Exercises - Lists

Assignment

Create functions on lists in the context of points, line segments, lines and polygons.

Description

Lists can be used to represent many things. In this exercise we will use lists to represent points, line segments, lines, and polygons.

Using these representations we will write functions that operate on these spatial objects and determine interesting properties of these objects.

Assignment

Write the following functions:

- Function p_distance, which determines the distance between two points.
- Function s_length, which determines the length of a line segment.
- Function l_length, which determines the total length of a line.
- Function s_areCrossing, which determines whether two line segments are intersecting. Returns a boolean.
- Function l_selfIntersect, which determines whether a line intersects itself. Returns a boolean. (needs iteration!).
- Boolean function l_isClosed, which determines whether a line is closed.
- Function countIntersects, which determines how many times a line segment intersects with a polygon. (iteration!)
- Boolean function p_inPolygon, which determines whether a point is inside a polygon.
- Boolean function pol_contains, which determines whether one polygon is entirely inside another polygon.

Hints

Points look like [1, 2] etc.

Line segments look like [[1, 2], [3, 4]] etc.

Lines look like [[1, 2], [3, 4], [5, 6], [7, 8]] etc.

Polygons look like [[1, 2], [3, 4], [5, 3], [1, 2]] etc.

Whether lines are intersecting or not can be determined from the values resulting from the following formula:

$$v = \frac{y_2 - y}{y_2 - y_1} - \frac{x_2 - x}{x_2 - x_1}$$

If v is equal to zero, then point [x, y] is on the line running through the points [x_1, y_1] and [x_2, y_2]. All points on one side of the line have the same sign. In other words, two different points having an opposite sign of v are located on opposite sides of the line through the points [x_1, y_1] and [x_2, y_2]. Thus, by using the value v in a smart way we can discover whether two line segments [[x_1, y_1], [x_2, y_2]] and [[x_3, y_3], [x_4, y_4]] are intersecting! We will have to test the points of line segment 1 on the line segment 2 *and vice versa!*

The rest is rather a matter of administration. More complex functions are built using simpler functions.

Functions that work on line segments may be extended to lines by iterating over the line segments contained by the lines.

The difficult part of this exercise may be in deciding when to include the end points of the line segments and when not to include them. Especially when dealing with polygons, some exceptions may be difficult to deal with in a general sense!

The point-in-polygon problem can be solved by taking the point of which we want to know whether it is inside or not, and create a line segment to infinity and count the number of intersections with the polygon. Odd counts indicate that the point is inside, even counts tell us that the point is outside the polygon. In the figure below the count is 3, which means the point is indeed inside the polygon.