Необходимо выстроить архитектуру, которая будет позволять за счет инъекций функций построить спектр алгоритмов, которые будут решать разного рода задачи, использующие в основе алгоритм поиска точек пересечения множества отрезков.

В рамках данной работы предлагается разработать библиотеку, содержащую реализацию:

* набора базовых геометрических объектов (точка, отрезок, цепь, контур)
* набора контейнеров для них (набор отрезков, набор контуров), специальные контейнеры, удовлетворяющие некоторым условиям (слой отрезков, слой контуров; условия будут описаны далее)
* базовой структура, в основе которой лежит красно-черное дерево
  + будет использоваться в реализации метода заметающей прямой
  + имеет следующие операции: добавление элемента, поиск максимального и минимального элемента, удаление, поиск следующего/предыдущего для заданного элемента, позиционный доступ к элементу (по индексу). Сложность всех операций – логарифмическая
* алгоритма поиска пересечения пары отрезков (константная сложность)
* алгоритма поиска пересечения множества отрезков (на базе метода заметающей прямой, сложность )
* набора функций преобразования (все имеют сложность )
  + набор отрезков -> слой отрезков
  + слой отрезков -> набор отрезков
  + слой контуров -> набор контуров
  + набор контуров -> слой отрезков

Библиотека должна уметь выполнять чтение данных в текстовом формате. Структура и формат текстового файла необходимо согласовать с заказчиком.

***Описание объектов***

1. Точка – структура, описывающая геометрическую точку в двумерного пространстве (хранит в себе две координаты – x, y).
2. Отрезок – геометрический объект, состоящий из начальной и конечной точки.
3. Набор отрезков – представляет собой структуру (контейнер), которая хранит в себе множество отрезков
4. Цепь – набор отрезков, в котором конец каждого отрезка совпадает с началом последующего
5. Контур – цепь, в которой начальная точка первого отрезка совпадает с конечной точкой последнего отрезка
6. Метка – специальный маркер, показывающей значение некоторого признака у объекта (например, принадлежность контуру)
7. Слой отрезков – набор отрезков, которые могут касаться друг друга (в т. ч. только в одной точке), но не пересекаются
8. Слой контуров – множество непересекающихся контуров, для которых задан лексикографический порядок на основе меток. Пример задания порядка: каждый контур имеет метку с номером. Это значит, что контуры будут храниться в слое в порядке возрастания номеров. Также нужно рассмотреть вариант работы с этим объектом как с множеством отрезков, принадлежность контуру каждого определяется на основе меток.

***Функции поиска точек пересечения***

1. Функция поиска точек пересечения двух отрезков
   1. Принимает на вход два отрезка
   2. Результат работы: точка пересечения отрезков
   3. Сложность O(1)
2. Функция поиска точек пересечения множества отрезков. Должна быть реализована на основе метода заметающей прямой с использованием красно-черного дерева.
   1. Принимает на вход набор отрезков, функцию разметки ребер (принимает на вход ребро, возвращает значение метки), функцию обратного вызова (callback), которая принимает в качестве входных аргументов два пересекающихся отрезка и точку их пересечения, возвращает булево значение, сигнализирующее, нужно ли продолжать поиск точек пересечения
   2. Функция ничего не возвращает. Уведомление о результате работы выполняется с помощью callback функции путем вызовов ее с передачей пары пересекающихся отрезков и точки их пересечения.
   3. Cложность

***Функции преобразования некоторых объектов***

1. Выполняются за счет вызова функции поиска точек пересечения отрезков. Cложность
   1. Преобразование набора отрезков в слой отрезков. Выполняется за счет вызова функции поиска точек пересечения отрезков.
   2. Преобразование набора контуров в слой контуров
2. Требуют минимальных преобразований
   1. Преобразование слоя отрезков в набор отрезков (не требует преобразований с точки зрения алгоритмов и способа хранения в памяти, сложность константная)
   2. Преобразование слоя контуров в набор контуров (не требует преобразований с точки зрения алгоритмов и способа хранения в памяти, сложность константная)
   3. Преобразование набора контуров в слой отрезков (результат работы – объединение множеств отрезков каждого контура в единое множество отрезков, сложность линейная)

***Логические функции***

1. Для слоя отрезков

Функция принимает на вход слои отрезков (или наборы отрезков), на выходе возвращает слой отрезков, отфильтрованный по меткам. Т. е. данная функция выполняет поиск точек пересечения между всеми отрезками, размечает все отрезки значениями меток; задается набор допустимых меток, разметка отрезков выполняется с помощью инъекции функции разметки ребер в функцию поиска пересечения отрезков, а затем функция фильтрации по меткам выделяет отрезки, метки которых соответствует критериям отбора этой функции.

1. **Для слоя контуров**

Функция принимает на вход слои контуров (или наборы контуров), на выходе возвращает слой контуров. В процессе работы функции задается набор допустимых меток, выполняется поиск точек пересечение всех отрезков, далее выполняется анализ областей – их разметка с помощью инъекции функции разметки областей. После выполняется фильтрация областей по меткам, выделяется слой отрезков, соответствующий критериям отбора.

***Демонстрационные примеры***

Эти примеры будут являться демонстрацией работы функций библиотеки, показывать полный цикл обработки входных данных по некоторому алгоритму, а также будут использоваться для замеров производительности библиотеки в целом. В качестве таких примеров предлагается реализовать функцию логического «И» для слоя отрезков и логического «Или» для слоя контуров.

Идея структуры «Pipeline» для реализации логических функций

Предлагается не реализовывать отдельно 2 схемы логических функций, а реализовать структуру **«**Pipeline», которая будет конфигурироваться логическими блоками, выполняющими преобразования и модификации. Данная структура будет предоставлять интерфейс установления порядка выполнения логических блоков, а также их связывания.

Список логических блоков:

1. Блок установки меток
2. Блок поиска точек пересечения пары отрезков
3. Блок поиска точек пересечения множества отрезков
4. Блок фильтрации
5. Блок преобразования (переход от набора отрезков к слою отрезков и т. д.)

Каждый из этих блоков также является конфигурируемым. Например, блок поиска точек пересечения множества отрезков «конфигурируется» функцией разметки ребер.

Такой подход, возможно, позволит проще выстраивать логические функции, которые в своей основе используют поиск точек пересечения отрезков.