Exercise 5

Compute a Delaunay triangulation of a set of points in general position

In this laboratory I used the same data structure and algorithm for the triangulation, that is explained below.

For triangulating a set of points I first built a DCEL data structure made of Vertices, Edges and Faces. A Vertex stores its position and an edge that starts from it. An Edge stores the Vertex from which it starts, the Face that it has to the right, its next Edge and its twin Edge (the same edge but starting from the other Vertex). Finally, A Face stores one of the Edges that have it to the right. All instances are stored in vectors of pointers and each structure has pointers to the data referenced. In the end all instances are deleted.

The algorithm implemented is the incremental algorithm with auxiliary point explained in class. But in this case, after inserting each point I apply the Delaunay property as explained in the laboratory documentation.

Lanzarote input file: dealing with the boundary

For removing the boundary I followed the indexes of the boundary vertex. The algorithm advances by rotating over the edges, of the current vertex, to the right of the boundary until the next boundary vertex is found and removes the faces to the left of the edges at every step.

In my implementation, there is a hack such that the algorithm finishes as for one vertex it keeps rotating around it (probably because of a bug in the consistency of the DCEL that I couldn't find). The fragment of code is commented explaining why it is there.