2 Delaunay triangulation: definitions, motivations, properties, classical algorithms.

2.1 Drawing

Draw the Delaunay triangulation of the attached point set.

2.2 Nearest neighbor graphs

S a set of n points. $q_0 \in S$. Let q_1 denote the nearest neighbor of q_0 in $S \setminus \{q_0\}$. Let q_2 denote the second nearest neighbor of q_0 in S, i.e., the nearest neighbor in $S \setminus \{q_0, q_1\}$. Similarly q_i the i^{th} nearest neighbor.

The directed nearest neighbor graph of S is the graph whose vertices are the points in S and pq is an edge of the graph if q is the nearest neighbor of p.

Fact: The degree of the nearest neighbor graph is ≤ 6. (proof optional).

2.2.1 Nearest neighbor

Prove that q_0q_1 is an edge of the Delaunay triangulation of S.

2.2.2 Second nearest neighbor

Prove that q_0q_2 or q_1q_2 is an edge of the Delaunay triangulation of S.

2.2.3 k^{th} nearest neighbor

Prove that $\forall k \exists i < k$ such that q_kq_i is an edge of the Delaunay triangulation of S.

2.2.4 Nearest neighbor graph

Write an algorithm that takes the Delaunay triangulation of S and output the directed nearest neighbor graph of S.

You can write things like:
`for` v `enumerating all vertices of` $DT(S)$,
`for` w `enumerating the neighbor of` v `in` $DT(S)$,
or `output edge`(v, w),
or $v.color = red$ to add some information in a vertex (or edge or...)

What is the complexity of this algorithm?

2.2.5 Nearest neighbor graph

Write an algorithm that takes the Delaunay triangulation of S and output the directed second nearest neighbor graph of S.

What is the complexity of this algorithm?

Draw the Delaunay triangulation