*Выполнил Овсянников Артем, 22ПИ-1*

*Контест сдан с почты:* [***asovsyannikov\_2@edu.hse.ru***](mailto:asovsyannikov_2@edu.hse.ru)

***Комменатрий:***

***data*** *- исходные данные и полученные замеры*

***algorithms*** *- три алгоритма поиска*

***tests*** *- тестирование алгоритмов*

Лабораторная работа №2

**Цель:** реализовать три алгоритма для поиска, скольким прямоугольникам принадлежит точка. Выяснить при каком объеме начальных данных и точек какой алгоритм эффективнее.

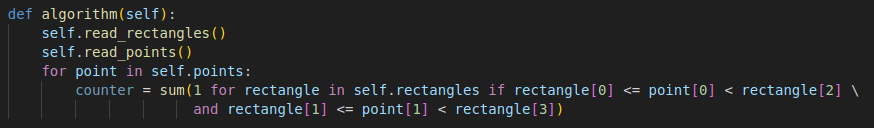
**Алгоритмы:**

1. *Алгоритм перебора*

Этот алгоритм не нуждается в предварительной обработке входящих данных.

Он последовательно проверяет все прямоугольники для каждой отдельной точки, чтобы определить, находится ли она внутри границ прямоугольника.

Эффективность алгоритма линейно зависит от количества прямоугольников в наборе данных.

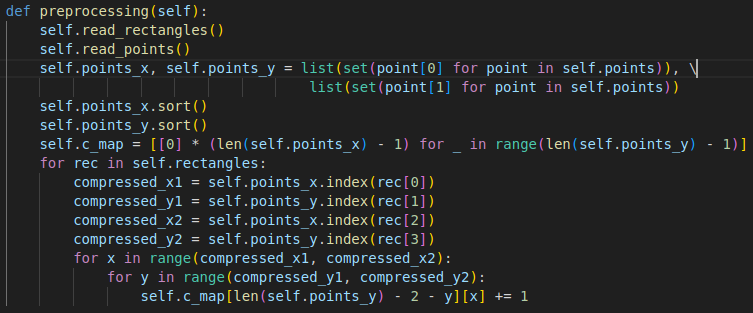


1. *Алгоритм на карте*

Перед началом работы алгоритма необходимо сжать координаты прямоугольников вдоль каждой оси. Это достигается путем создания двух массивов, содержащих уникальные координаты X и Y соответственно.

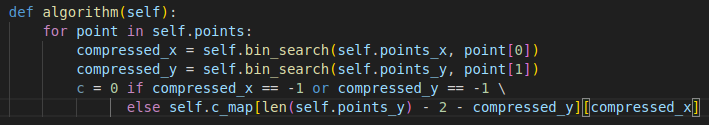
После этого мы имеем два массива, где каждой сжатой координате точки, определяющей прямоугольник, соответствует ее индекс X и Y в этих массивах. Затем строится карта, где каждой точке присваивается количество прямоугольников, которые ее охватывают.

Время работы алгоритма подготовки кубически зависит от входных данных.



Далее нужно сжать координаты точки и затем проверить, принадлежит ли она нашей области или нет, а затем выводить ответ.

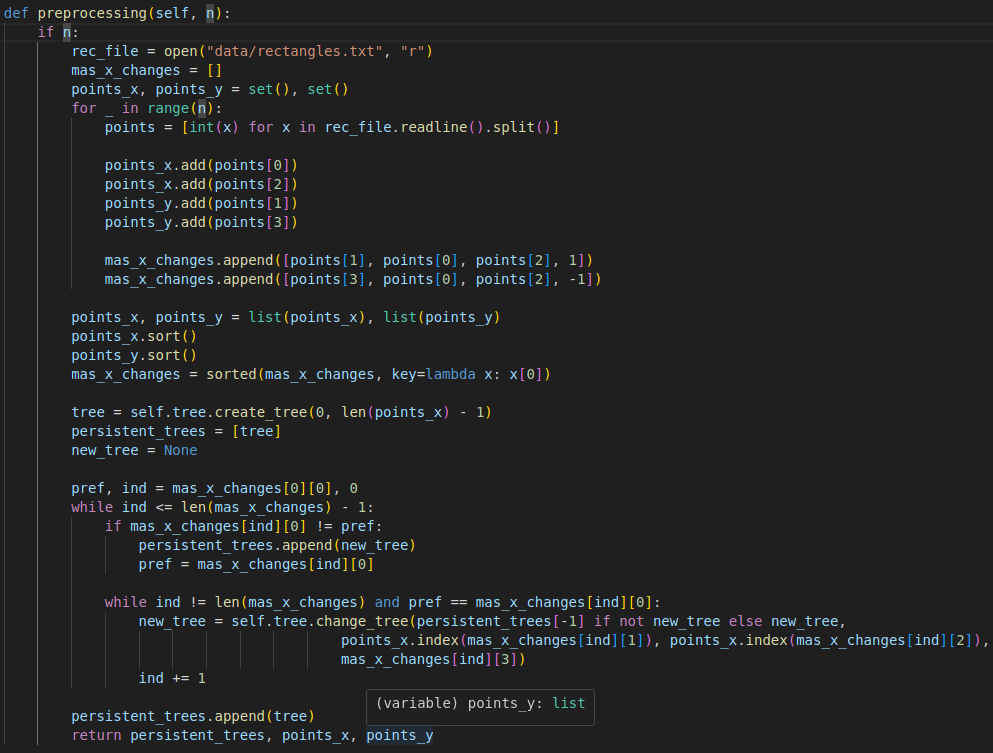
Эффективность данного алгоритма логарифмически зависит от входных данных.



1. *Алгоритм на дереве*

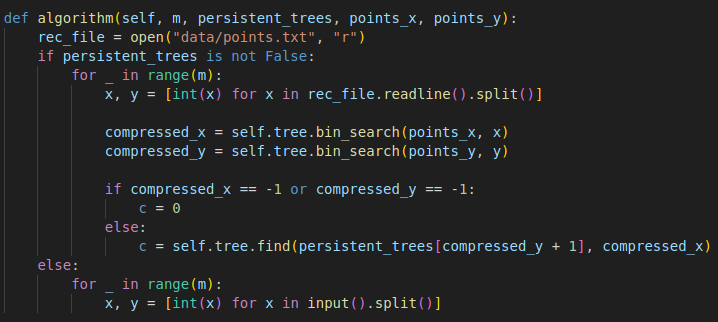
Алгоритм формирует уникальные массивы координат X и Y для прямоугольников. Используя эти массивы, происходит сжатие координат точек. Затем создается дерево отрезков, где для каждой Y-координаты, обозначающей край прямоугольника, генерируется новая версия дерева. После этого строится персистентное дерево отрезков на основе этих модификаций.

Алгоритм подготовки в данном случае линейно-логарифмически зависит от входных данных.



Выбирается нужная версия дерева по по сжатой координате Y(тот же способ сжатия) и происходит проход по дереву и подсчету количества прямоугольников, которым точка принадлежит.

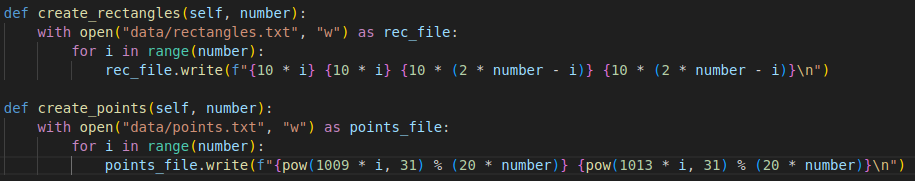
Поиск ответа на вопрос поставленной задачи зависит логарифмически от входных данных.



**Генерация входных данных:**

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))}.

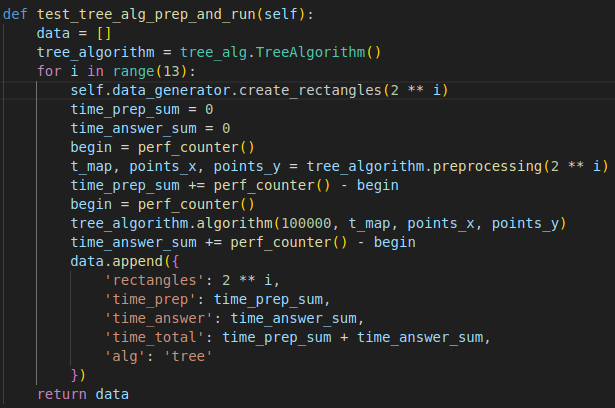
Для тестового набора точек используется неслучайный набор распределенных более-менее равномерно по ненулевому пересечению прямоугольников (хэш функции от i с разным базисом для x и y). (p\*i)^31%(20\*N), p-большое простое, разное для x и y (выбраны 1009 и 1013).



**Тестирование:**

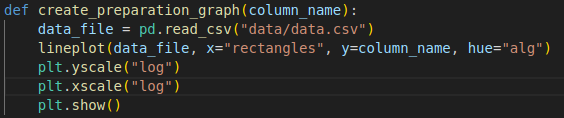
Для каждого алгоритма проводилось тестирование - замер времени подготовки и ответа. Количество прямоугольников - от 1 до 2 \*\* 12, количество точек - 100000.

Пример тестирующей функции:



**Визуализация результатов тестирования:**

На основе полученных тестирующими функциями данных, были построены графики зависимости объема входных данных и времени работы алгоритма.



Графики:

