/\*\*

**@file** algorithms/monotone\_*polygone\_triangulation.dox*

*@author Taisiya Osipova*

**@page** graham\_*scan Триангуляция монотонных полигонов*

*@section overview Обзор*

Алгоритм для построения

Реализация использует подход с построением левой и правой цепей

**@section** complexity Сложность

- Сложность алгоритма: O(n log n)

- Память: O(n)

n - количество вершин полигона

**@section** implementation Реализация

1. Многоугольник делится на левую и правую монотонные цепи относительно самой верхней и самой нижней вершин

2. Цепи сортируются по убыванию координаты y

3. Вершины из обеих цепей объединяются в один упорядоченный список (merged), сохраняя порядок по y

4. Вершины обрабатываются в порядке объединённого списка:

- Если текущая вершина и вершина на стеке принадлежат разным цепям, добавляются диагонали ко всем вершинам в стеке

- Если они в одной цепи, проверяется выпуклость (через векторное произведение), и добавляются диагонали, пока условие выполняется

5. Оставшиеся вершины в стеке соединяются диагоналями с последней обработанной вершино

**@subsection** details Детали реализации

1. Сортировка точек.

- В порядке убывания y-координаты

- В случае равенства по возрастанию x-координаты

2. Построение левой цепи.

- Последовательно добавляем точки, лежащие от самой верхней вершины к самой нижней по часовой стрелке

3. Построение правой цепи.

- Последовательно добавляем точки, лежащие от самой верхней вершины к самой нижней против часовой стрелки

4. Вспомогательные функции.

- crossProduct(a, b, c) вычисляет векторное произведение

- isPolygonEdge(polygon, end1, end2) проверяет, является ли отрезок стороной полигона

\*\***/**