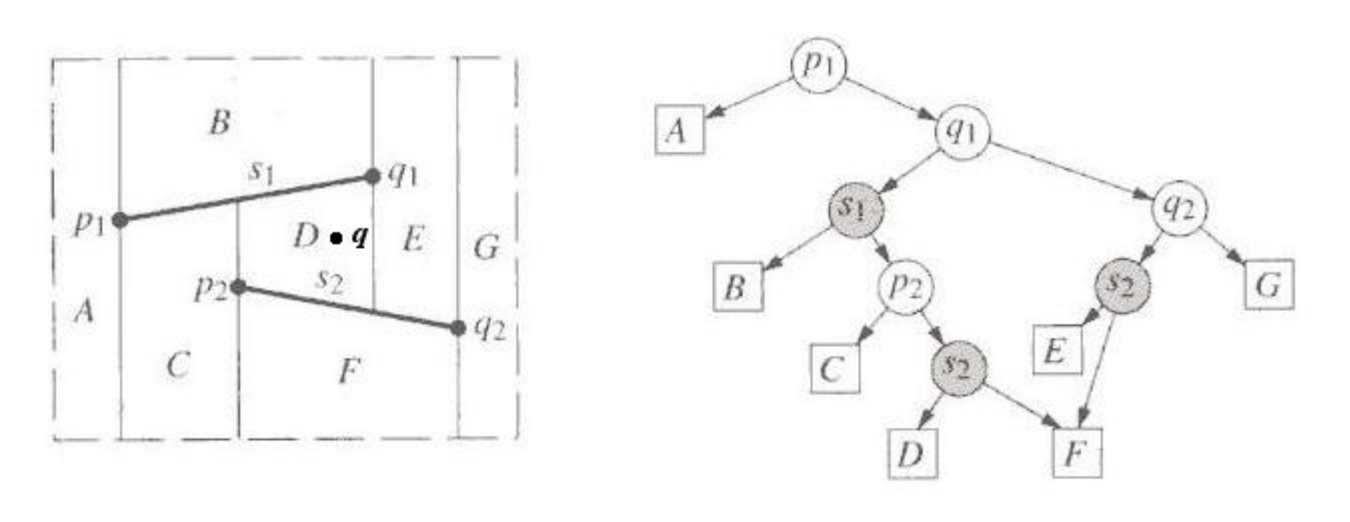
Przypadek 4 : koniec odcinka “p” (punkt L) znajduję się bliżej niż punkt ograniczający “stary” obszar ze strony prawej (punkt P)

W wyniku tego podziału obszar znajdujący się ponad prostą “j” zostanie zastąpiony przez poniższe poddrzewo





Przykładowa struktura poszukiwań dla dwóch odcinków



Lokalizacja w jakim obszarze znajduje się punkt

Mając już utworzoną strukturę przeszukiwań w bardzo łatwy sposób można zlokalizować położenie danego punktu w czasie O(logN). Wyszukiwanie jest podobne jak w zwykłym drzewie :

- Jeśli aktualnie porównywany węzeł jest punktem to porównujemy współrzędne “x”, jeśli analizowany punkt ma większą współrzędną x-ową niż węzeł to idziemy w prawo, w przeciwnym razie poruszamy się w lewo.
- Jeśli aktualnie porównywany węzeł jest odcinkiem, to wtedy sprawdzamy czy szukany punkt znajduję się pod odcinkiem, czy nad odcinkiem : jeśli znajduję się pod odcinkiem to idziemy w prawo, w przeciwnym idziemy w lewo.
- Jeśli węzeł jest obszarem to kończymy poszukiwania - jest to poszukiwany obszar.

Złożoność obliczeniowa

- Budowa struktur do wyszukiwania punktów - O(NlogN)
- Lokalizacja punktu - O(logN)

Źródła

“Computational Geometry” - Mark de Berg, Otfried Cheong, Marc van Kreveld, Mark Overmars