Dokumentacja

Klasy użyte na potrzeby realizacji algorytmu:

- `Point` - reprezentuje punkt w przestrzeni za pomocą współrzędnych `x` i `y`
- `Segment` - reprezentuje odcinek w przestrzeni za pomocą `start` i `end` typu `Point`, gdzie zawsze `start.x <= end.x`
- `Trapezoid` - reprezentuje trapez w sposób jaki jest potrzebny do realizacji algorytmu:
  - `leftp` = Punkt lewej krawędzi
  - `rightp` = Punkt prawej krawędzi (punkt)
  - `top` = Górna krawędź (odcinek)
  - `bottom` = Dolna krawędź (odcinek)
  - `left_top` = Wskaźnik do lewego, górnego sąsiada (jeśli brak sąsiada to `None`)
  - `left_bottom` = Wskaźnik do lewego, dolnego sąsiada (jeśli brak sąsiada to `None`)
  - `right_bottom` = Wskaźnik do prawego, dolnego sąsiada (jeśli brak sąsiada to `None`)
  - `right_top` = Wskaźnik do prawego, górnego sąsiada (jeśli brak sąsiada to `None`)
  - `node` = Wskaźnik do odpowiadającego węzła w grafie
- `Node` - Węzeł w grafie acyklicznym skierowanym. Cechuje go:
  - typ (ściana `WALL` | x-węzeł `X_NODE` | y-wezęł `Y_NODE`),
  - value (odpowiednio: trapez, punkt, odcinek)
  - wskaźniki do lewego i prawego dziecka oraz do rodzica
- `Graph` - acykliczny graf skierowany służący jako graf wyszukiwania. Jego oczekiwany rozmiar jest rzędu O(n). Posiada metody:

- `find(point_to_find)` Zwraca trapez, który zawiera dany punkt lub -1 jeśli procedura nie powiodła się. Złożoność czasowa tej metody O(log n).
- `add_trapezoid(trapezoid, new_node)` - zamienia w grafie węzeł reprezentujący `trapezoid` na nową strukturę `new_node. Złożoność czasowa tej operacji to O(1).

Funkcje realizujące algorytm:

Nazwa funkcji	Krótki opis	Argumenty	Zwracana wartość	Złożoność czasowa
`create_trapezoid_map_and_search_graph`	Realizuje cały algorytm. Tworzy mapę trapezową, strukturę wyszukiwań i ją oraz tworzy dane do wizualizacji.	(segments, MAX_Y, MIN_Y, MAX_X, MIN_X)	D, scenes	O(n*logn)
`new_segment_in_map`	Przetwarza nowy odcinek i aktualizująca wszystkie struktury (mapę trapezową oraz graf wyszukiwań).	(found_trapezoid, new_segment, visual_representation, D)	None	O(k), k- ilość zakt. trapezów
`prepare_data`	Przygotowuje dane na potrzeby algorytmu.	(segments //as lists)	[Segment], MAX_Y, MIN_Y, MAX_X, MIN_X	O(n)
`draw_collection`	Zwraca dane łatwe do wizualizacji.	(elements //as objects)	elements_as_lists	O(n)
`is_above_segment`	Odpowiada na pytanie czy dany odcinek leży nad (po lewej stronie) odcinka.	(point_to_check, segment)	`True` or `False`	O(1)

Dodatkowo funkcje `first_segment_insertion` , `update_first_element` , `update_last_element` , `update_middle_element` są używane do rozpatrują konkretne przypadki przecinania trapezu przez nowy odcinek i aktualizują mapę trapezową. Ich złożoność to O(1), jednak funkcja

update_middle_element jest rekurencyjnie wywoływana k-2 razy (k to liczba zaktualizowanych trapezów) - stąd złożoność `new_segment_in_map` = O(k). Odpowiadające im funkcje `update_graph_first_segment`, `update_graph_first_element`, `update_graph_last_element`, `update_graph_middle_element` aktualizują graf wyszukiwania zgodnie z wprowadzonymi wcześniej zmianami w mapie trapezowej. Ich złożoność to również O(1).