Zadanie: Pole sumy prostokątów

Najpierw przeczytaj CAŁY niniejszy opis. Potem implementuj. Każdy inny sposób podejścia do tego zadania grozi niepowodzeniem.

Całe zadanie ma na celu przećwiczenie algorytmu z zamiataniem. Będziemy mierzyć się z następującym problemem: Mamy zbiór być może nakładających się prostokątów w dwuwymiarowej przestrzeni Euklidesowej, których boki są równoległe lub prostopadłe do osi układu współrzędnych. Naszym zadaniem jest policzyć pole tych prostokątów, ale w ten sposób, że jeśli jakiś fragment przestrzeni należy do więcej niż jednego prostokąta, to i tak liczymy go tylko raz (czyli liczymy pole sumy teoriomnogościowej tych prostokątów).

Aby rozwiązać to zadanie, najpierw zmierzymy się z zadaniem prostszym, czyli tym samym problemem, ale w jednowymiarowej przestrzeni czyli policzeniem długości być może nakładających się odcinków (dany fragment liczymy tylko raz). Po pierwsze jest to łatwiejsze do wyobrażenia sobie, a poza tym jest niezbędne, aby potem policzyć pola prostokątów.

UWAGA DO TESTÓW: aby każdy nie musiał sobie rozrysowywać przykładów testowych, są one narysowane w plikach pdf. Jeśli masz problem z konkretnym testem, po prostu otwórz plik o odpowiadającej mu nazwie, np. test1.pdf i spróbuj dojść, co jest nie tak w Twoim algorytmie. Dla pierwszego etapu, z odcinkami, w niektórych testach, dla czytelności, odcinki nie leżą na jednej linii, a na wielu równoległych liniach. Oczywiście tak naprawdę one wszystkie leżą na jednej prostej.

Etap 1 – długość odcinków (1p)

Mamy zbiór być może nakładających się odcinków w jednowymiarowej przestrzeni Euklidesowej, które siłą rzeczy znajdują się na jednej prostej. Naszym zadaniem jest policzyć długość tych odcinków, ale w ten sposób, że jeśli jakiś fragment przestrzeni należy do więcej niż jednego odcinka, to i tak liczymy go tylko raz (czyli liczymy pole sumy teoriomnogościowej tych odcinków).

UWAGA TECHNICZNA: będziemy dostawać na wejściu ogólną klasę Segment, który to odcinek może być w dowolnej pozycji w dwuwymiarowej przestrzeni. Jednak mamy zagwarantowane, że wszystkie one będą ustawione na jednej PIONOWEJ linii prostej (czyli x-owa współrzędna jest taka sama, różni się jedynie y-owa). Dlaczego pionowa? Bo nam się przyda w drugim etapie, przy prostokątach.

Wyobraźmy sobie taki zbiór pionowych odcinków. Wyobraźmy sobie poziomą linię, która przemieszcza się od dołu do góry, zatrzymując się na punktach wyznaczających początek lub koniec jakiegoś odcinka. Każdy taki punkt i każde takie zatrzymanie się linii nazywamy źdarzeniem". Tak naprawdę w ogólności w przestrzeni mamy kilka serii nakładających się odcinków (zanim jeden odcinek się skończy, to już się zaczyna inny). Gdy poznamy długość każdej z takich serii, odpowiedzią będzie suma ich długości. Musimy wykryć, gdzie taka pojedyncza seria się zaczyna, a gdzie kończy, czyli mieć jej zakres. Zakres da nam długość.

Innymi słowy, musimy wykryć, w przy którym zdarzeniu weszliśmy w serię odcinków, a w którym zdarzeniu z niej wyszliśmy. Będzie to przypominać sprawdzanie, czy jesteśmy w nawiasie w tekście (z poprawnymi nawiasami). Będziemy zliczać otwierające się nawiasy (czyli punkty, które są początkami odcinków) i zamykające (punkty, które są końcami odcinków). Gdy różnica między tymi dwoma liczbami wyniesie zero, to właśnie zakończyła się seria.

A teraz szczegóły:

1. Stwórz strukturę, która ma indeks odcinka i informację, czy reprezentuje jego początek czy koniec. Struktura reprezentuje jeden z punktów krańcowych odcinka.
2. Stwórz listę tych struktur dla odcinków wejściowych.
3. Posortuj tę listę po y-owej współrzędnej.
4. Teraz zaimplementuj przechodzenie linii zamiatającej po zdarzeniach od dołu do góry. Pamiętaj, ile odcinków się zaczęło, a ile zakończyło.
5. Gdy zakończyła się seria nakładających się odcinków (czyli gdy liczba odcinków rozpoczętych i zakończonych jest taka sama) policz długość tej serii i dodaj do sumy ogólnej.
6. Wynikiem jest suma długości uzyskanych w poprzednim punkcie.

Etap 2 – pole prostokątów (1.5p)

Teraz, gdy mamy algorytm na długość odcinków, możemy przystąpić do prostokątów.

Wyobraź sobie prostokąty. Teraz wyobraź sobie pionową linię zamiatającą. Idzie ona od lewej do prawej. Tym razem zatrzymuje się ona za każdym razem, gdy natrafi na pionowy bok prostokąta (każdy prostokąt ma dwa takie boki). Takie zatrzymanie nazywamy zdarzeniem. W czasie zatrzymania obliczamy, jaka jest długość przecięcia wszystkich prostokątów z linią zamiatającą. Oznaczmy tę długość D. Nazwijmy ją wysokością. Teraz wiemy, że ta wysokość nie zmieni się do następnego zdarzenia. Dlatego jeśli jedno zdarzenie zdarzyło się na współrzędnej x1, a następne na x2, to pole prostokątów między tymi zdarzeniami wynosi (x2-x1)*D. Suma takich pól jest szukanym polem.

Szczegóły (zakładamy, że przechodzimy od lewej do prawej):

1. Użyj tej samej struktury, co w etapie 1. Stwórz listę zdarzeń odpowiadających pionowym bokom prostokąta. Tym razem wartość logiczna oznacza, czy to jest lewy bok prostokąta (bok otwierający prostokąt), czy prawy (zamykający prostokąt).

2. Posortuj zdarzenia (pionowe boki prostokąta) po x-owej współrzędnej.

3. Stwórz pomocniczą listę odcinków. Głównym problemem jest obliczenie przecięcia linii zamiatającej z prostokątami. Będziemy sobie radzić tak:

   - gdy natrafimy na odcinek, który otwiera prostokąt, będziemy go wrzucać na listę pomocniczą.
   - gdy natrafimy na odcinek, który kończy prostokąt, to będziemy z tej listy usuwać odpowiadający mu odcinek, (który ten prostokąt otwierał)

   W ten sposób zawsze na liście będziemy mieli odcinki wyznaczające zakresy pionowe prostokątów, z którymi linia zamiatająca się przecina w danym zdarzeniu. Oczywiście, one mogą się nakładać, tak jak prostokąty. Dlatego robiliśmy pierwszy etap.

4. Teraz przechodzimy linią zamiatającą. Wrzucamy i wyrzucamy odpowiednie odcinki z listy pomocniczej z poprzedniego punktu. Obliczamy (x2-x1)*D za ten fragment, który przebyliśmy od poprzedniego zdarzenia. Obliczamy, korzystając z etapu 1, łączną długość odcinków na liście pomocniczej. To jest nasze D dla następnej iteracji.

5. Zwracamy sumę pól.