

Algorytm

1. Dzielimy odcinki na poziome i pionowe.
2. Punkty końcowe i początkowe każdej z tej grupy odcinków sortujemy.
3. W ten sposób na każdej z osi x i y stworzyło się nam $O(n)$ punktów kontrolnych, czyli takich, które nas interesują bo jakiś odcinek się tam zaczyna lub kończy.
4. Sortujemy te punkty po pierwszej współrzędnej, następnie robimy kopie posortowanej tablicy i sortujemy je po 2 współrzędnej.
5. Oryginalnym odcinkom przypisujemy miejsca w posortowanych tablicach ich końcowego i początkowego wierzchołka.
6. Tworzymy drzewo przedziałowe 2D z leniwą propagacją dla każdej osi X, Y , które zlicza ile odcinków jest na danym przedziale, gdzie przedziały są od jakiegoś punktu początkowego/końcowego do innego punktu początkowego/końcowego odcinka z danej osi.
7. Tworzymy dwie zmienne m - zliczającą maksymalną liczbę punktów i l - trzymającą odcinki, które przecinają się z m punktami.
8. Dla każdego odcinka wyszukujemy w drzewie przedziałowym przeciwnej osi z iloma odcinkami się on przecina. Żeby móc wyszukać musimy poznać współrzędne, które nas interesują jako, że w drzewie przedziałowym trzymamy zmapowane współrzędne. Żeby to zrobić wyszukujemy binarnie w posortowanej liście punktów pierwszego nie większego punktu od naszego względem danej współrzędnej. Po wyszukaniu aktualizujemy m i l .
9. W m i l mamy wynik.

Złożoność

Pamięć:

$O(n)$

Czas:

Posortowanie punktów: $O(n \log(n))$

Stworzenie drzewa przedziałowego: $O(n \log(n))$

Binsearch dla każdego: $O(n \log(n))$

Wyszukanie w drzewie przedziałowym dla każdego: $O(n \log(n))$

Łącznie: $O(n \log(n))$