

Orthogonal Range Query 2D. Fractional cascading

Кочетов К.С. М4139

Задача:

Есть множество точек на плоскости, необходимо реализовать структуру, содержащую эти точки и поддерживающую запрос — найти точки, содержащиеся в прямоугольнике (x_1, y_1, x_1+w, y_1+h) . Асимптотика запроса $O(\log n)$

Протестировать корректность и скорость работы.

Реализация:

1. Класс `FCTree` содержит реализованную структуру
 - a. Метод `build` рекурсивно строит дерево по исходному множеству точек. Вершина дерева содержит кроме самой точки, еще ссылки на детей и сортированный массив 'y-ов' поддерева, где для каждого элемента хранится ссылка на ближайшего левого соседа в левом и правом поддеревьях. Это знание необходимо для алгоритма fractional cascading, чтобы не делать бинпоиск в каждом сортированном массиве 'y', а просто итерироваться по дереву за $O(1)$ после бинпоиска в одном, главном узле.
 - b. `getPointByRect` – метод, реализующий запрос на поиск точек в прямоугольнике. Сначала находим вершину, содержащуюся в $[x_1; x_2]$, затем делаем два бинпоиска в массивах 'y' его поддеревьев и в конце делаем проход по левому и правому поддереву, объединяем и возвращаем найденные точки.
2. Класс `NaiveSolver` реализует наивный метод решения поставленной задачи. Работает за линейное время
3. Класс `BinTree` реализует такую же структуру, как и `FCTree`, только без каскадирования. Работа с ссылками при поиске 'y' в массиве 'y-ов' заменена на бинпоиск по 'y'.
4. Класс `Tester` содержит тест на корректность (в сравнении с наивной реализацией), а также сравнение на скорость работы, результаты которой представлены ниже в таблице.
5. Класс `Utils` содержит простейшую реализацию метода бинарного поиска в отсортированной последовательности.

Сравнение по скорости:

Для получения следующих результатов выполнялось по 20000 запросов на поиск точек в прямоугольнике

Количество точек	FC time, сек	BinTree time, сек	Naive time, сек
2048	0.1602	0.2027	0.3438
8192	0.2141	0.2861	0.6930
32768	0.4625	0.6167	17.5055
131072	0.5216	1.0252	35.2189
524288	0.5468	3.7468	324.4263