1. Let R be a set of n red non-intersecting segments in the plane and let B be a set of n blue non-intersecting segments. Assume that no two segment intersect in more than a single point. Moreover assume R and B contain no vertical segements.

   Design an algorithm that *counts* the number of intersection points between red and blue segments. It should work in time $o(n^2)$.

   As a hint consider the following invariant for a sweep algorithm (where for various reasons we have the sweep line sweeping from left to right over the plane).

   At some point of time t let R_t be the set of red segments that intersect the sweep line L_t and let B_t the set of blue segments that intersect L_t. Similarly let $R_{<t}$ and $B_{>t}$ be the sets of red and blue segments to the left of the sweep line. For a non-vertical segment s let $\vec{s}$ denote the ray (halfline) that contains s and starts at the left endpoint of s.

   Invariant: The algorithm maintains the number X, which is the number of intersection points between segments in $R_{<t}$ and $B_{>t}$ plus the number of intersection points between rays $\vec{r}$ and $\vec{b}$ with $r \in R_t$ and $b \in B_t$.

   You may make use of a data structure that stores a set of pairs of numbers (u, v) with operations $\text{Insert}(u, v)$ and $\text{Delete}(u, v)$ and which for $\text{Query}(a, b)$ returns the number of pairs (u, v) stored in the structure with $u < a$ and $v > b$. You may assume that the update operations take logarithmic time, whereas a query takes time $O(\log^2 n)$.

2. Prove that for any $n > 3$ there is a set of n sites in the plane such that one of the cells of $\text{VoD}(T)$ has $n - 1$ vertices.

3. Prove or disprove that the points of a set T whose Voronoi regions are unbounded are exactly the ones that lie on the boundary of the convex hull of T.

4. For a set T of points in the plane the Euclidean minimum spanning tree, is the minimum spanning tree of the complete graph on T where for $p, q \in T$ the edge connecting p and q is assigned the distance between p and q as its weight.

   (a) Prove that the Euclidean minimum spanning tree of T is a subgraph of the Delaunay triangulation of T.

   (b) Give a fast algorithm for computing the Euclidean minimum spanning tree of T. What running time can you achieve?