

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 отсортированный 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 бинарный поиск в каждом отсортированном массиве 'y', а просто итерироваться по дереву за $O(1)$ после бинарного поиска 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 бинарных поиска 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 `Tester` содержит тест на корректность (в сравнении с наивной реализацией), а также сравнение на скорость работы, результаты которой представлены ниже в таблице.
4. Класс `Utils` содержит простейшую реализацию метода бинарного поиска в отсортированной последовательности.

Сравнение по скорости:

Для получения следующих результатов выполнялось по 5000 запросов на поиск точек в прямоугольнике

Количество точек	FC time, сек	Naive time, сек
32	0.1182	0.0068
128	0.0192	0.0075
512	0.0849	0.0310
2048	0.0380	0.1267
8192	0.1803	0.4629
32768	0.2124	6.1895
131072	0.2085	9.4562
524288	1.4461	94.7736

Тенденция заметна. Вначале FC проигрывал из-за затрат на построение дерева, которые окупаются при увеличении количества точек в структуре.