TP5: Convex Hulls and Digital Convex Hulls

In this TP, the idea is to implement a convex hull algorithm and to experiment its complexity (number of edges) on digital objects.

Preliminaries Writing svg files

In this lab, you will need to write images of points and lines. A practical way is to write svg files, which are text files using an xml markup language. They can be directly opened, for instance, in your browser. A simple svg file writing some text and plotting one dot and one line looks like:

```
<svg width="512" height="512"  xmlns="http://www.w3.org/2000/svg">
  <line x1="15" y1="5" x2="500" y2="95" stroke="red" />
  <circle cx="256" cy="256" r="5" fill="blue" />
  <text x="210" y="60">Text</text>
</svg>
```

Exercise 1 Orientation predicate

Question: Implement the $Orientation(p, q, r)$ predicate as discussed in the lecture.

- To avoid numerical issues, we encourage you to consider points in $\mathbb{Z}^2$ on a digital domain (instead of $\mathbb{R}^2$).
- To display a line in `DGtal` between `Points` p and q, you can use the following method of the `Board` class: `board.drawLine( p[0], p[1], q[0], q[1]);`

Question: Test the *Orientation* predicate with an implementation of the segment-segment intersection detection (cf lecture). Experiment your intersection test on all cases (regular intersection, alignement, no intersection, intersection point is a vertex, ...).

The rest of the TP focuses on the implementation and the experimentation of a *convex hull algorithm*, namely, Graham's scan algorithm. At the end we would like you:

- To test the convex hull construction on point sets defined by the digitization (at a given resolution h) of a disc defined as a digital set. You will write an svg file to display your result.
- To plot (using `gnuplot` or svg) the number of edges when $h \to 0$ in log-scale. The aim is to observe the $N^{2/3}$ behavior we discussed in the lecture for convex hull in $N \times N$ domains.

Graham's scan implementation on the point set ($O(n.logn)$) should only use the $Orientation(p, q, r)$ predicate and point coordinate comparisons.

Exercise 2 - Convex hull

Questions:

- Implement the Graham's scan algorithm as described in the lecture:
  - First, sort the points by polar angle (cf the `qsort` C function man page or the C++ `std::sort` to do the sort). As discussed in the lecture, you would just have to replace the comparison function/functor by the Orientation predicate with fixed p_0.
  - The second step consists in a simple stack based removal (using for example `std::queue`).
- Display the number of edges as a function of h in a graph to observe the $N^{2/3}$ behavior.

Note that Graham's scan can be sped up just considering border point and not interior points (for experiments on digital shapes).