Exercise 1 (Delaunay Triangulations and MSTs):
Show: For a point set $S$, a minimum spanning tree of $S$ is a subset of the Delaunay triangulation of $S$.
(15 points)

Exercise 2 (Delaunay Triangulations and Edge Flips):
Show for the edge-flip algorithm (Algorithm 1) from the tutorial that for every $n$, there is a triangulation of $n$ points that requires $\Omega(n^2)$ flips to transform it into a Delaunay triangulation.
(15 points)

Exercise 3 (Lower Bound for Voronoi):
Assume that an algorithm $ALG$ is able to compute the Voronoi diagram of a point set $P$ with $|P| = n$ in $O(f(n))$ with $f(n) \in o(n \log n)$ (that is, faster than $O(n \log n)$).
Prove that this is not possible by showing that you could use $ALG$ to sort $n$ numbers in $O(n + f(n))$.
(15 points)
Exercise 4 (Voronoi Lookup):
Given: A point set $P$ with $n \in \mathbb{N}$ points and its Voronoi diagram $Vor(P)$.

Give a method for the construction of a data structure that allows to find the nearest site for an arbitrary point $q \in \mathbb{R}^2$ (that is, to determine in which cell $q$ lies).

Generating the data structure can be arbitrarily complex; once it is established, it should allow finding the next site for arbitrary $q$ in time $O(\log n)$. Explain why your method complies with a lookup time of $O(\log n)$.

Hint: In case $q$ is located on a Voronoi edge or a Voronoi vertex the next site is not uniquely defined. Your lookup should simply give one of those next sites.

(15 points)