

Выполнил Газизуллин Нияз Айратович 22ПИЗ

Контекст сдан с почты: nagazizullin@edu.hse.ru

Лабораторная работа №2

Цель

Реализовать три алгоритма поиска числа прямоугольников, к которым принадлежит точка. Выяснить эффективность алгоритмов при разном объеме начальных данных и точек.

Алгоритмы

1) Алгоритм перебора

Препроцессинг:

Отсутствует

Работа алгоритма:

Поочередный перебор прямоугольников для проверки принадлежности точки к текущему прямоугольнику.

2) Алгоритм на карте

Препроцессинг:

Сжимаются координаты прямоугольников вдоль осей и строим по ним матрицу (карту), где у каждой точки будет количество прямоугольников, которые охватывает эта точка.

Работа алгоритма:

Сжимаются координаты точки вдоль осей и находится соответствующая ячейка на карте.

3) Алгоритм на дереве

Препроцессинг:

Сжимаются координаты прямоугольников вдоль осей. Затем для каждого элемента 1 из осей (в моем случае оси X) строится персистентное дерево отрезков, которое будет модифицироваться, при переходе к следующему элементу оси. Дерево будет хранить полуинтервал и количество прямоугольников этом полуинтервале.

Работа алгоритма:

Сжимаются координаты точки вдоль осей. Затем находится соответствующее дерево отрезка (по оси X) и осуществляется проход (по Y) по этому дереву с подсчетом значений.

Генерация входных данных

Количество прямоугольников - от 1 до 2^{10}

Количество точек - 1000

Для тестового набора прямоугольников используется набор вложенных друг-в-друга с координатами с шагом больше 1 - $\{(10*i, 10*i), (10*(2N-i), 10*(2N-i))\}$.

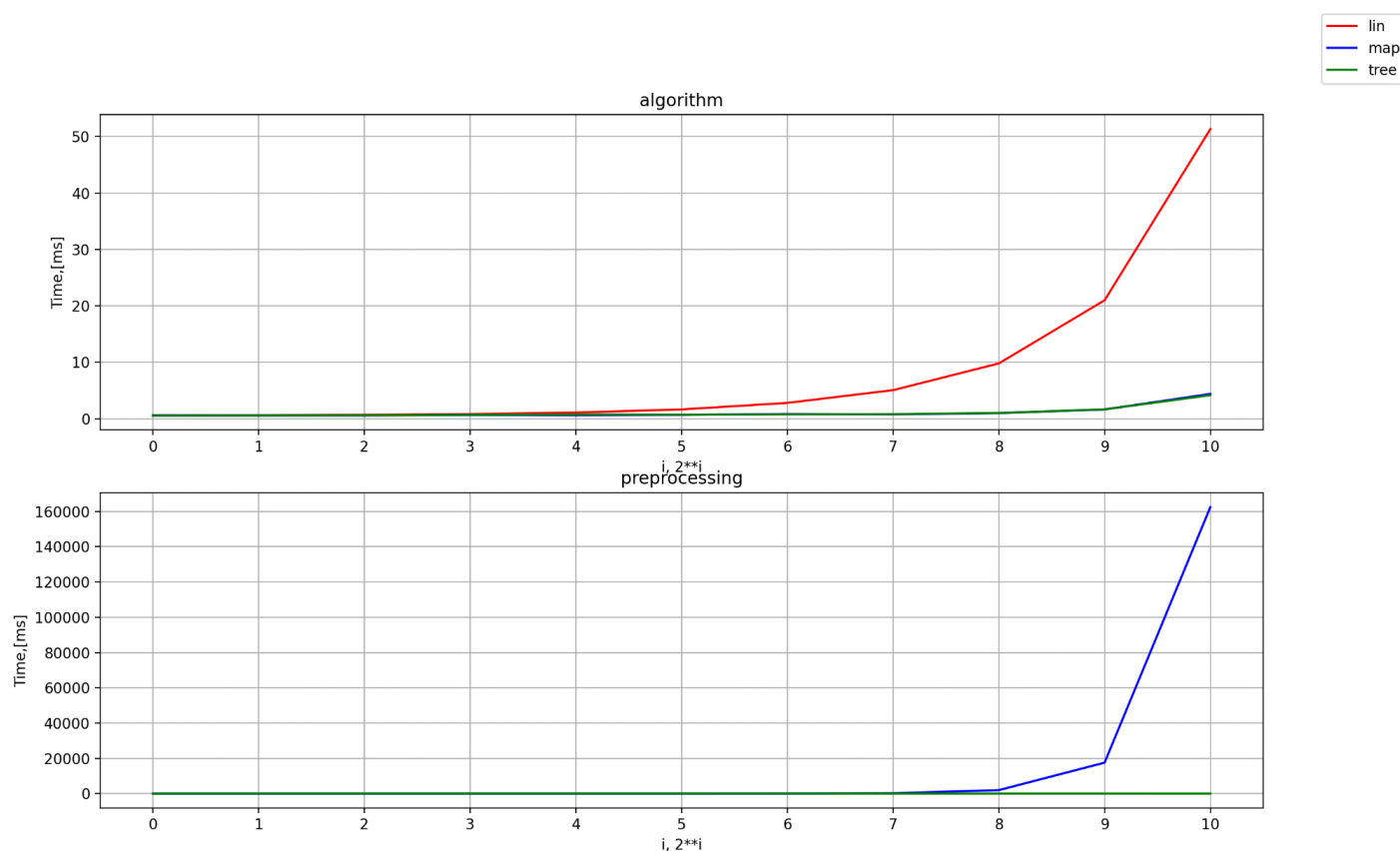
Для тестового набора точек, рекомендуется используется неслучайный набор распределенных более-менее равномерно по ненулевому пересечению прямоугольников - хэш функции от i с разным базисом для x и y .

$(p*i)^{31\%(20*N)}$, p -большое простое, разное для x и y

1423 - для X

1583 - для Y

Графики



Вывод

Исходя из результатов проделанной работы, можно сделать вывод, что алгоритм перебора является самым эффективным решением при малом количестве входных данных, так как не требует подготовки. Также стоит

заметить, что алгоритм на карте и алгоритм на дереве лучше работают при больших данных и показывают примерно одинаковое время работы алгоритма. Однако, алгоритм затрачивает много время на подготовку, что делает алгоритм на персистентном дереве отрезков лучшим вариантом при большом объеме входных данных.