This homework exam has 5 questions for a total of 90 points. You can earn 10 additional points by a careful preparation of your hand-in: using a good layout, good spelling, good figures, no sloppy notation, no statements like “The algorithm runs in $n \log n$.” (forgetting the $O(\cdot)$ and forgetting to say that it concerns time), etc. Use lemmas, theorems, and figures where appropriate. Your final grade will be the number of points divided by 10. Unless stated otherwise, both randomized and deterministic solutions are allowed. In case you are asked to analyze the running time and your algorithm is randomized, analyze its expected running time.

### Question 1 (15 points)

Let $S$ be a planar subdivision with $n$ vertices, represented as a DCEL. Give pseudo-code for an algorithm that, given a pointer to a vertex $v$, test if $v$ is incident to a face that is a triangle. Your algorithm should use the ‘Twin’, ‘NextEdge’, ‘PrevEdge’, etc. fields to navigate (i.e. you cannot assume you can directly access a list of vertices). Describe in a few sentences the main idea of your algorithm and give its running time.

### Question 2 (10 points)

Let $P$ be a set of $n$ points in $\mathbb{R}^2$, and let $R$ be the shortest (in terms of Euclidean length) closed curve such that all points of $P$ lie inside (or on the boundary of) the area enclosed by $R$. Prove that $R$ is the convex hull $CH(P)$ of $P$.

### Question 3 (20 points)

Given a set $R$ of $n$ “red” points and a set $B$ of $n$ “blue” points in $\mathbb{R}^2$. Develop an expected $O(n)$ time algorithm that can test if there exists a line $\ell$ that separates $R$ from $B$, that is, such that all points in $R$ lie right of $\ell$ and all points in $B$ lie left of $\ell$. Prove that your algorithm is correct and achieves the desired running time. You may assume that any line contains at most two points of $R \cup B$ (i.e. there are no three colinear points).

### Question 4

Let $p \in \mathbb{R}^2$ be a point, and let $S$ be a set of $n$ disjoint open-ended line segments in the plane. You may assume that the set containing $p$ and all endpoints of the segments in $S$ has no three colinear points (and thus $p$ does not lie on any of the segments).

1. **(10 points)** Develop an algorithm to compute the length of a longest (open) segment $s$ that contains $p$ but does not intersect any segment in $S$. If segment $s$ does not exist your algorithm should return $\infty$. Prove that your algorithm is correct and analyze its running time.

   Note: the number of points rewarded for this question will depend on the running time of your algorithm.

2. **(8 points)** Is your algorithm still correct if the segments in $S$ may intersect? If so, argue why, if not, give an example why not, and describe how to fix it. You do not have to argue about the running time of your algorithm in this scenario.
Question 5

Let $P$ be a set of $n$ points in $\mathbb{R}^2$, let $D(c)$ be a unit disk, that is, a disk of radius one and center $c$, and let $P_c = P \cap D(c)$ be the subset of $P$ that lies in a unit disk centered at $c$.

1. (10 points) Prove that there are at most $O(n^2)$ different sets $P_c$ over all points $c \in \mathbb{R}^2$.

2. (7 points) Give a construction that shows that the above bound is tight in the worst case. In other words, show that there can sometimes be $\Omega(n^2)$ different sets $P_c$.

3. (10 points) Let $k$ be a small constant, i.e. $k \in O(1)$. Sketch an $O(n^2 \log n)$ time algorithm that can compute the number of subsets of $P$ of size $k$ that can be covered exactly (i.e. the disk contains no additional points from $P$) by a unit disk. Two or three paragraphs of description is sufficient; you do not have to prove correctness or give the full analysis.