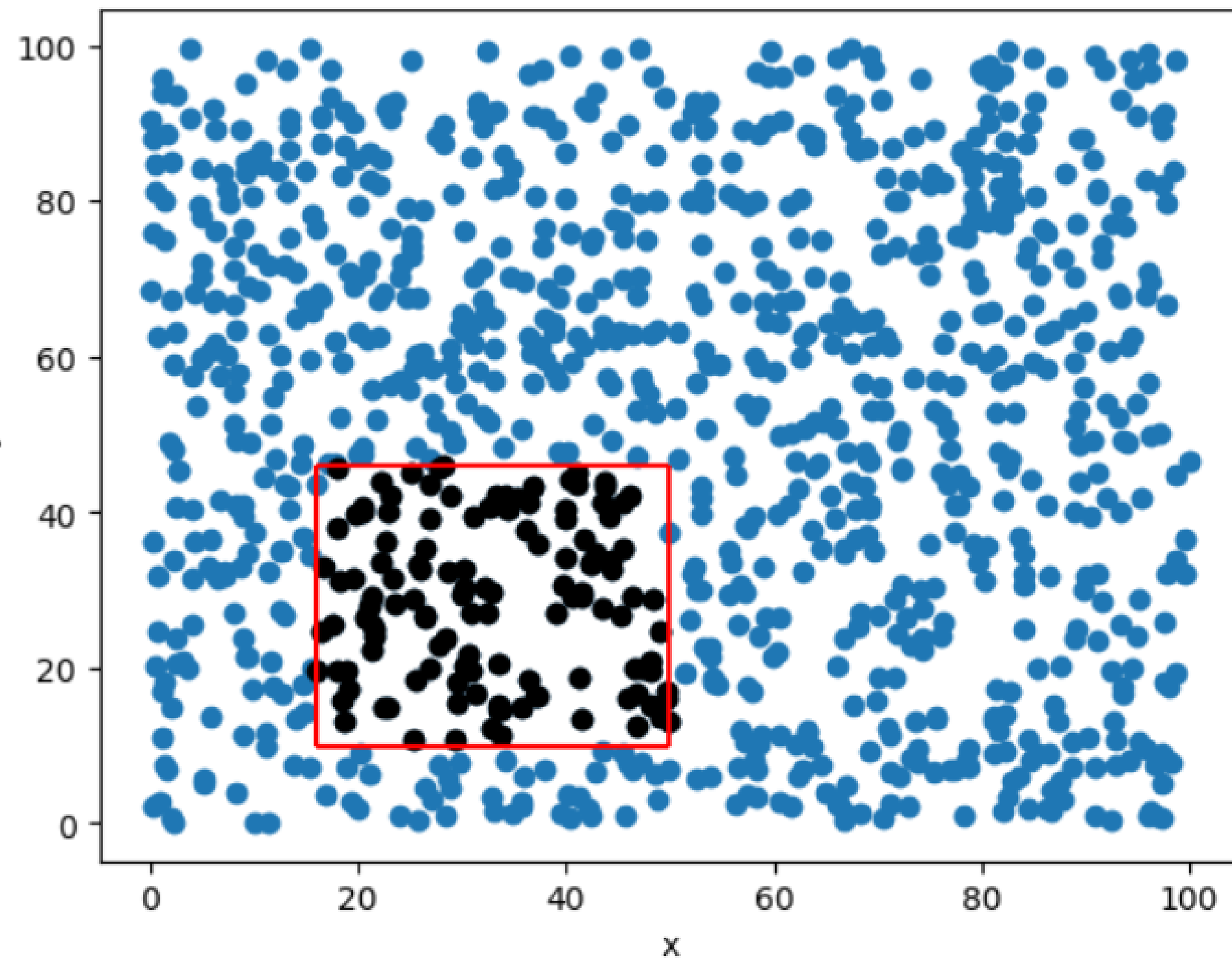


Wyszukiwanie geometryczne – przeszukiwanie obszarów ortogonalnych

QuadTree, oraz Kd-drzewa

Radosław Rolka, Weronika Wojtas



Opis problemu

Dane są punkty P1, oraz P2 na których możemy oprzeć prostokąt. Jak znaleźć punkty Q ze zbioru P, których obie współrzędne są większe niż współrzędne punktu P1, oraz mniejsze niż współrzędne punktu P2?

Celem projektu jest zaimplementowanie struktur QuadTree, oraz KdTree, które pozwalają w szybkim czasie odpowiadać na to pytanie.

Analiza podstawowego algorytmu

Możemy odpowiedzieć na to pytanie przechodząc po całym zbiorze punktów sprawdzając warunek dla każdego z nich.

Porównanie algorytmów



Algorytm podstawowy

- Złożoność czasowa $O(n)$



QuadTree

- Złożoność czasowa $O(\log n + k)$



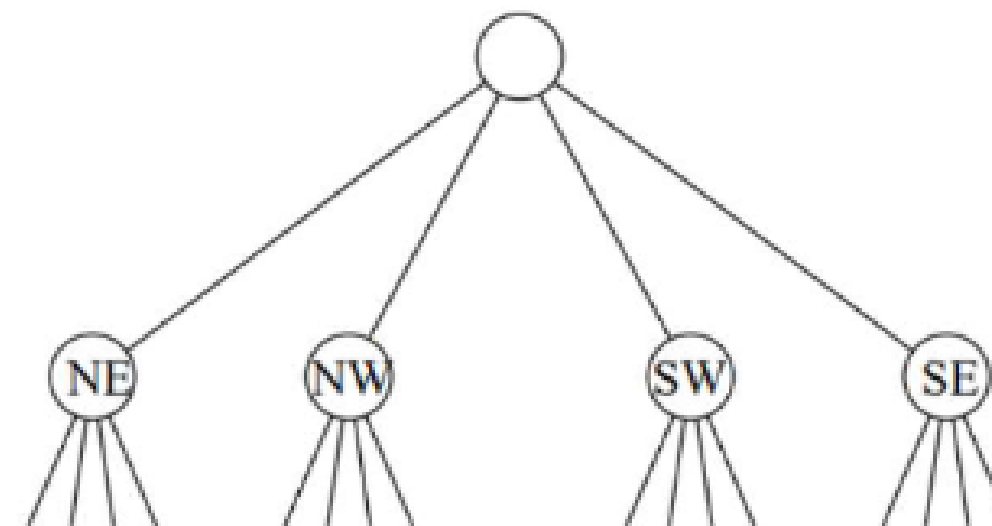
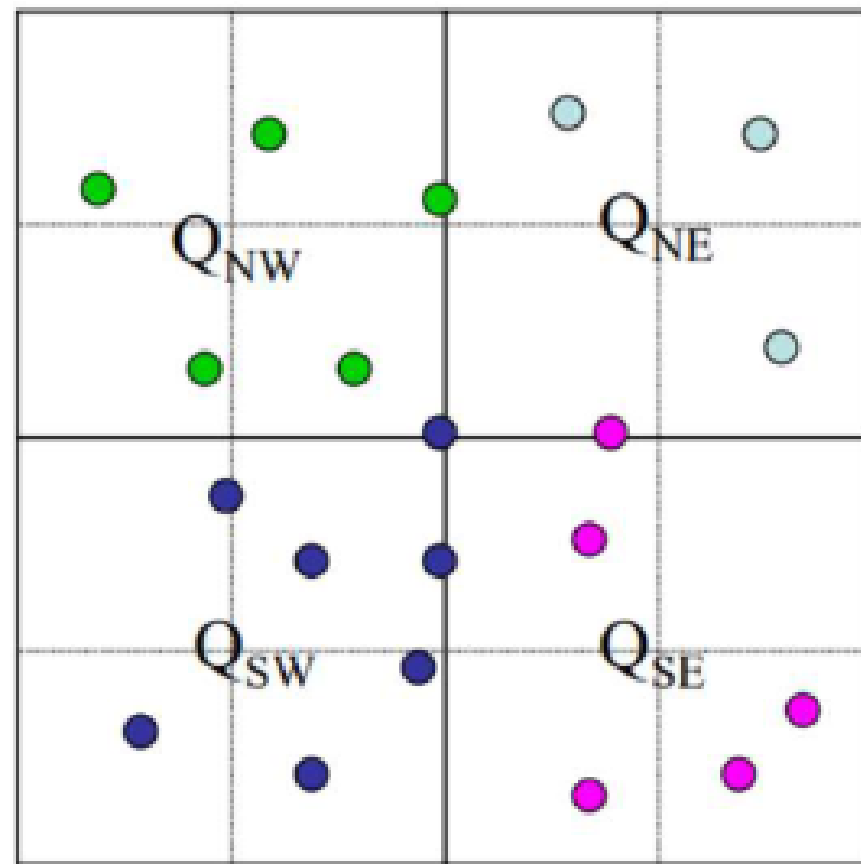
KdTree

- Złożoność czasowa $O(\log n + k)$

k - liczba punktów wynikowych

QuadTree - drzewo ćwiartkowe

QuadTree jest strukturą drzewiastą służącą do przeszukiwania przestrzeni dwuwymiarowej. Proces "budowania" drzewa polega na dzieleniu przestrzeni na ćwiartki tak długo, aż w pojedynczej ćwiartce nie znajdzie się oczekiwana lub mniejsza ilość punktów.



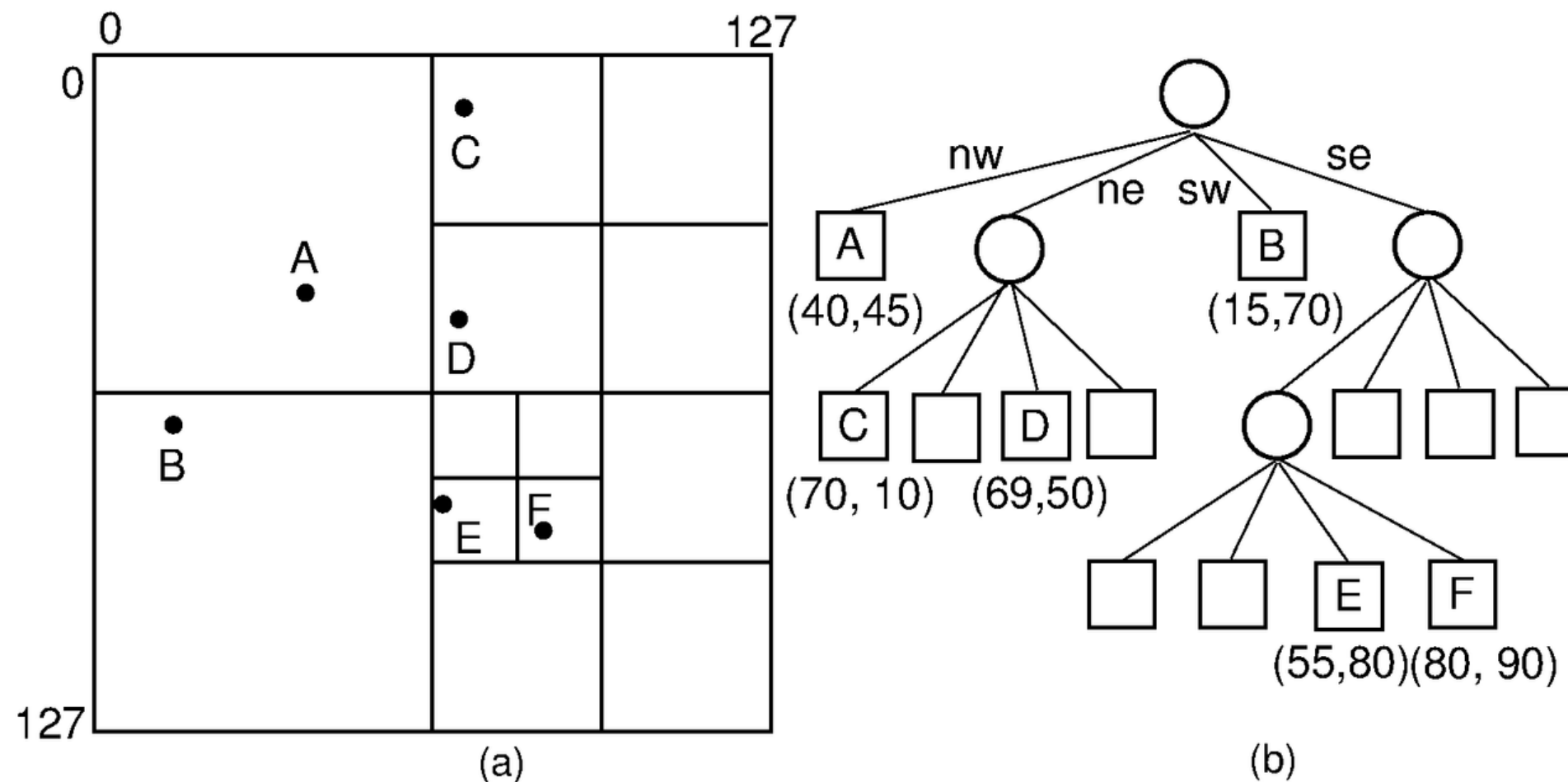
QuadTree - budowa drzewa



Obliczenie wielokątu opartego na zbiorze punktów - ustawiany jest on jako korzeń drzewa



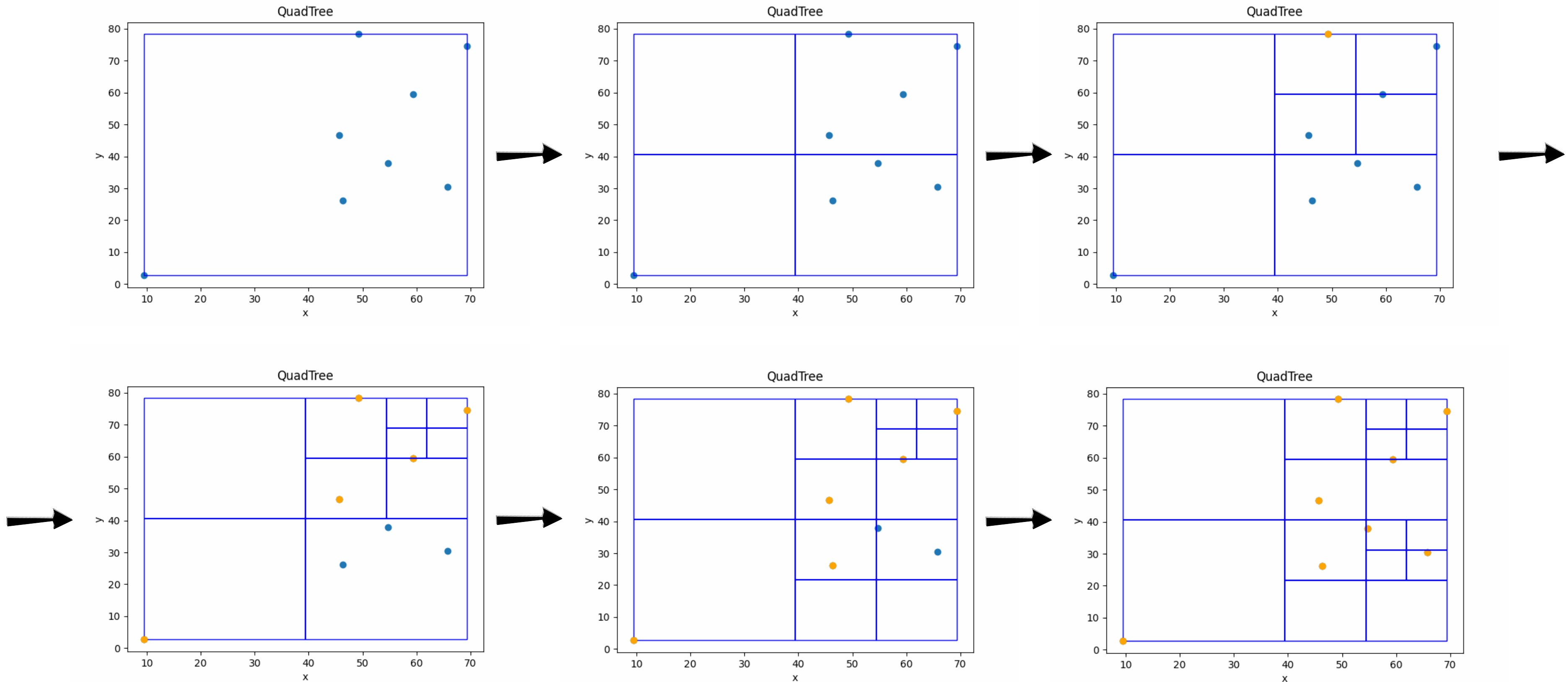
Rekurencyjny podział wielokątu na ćwiartki, do momentu aż w każdej z ćwiartek będzie oczekiwana (zadana w konstruktorze) lub mniejsza liczba punktów



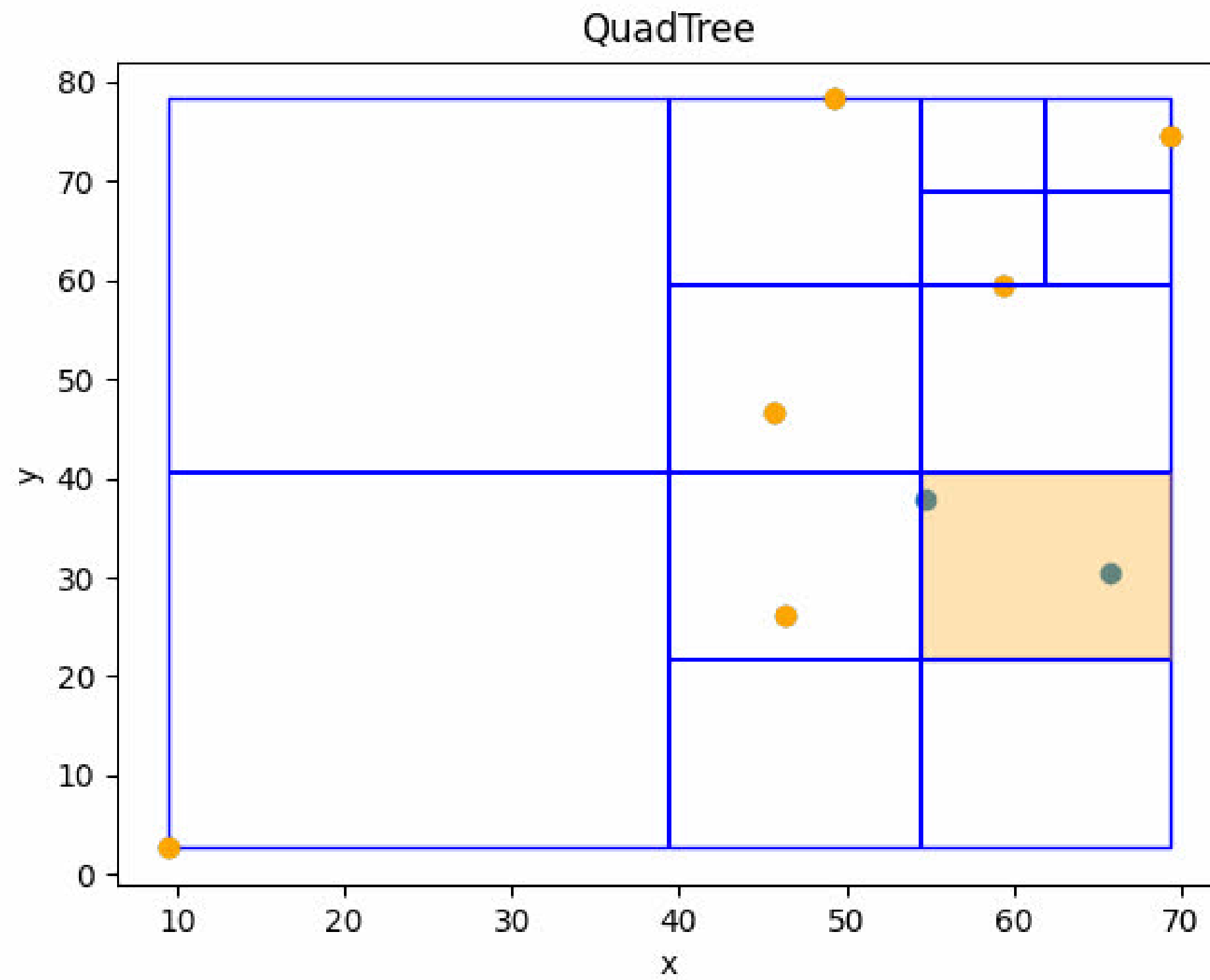
Przykład konstrukcji dla zadanej liczby 1 punktu w liście

QdTree - Wizualizacja budowy

Żółte kropki na wykresach odpowiadają wartościom sklasyfikowanym jako liście



QdTree - Wizualizacja budowy

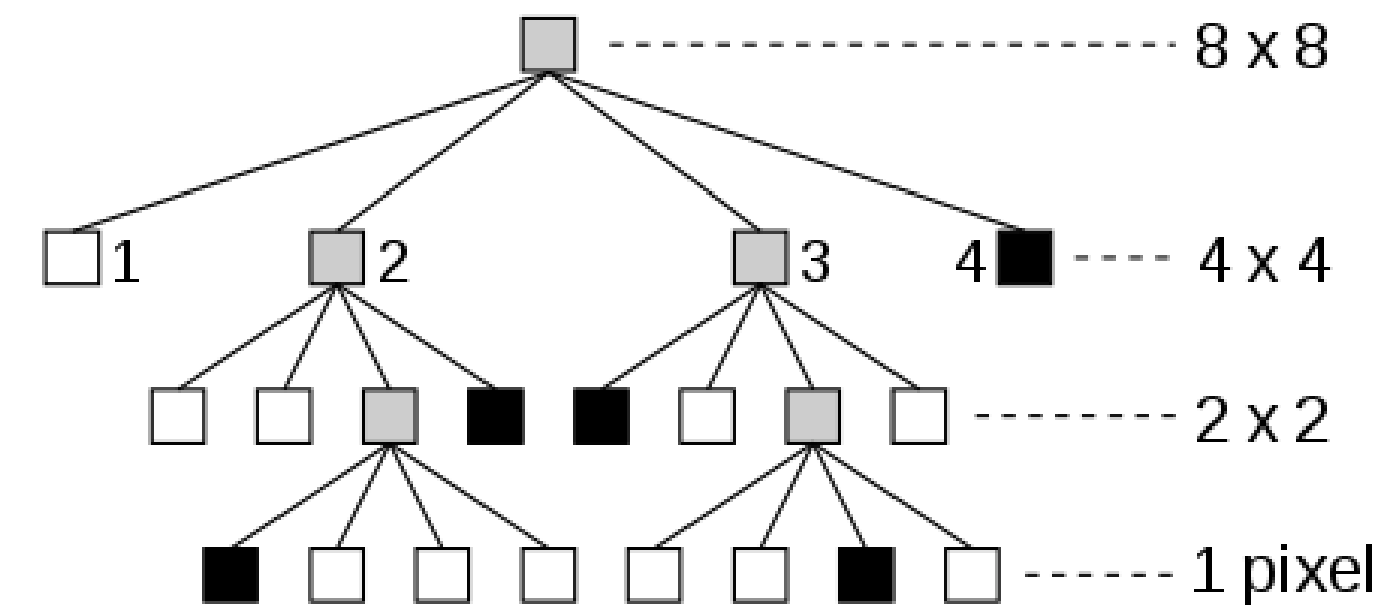
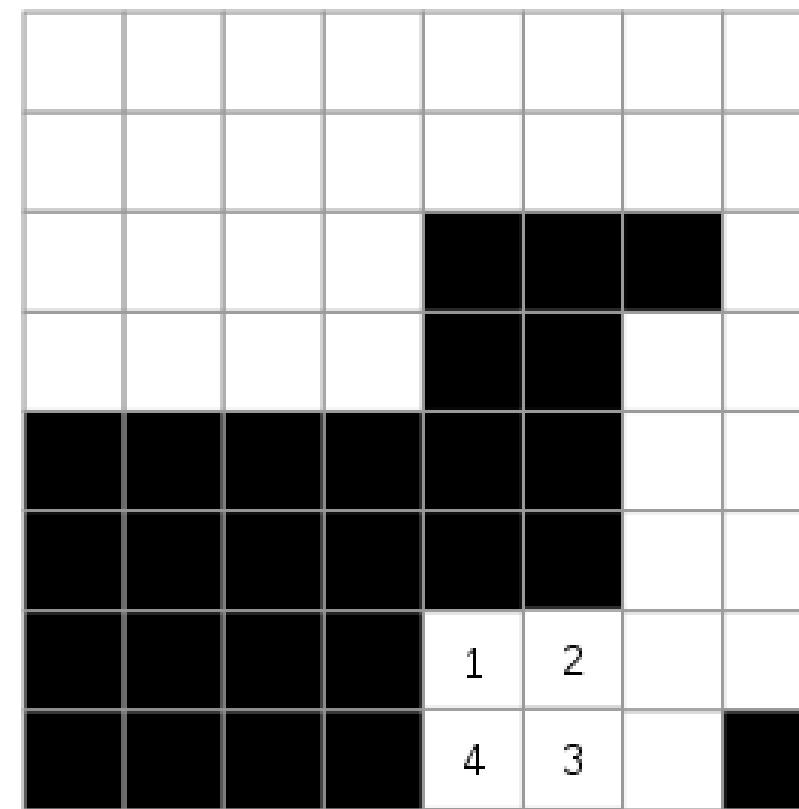


QuadTree - zastosowania

QuadTree jest strukturą używaną głównie w informatyce graficznej, przetwarzaniu obrazu, analizie przestrzennej oraz w problemach związanych z organizacją danych przestrzennych.

Przykłady zastosowania struktury

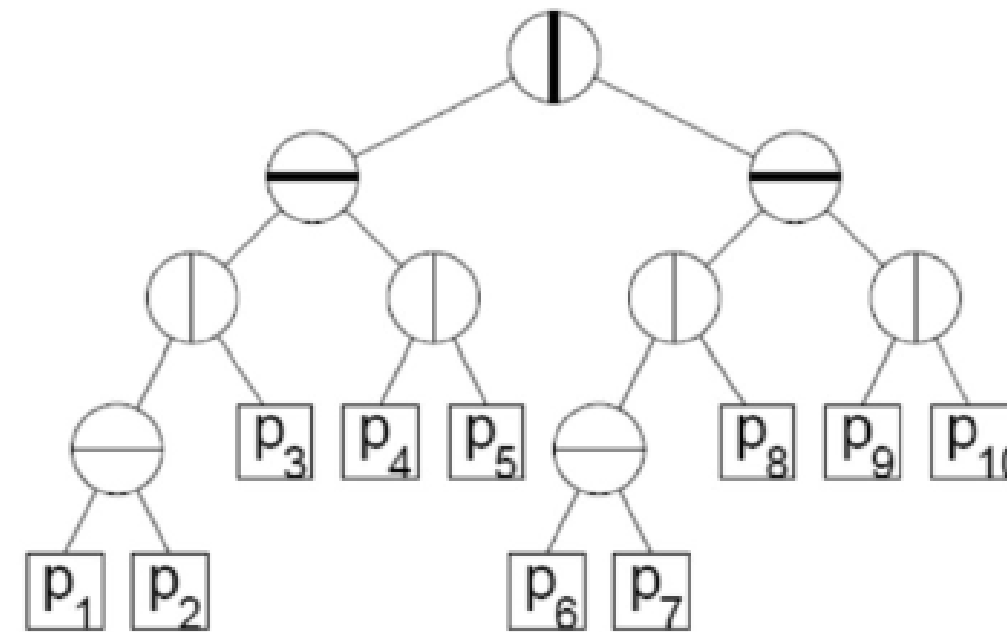
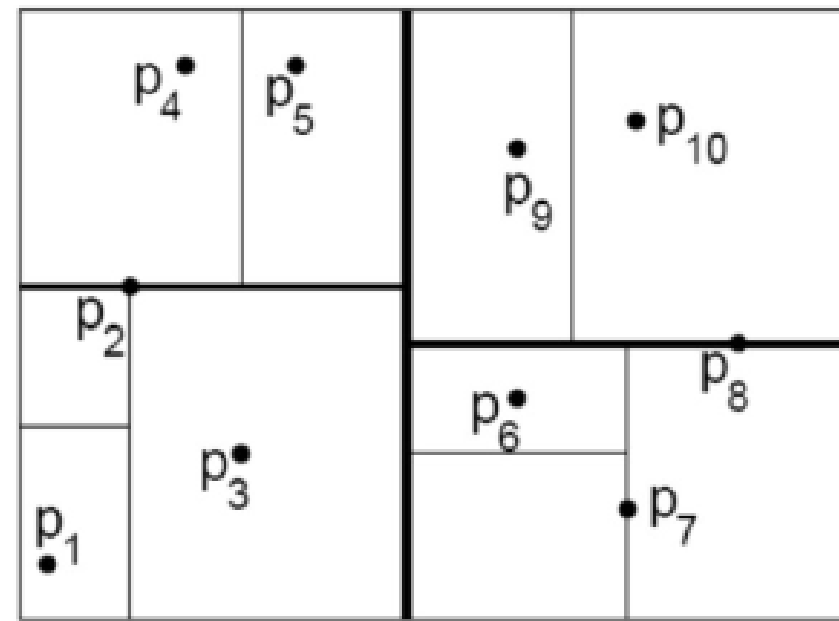
- Kompresja obrazu
- Segmentacja obrazu
- Renderowanie terenu
- Klipping (proces eliminacji obiektów które nie mieszczą się w określonym obszarze widoczności)
- Wyszukiwanie punktów
- Przyspieszanie zapytań przestrzennych
- Zarządzanie kolizjami
- Klastrowanie danych



Wykorzystanie QuadTree do kompresji mapy bitowej

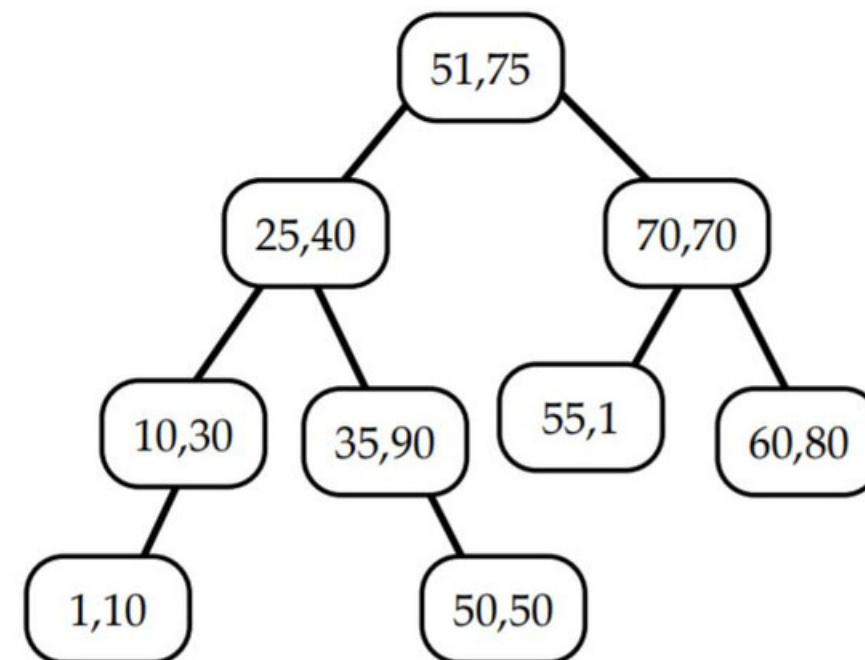
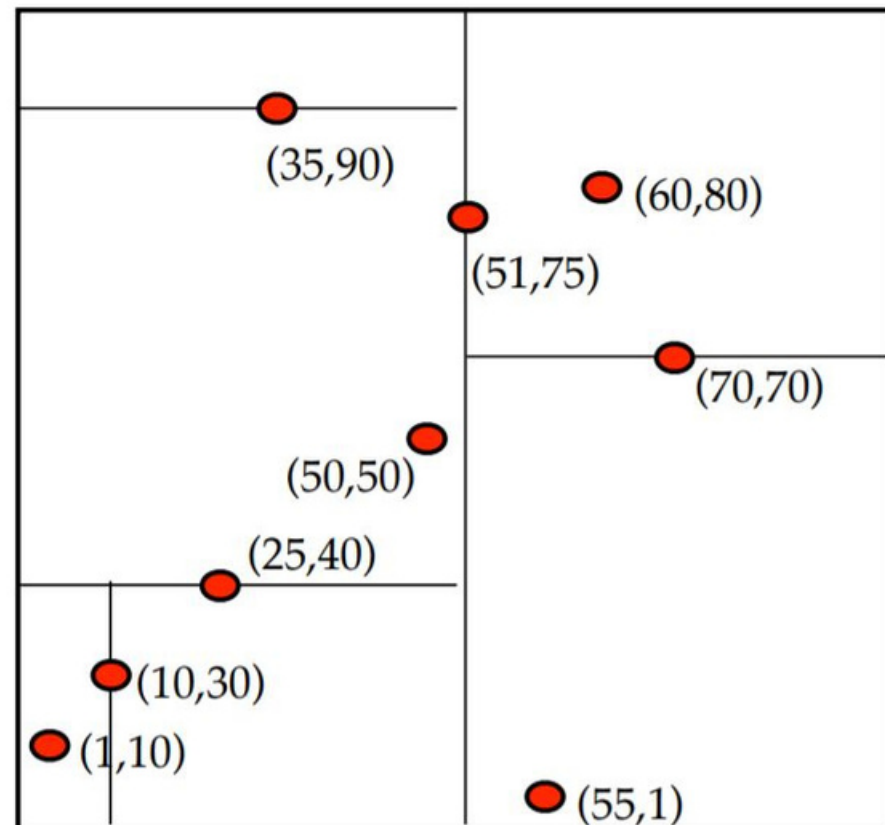
KdTree - drzewo k-wymiarowe

KdTree jest strukturą opartą na idei drzewa binarnego, służy do organizacji i przechowywania punktów przestrzeni k-wymiarowej. KdTree dzieli swój zbiór na dwie prawie równe części zapewniając lepszą wydajność w pesymistycznych przypadkach niż QuadTree.



W każdym poziomie drzewa dochodzi do podziału według jednej ze współrzędnych, co zapewnia łatwe skalowanie drzewa na wyższe wymiary.

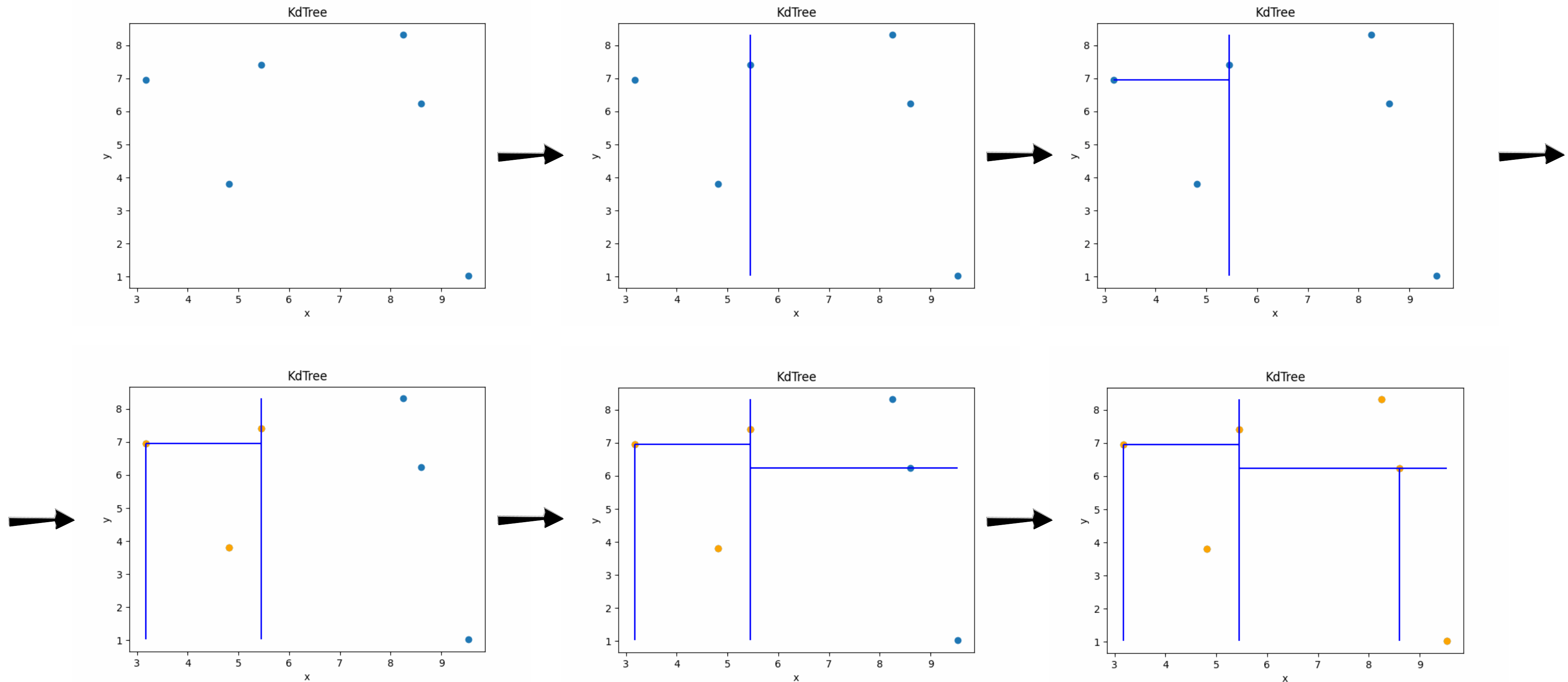
KdTree - budowa drzewa



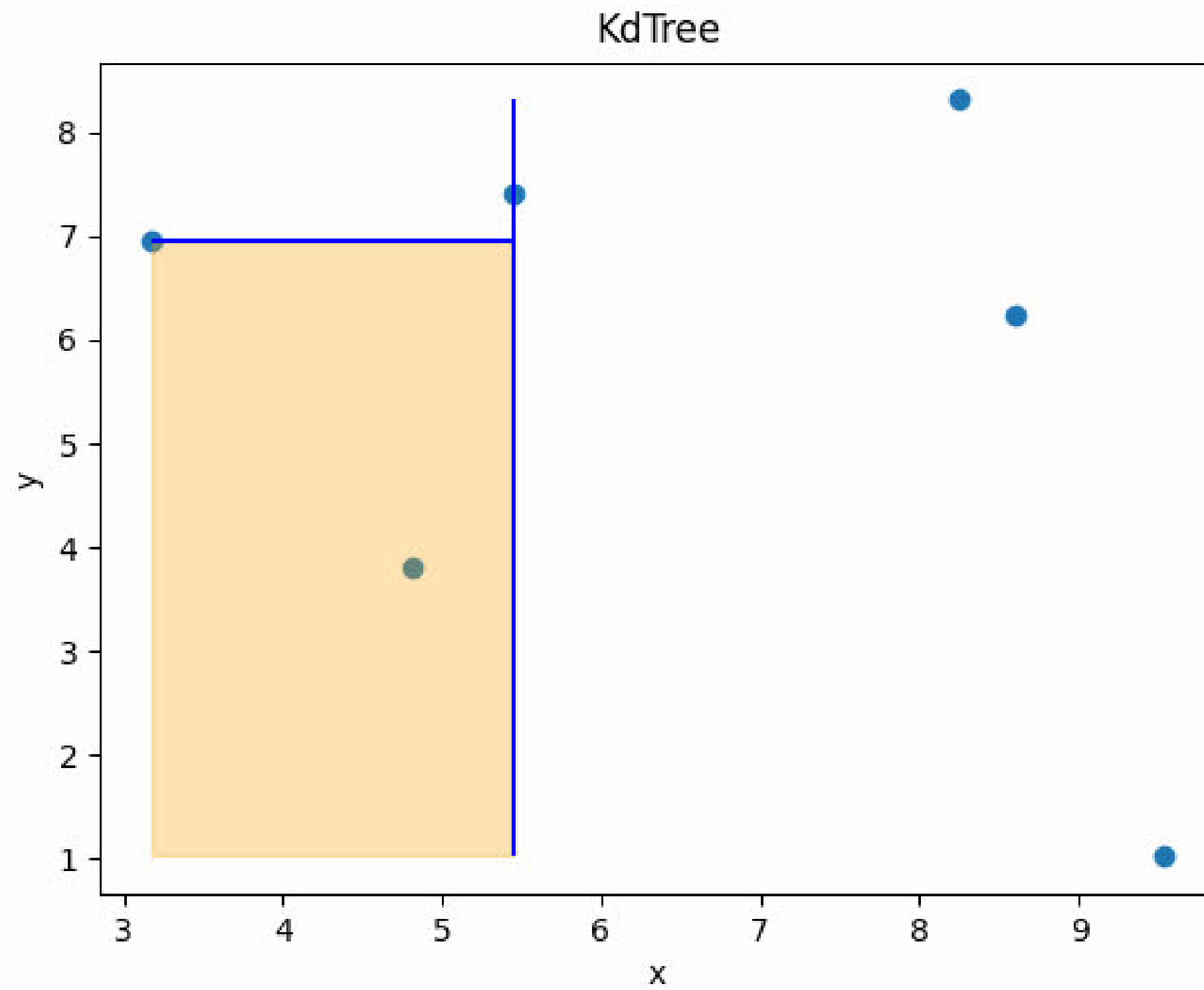
- ✎ Ustawienie prostokątu opartego na zbiorze punktów jako korzenia
- ✎ Wykonywanie funkcji rekurencyjnej do momentu podziału wszystkich punktów na pojedyncze liście
- ✎ Dla każdego węzła nie będącego liściem dokonywane jest sortowanie punktów w zależności od współrzędnej występującej na poziomie węzła w drzewie. Ustawiany jest środek podziału (median) jako $(\text{liczba punktów} - 1) / 2$.
- ✎ Stawiana jest linia podziału przechodząca przez punkt z indeksem median. kierunek linii zależy od współrzędnej poziomu węzła w drzewie. Tworzone są dzieci węzła posiadające powstałe zbiory punktów

KdTree - Wizualizacja budowy

Żółte kropki na wykresach odpowiadają wartościom sklasyfikowanych jako liście



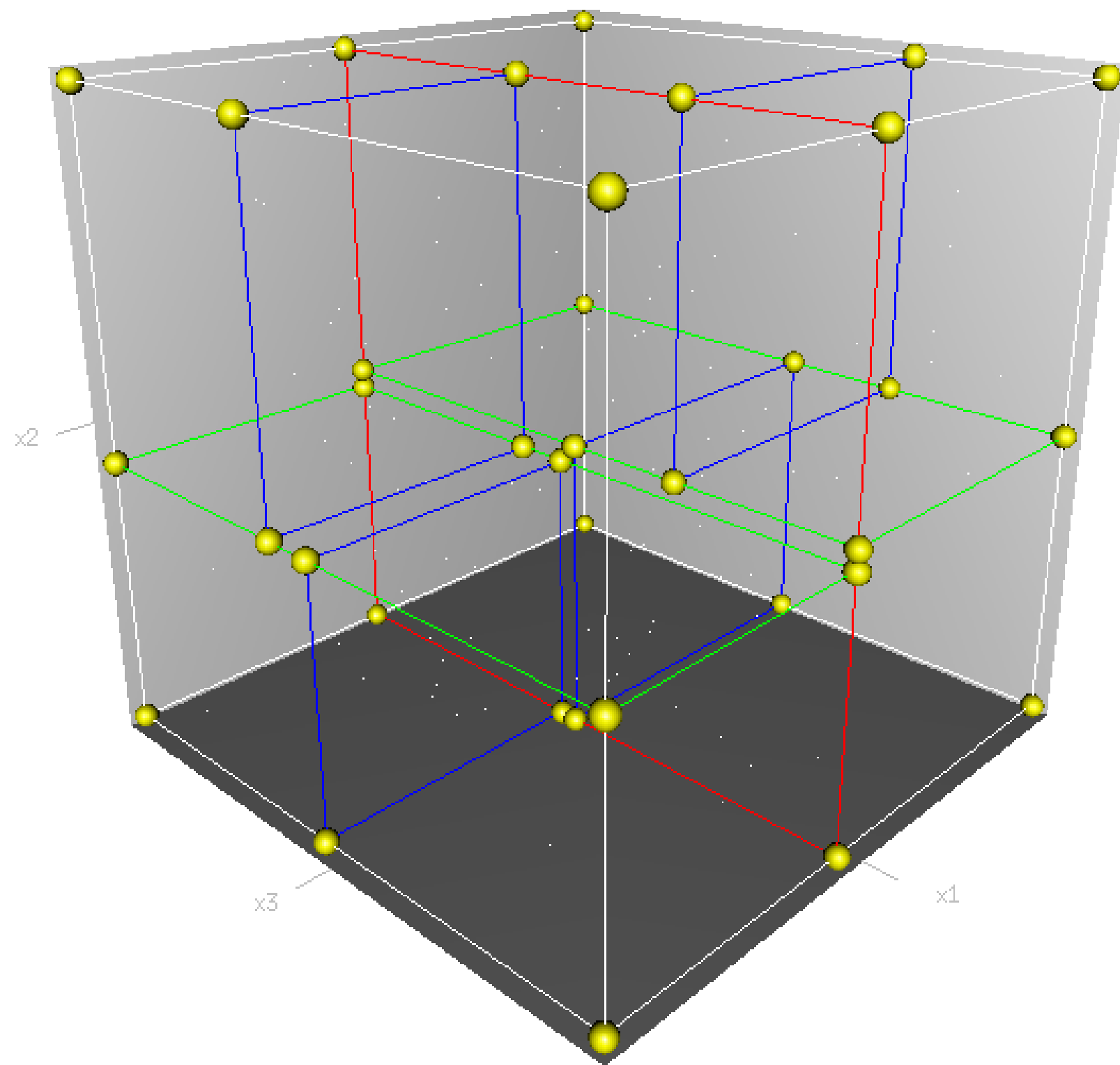
KdTree - Wizualizacja budowy



KdTree - zastosowania

Kd-drzewa są stosowane w różnych dziedzinach, gdzie efektywne przeszukiwanie przestrzeni wielowymiarowej jest kluczowe.

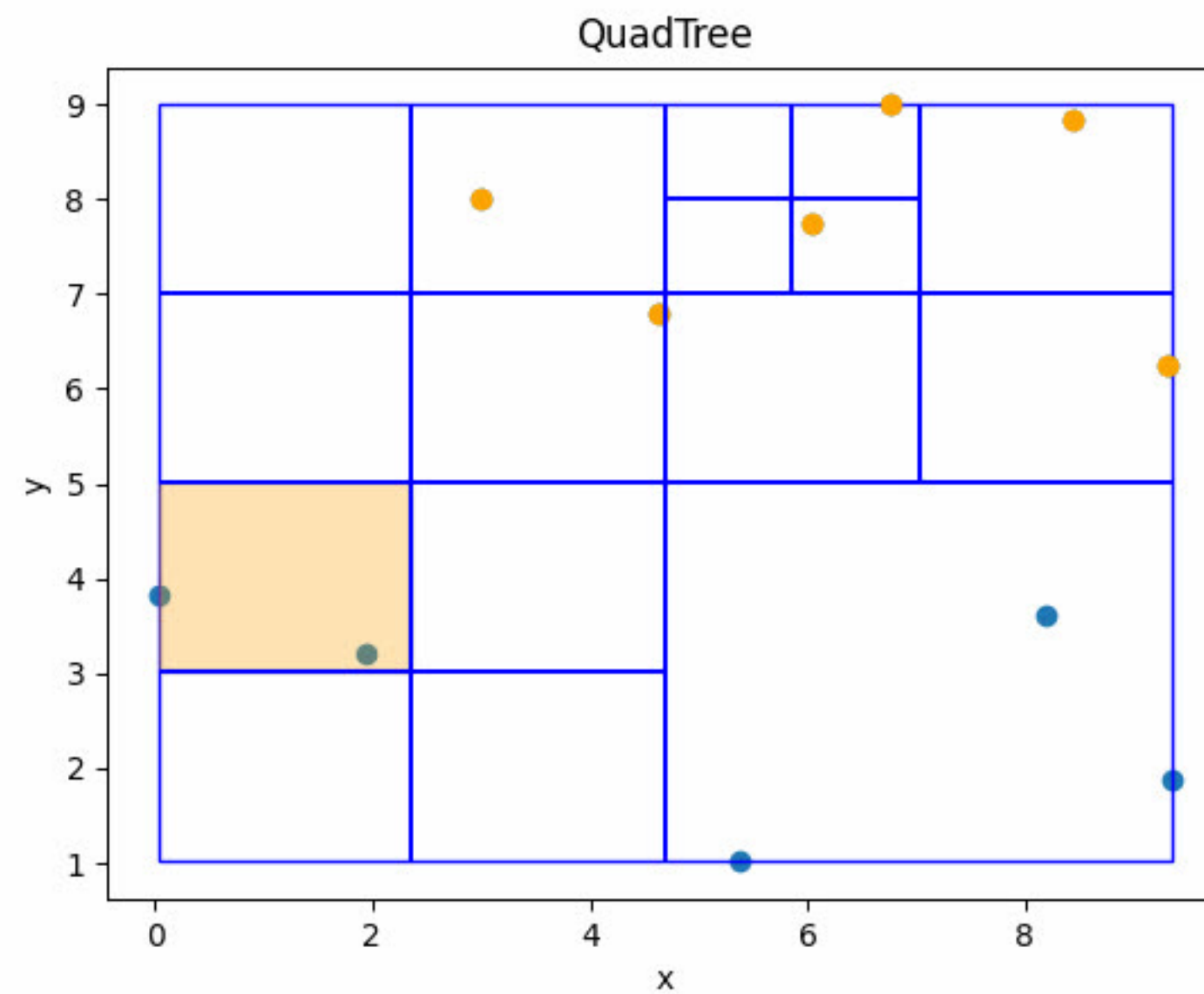
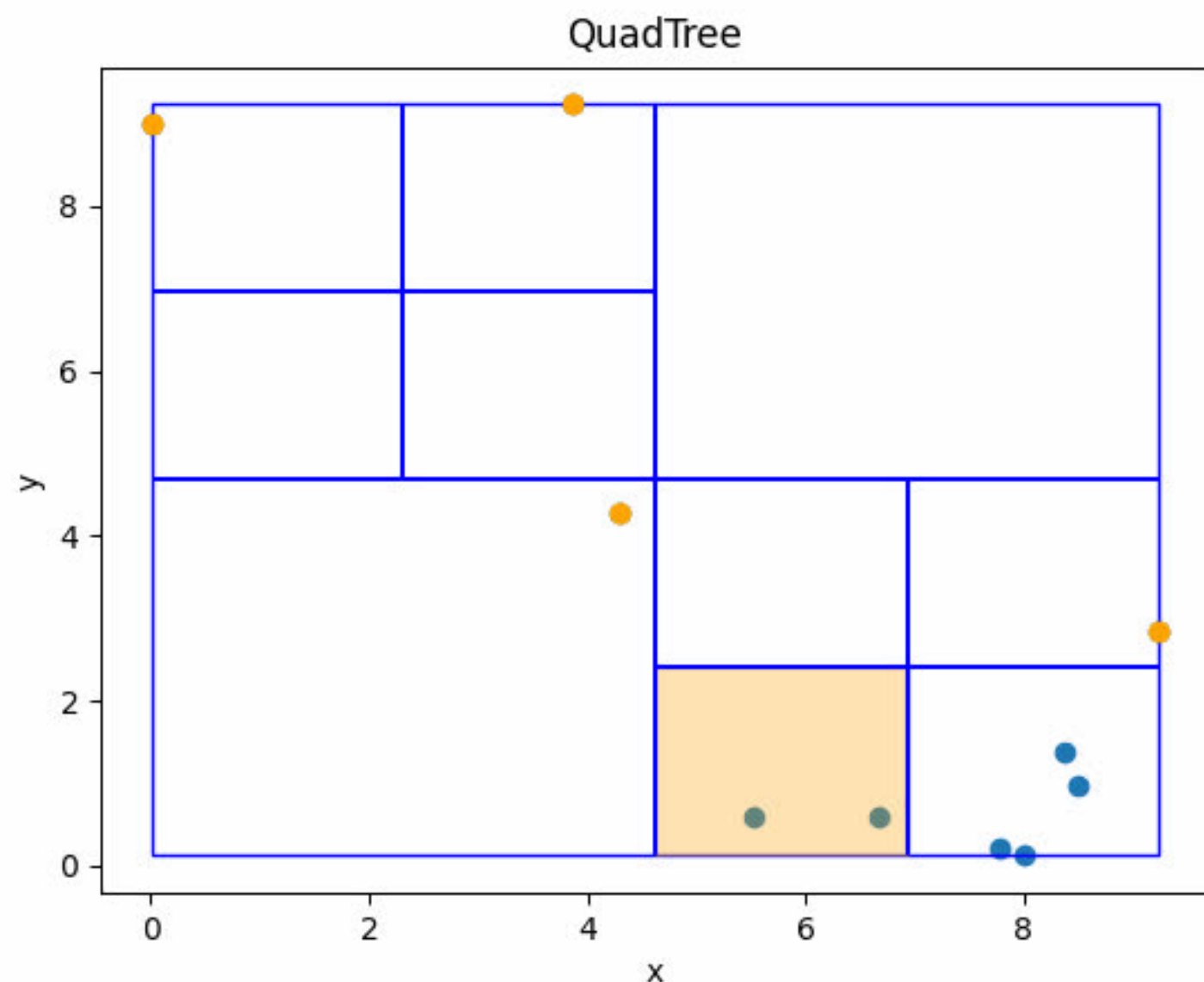
- Bazy danych - wyszukiwanie przedziałów (zakresów)
- Grafika komputerowa - renderowanie scen 3D
- Analiza danych przestrzennych - wyszukiwanie najbliższych sąsiadów
- Robotyka i nawigacja - planowanie tras (zapobieganie kolizjom)
- Analiza obrazów - wyszukiwanie podobieństw



Przykład podziału przestrzeni za pomocą KdTree

QdTree - wyszukiwanie punktu w drzewie

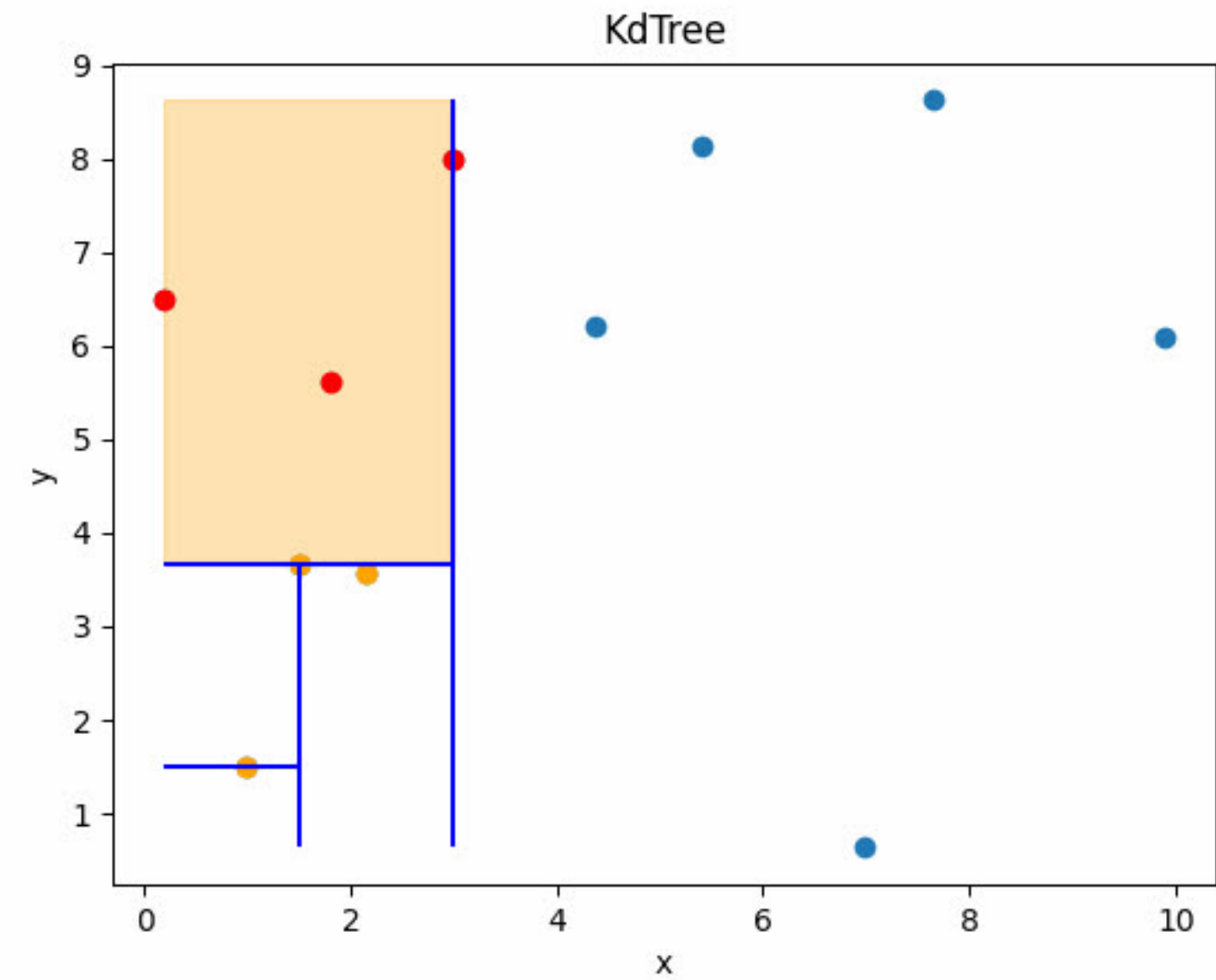
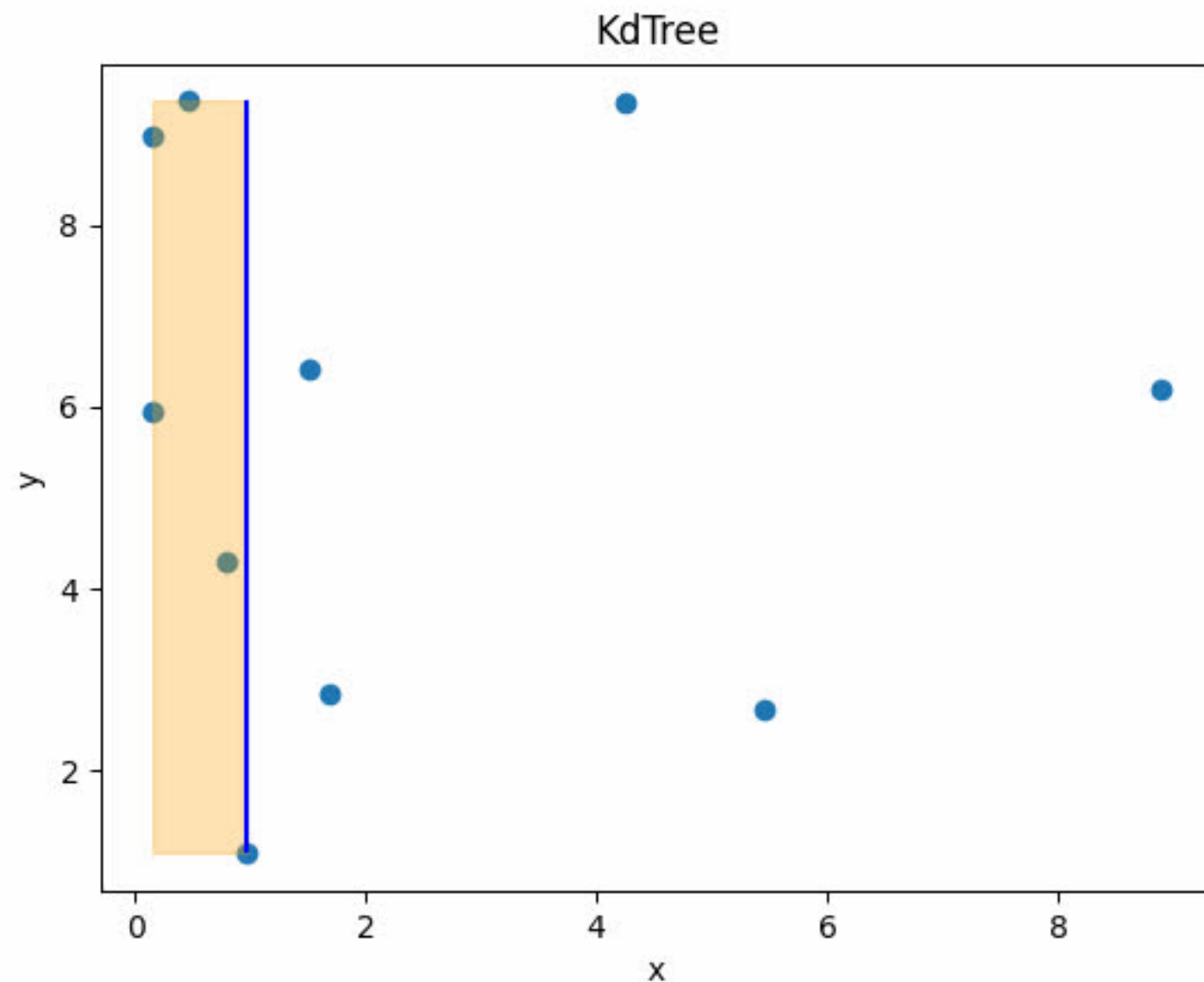
Rekurencyjnie sprawdzamy węzły zaczynając od korzenia. Jeśli węzeł jest liściem przeszukujemy zbiór jego punktów. W przeciwnym przypadku przeszukujemy te dzieci węzła które mogą zawierać szukany punkt.



Pomarańczowy obszar na wykresie to ćwiartki które algorytm przeszukiwał. Punkt czarny oznacza brak punktu w drzewie (lewy obrazek), punkt zielony oznacza znalezienie szukanego punktu (prawy obrazek)

KdTree - wyszukiwanie punktu w drzewie

Algorytm analogiczny do QuadTree. Różnią się tylko ilością dzieci. W przypadku KdTree przeszukiwana jest dwójka potomków zamiast czwórki.



Pomarańczowy obszar na wykresie to ćwiartki które algorytm przeszukiwał. Punkt czarny oznacza brak punktu w drzewie (lewy obrazek), punkt zielony oznacza znalezienie szukanego punktu (prawy obrazek)

QuadTree, KdTree - wyszukiwanie zbioru punktów

Zasada działania algorytmu jest jednakowa dla obu struktur. Przeszukiwanie drzewa zaczynamy od korzenia. Sprawdzamy czy szukany obszar przecina się z obszarem zbioru punktów (korzeniem). Wykonujemy funkcję rekurencyjną, której kroki możemy podzielić na kilka typów węzłów.



WĘZEŁ LIŚĆ

Dodajemy wszystkie punkty z liścia zawarte w szukanym prostokącie do wyniku



WĘZEŁ CZĘŚCIOWO ZAWARTY W SZUKANYM PROSTOKĄCIE

Wywołujemy rekurencyjne wyszukiwanie wszystkich jego dzieci



WĘZEŁ W CAŁOŚCI ZAWARTY W SZUKANYM PROSTOKĄCIE

Dodajemy wszystkie punkty z węzła do wyniku

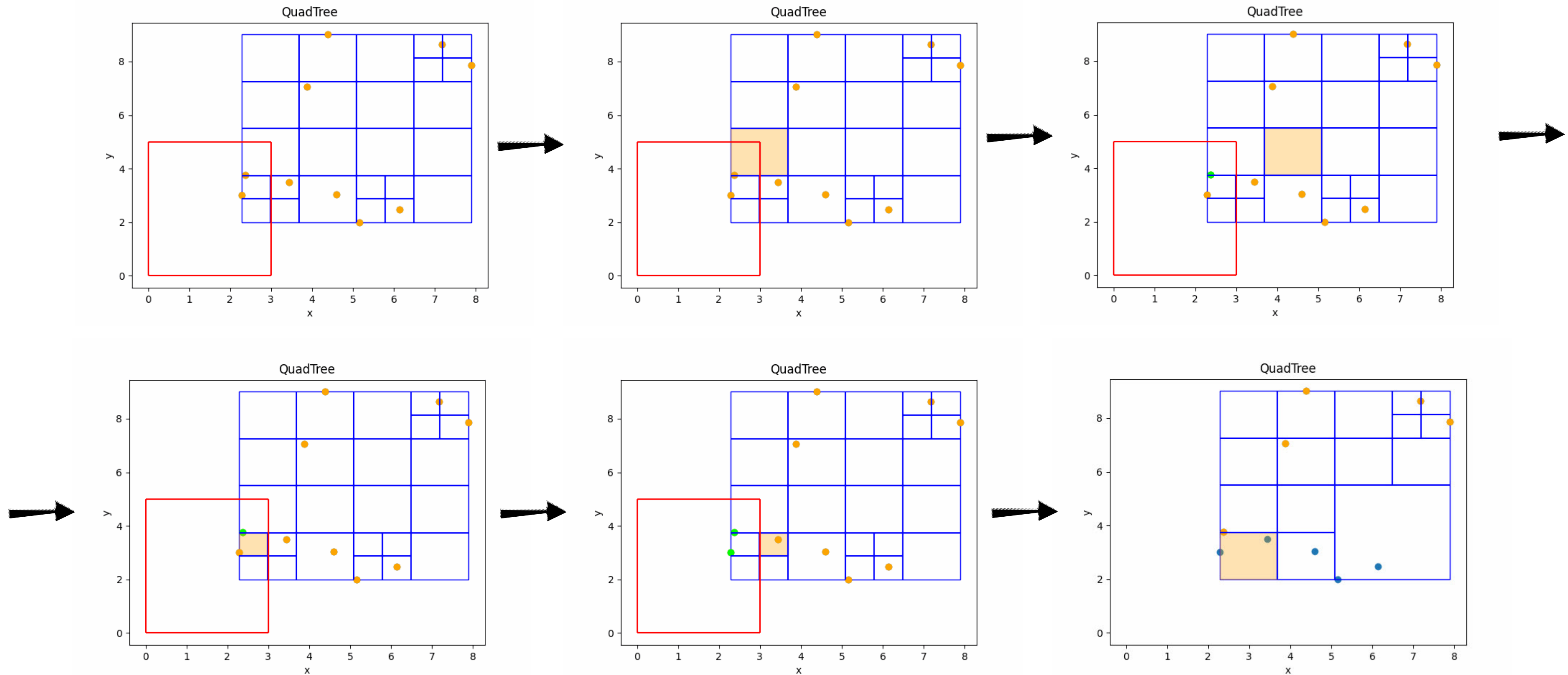


WĘZEŁ NIE POSIADAJĄCY PRZECIĘĆ Z SZUKANYM OBSZAREM

Kończymy wyszukiwanie w tym węźle

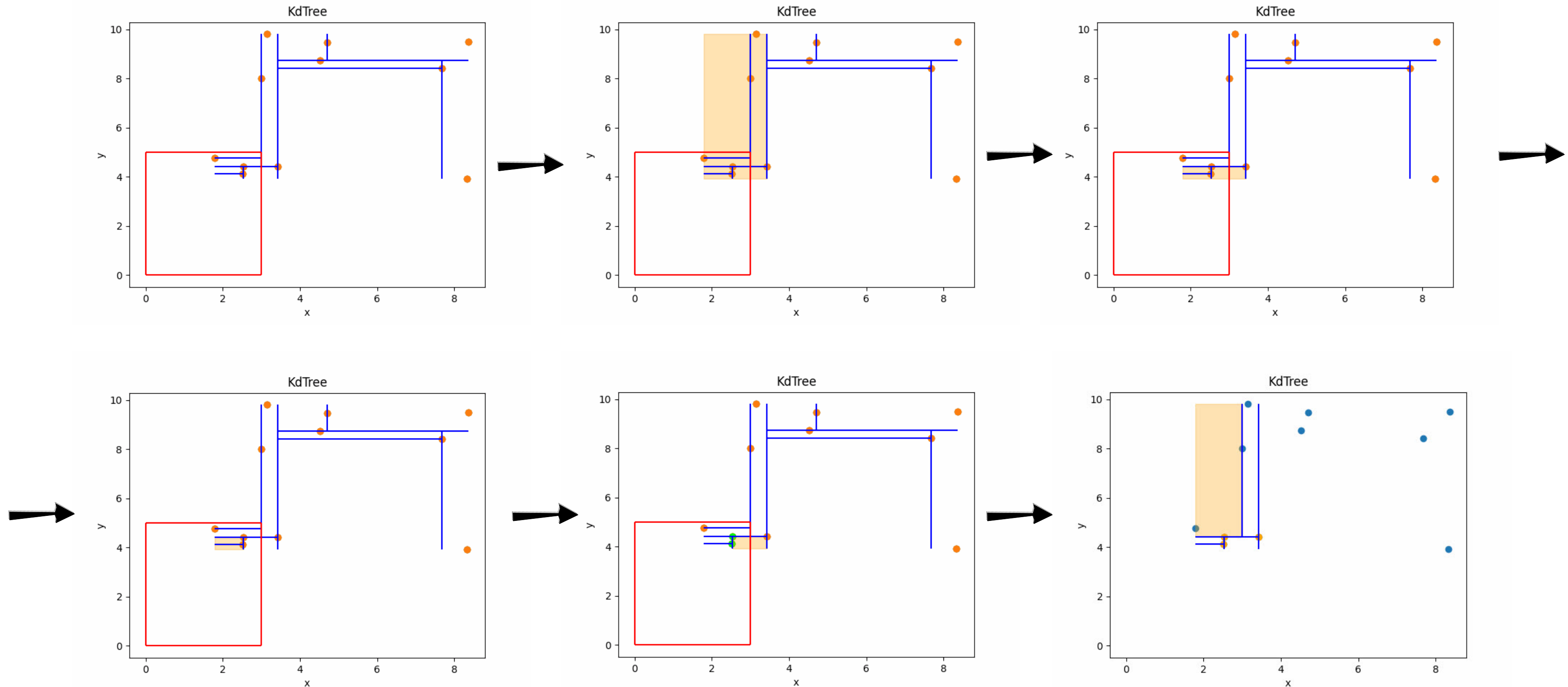
QdTree - Wizualizacja wyszukiwania

Zielone kropki na wykresie odpowiadają znalezionym wartościom, żółty obszar jest aktualnie przeszukiwanym obszarem. Czerwony obszar jest prostokątem oczekiwanym. Pokazanych jest pierwsze kilka kroków algorytmu.



KdTree - Wizualizacja wyszukiwania

Zielone kropki na wykresie odpowiadają znalezionej wartości, żółty obszar jest aktualnie przeszukiwanym obszarem. Czerwony obszar jest prostokątem oczekiwanym. Pokazanych jest pierwsze kilka kroków algorytmu



Wady i zalety struktur



QUADTREE

- Skuteczne w przypadku klastrów danych
- Prosta implementacja w dwóch wymiarach
- Proste zwiększanie liczby punktów w liściu
- Efektywne w statycznych lub mało dynamicznych strukturach
- Mało wydajne w przypadku równomiernie rozłożonych danych
- Mało wydajne w przypadku dynamicznych danych



KDTREE

- Efektywne dla równomiernie rozłożonych danych
- Łatwo skalowalne na dane o większej wymiarowości
- Wydajne dla dynamicznych struktur
- Mniejsza czytelność (trudniejsze w implementacji)

Obie struktury zapewniają szybkie przeszukiwanie przestrzeni, jednak są wrażliwe na strukturę danych

Dziękujemy za uwagę

Radosław Rolka, Weronika Wojtas

Źródła

- <https://github.com/aghbit/Algorytmy-Geometryczne>
- <https://en.wikipedia.org/wiki/Quadtree>
- https://en.wikipedia.org/wiki/K-d_tree
- https://en.wikipedia.org/wiki/Nearest_neighbor_search
- https://en.wikipedia.org/wiki/Range_searching
- <https://www.agh.edu.pl/o-agh/multimedia/znak-graficzny-agh/>
- https://github.com/Goader/KDTree_QuadTree/tree/main
- Grafiki google