Requirements: Assignments have to be typeset in English. They have to be handed in by each student separately, except for exercises marked with (G), which should be solved and handed in by each (project) group. Please include your name on top of the first sheet.

How to describe algorithms: Whenever you are asked to describe an algorithm, you should present three things: the algorithm, a proof of its correctness, and a derivation of its running time.

A careful geometric proof can be quite cumbersome at times, so you are highly encouraged to use intuition and give illustrations whenever appropriate. However, beware of “proof by picture”: a poorly drawn figure might make you ignore special cases that are not apparent in your drawing. Try to keep your answers concise and to the point. Never give detailed code. Only give sufficient detail to convince us that your method and your running time analysis are complete and correct.

Unless stated otherwise, you can assume that objects are in general position, that is, no three points are collinear and no four points are cocircular. You may also assume that any geometric primitive that involves only a constant number of objects which are each of constant complexity can be computed in $O(1)$ time.

Exercise 1: [4 points] We called a quadtree balanced if two adjacent squares of the quadtree subdivision differ by no more than a factor of two in size. To save a constant factor in the number of of extra nodes needed to balance a quadtree, we could weaken the balance condition by allowing adjacent squares to differ by a factor of four by size. Can you still complete such a weakly balanced quadtree subdivision to a conforming mesh such that all angles are between $45$° and $90$° by using $O(1)$ triangles per square?

Exercise 2: [6 points] Let $P$ be a set of $n$ points in $\mathbb{R}^d$.

(a) Let $p$ be a point in $P$ and $q$ be a nearest neighbor of $p$ in $P$, i.e., $q$ is a point in $P \setminus \{p\}$ with minimal distance to $p$. Consider an arbitrary $s$-WSPD of $P$ with $s > 2$. Let $\{A, B\}$ be the pair in the decomposition with $p \in A$ and $q \in B$. Prove that $A = \{p\}$. Argue that given a WSPD you can compute the closest pair in $P$ in $O(n)$ time.

(b) Use (a) to argue that the size (i.e., the number of pairs) of an $s$-WSPD with $s > 2$ is at least $n/2$ pairs. Use the construction of a spanner based on a WSPD from the lecture to argue that for $s > 4$ the size of an $s$-WSPD is at least $n - 1$. A brief answer is sufficient.

(c) Prove or disprove: For every point set $P$ of $n$ points and every separation ratio $s$ there is a WSPD $\{A_i, B_i\}$ for $P$ with separation ratio $s$ and $\sum |A_i| + |B_i| = O(n \log n)$.

Exercise 3(G): [4 points] In this exercise you will give an algorithm for computing the $\Theta$-graph of a point set $P$ of size $n$ in the plane. See \url{http://www.win.tue.nl/~kbuchin/}.
teaching/2IL55/theta-graph.pdf for the \( \Theta \)-graph. For \( A \subset \mathbb{R}^d \) and \( p \in \mathbb{R}^d \) we denote by \( A + p \) the set \( A \) translated by \( p \), that is, \( A + p := \{ x + p \mid x \in A \} \). For instance, if \( \ell \) is a line through the origin, then \( \ell + p \) is the line through \( p \) parallel to \( \ell \).

(a) Consider the following problem first. Given a fixed (non-vertical and non-horizontal) line \( \ell \) through the origin. We want to maintain a data structure on \( P \) supporting in \( O(\log n) \) time insertion, deletion and the following operation:

\[ \text{LeftMostAbove}(p) : \text{for a query point } p \in P \text{ compute the leftmost point in } P \text{ above } \ell + p. \]

For this we maintain a balanced binary search tree storing the points \( p \in P \) at the leaves sorted by the position at which \( \ell + p \) intersects the \( x \)-axis. The internal nodes contain information to guide the search and they contain the point with minimum \( x \)-coordinate in the corresponding subtree (i.e., the "leftmost" point but not necessarily leftmost in the tree). Show how to compute \( \text{LeftMostAbove}(p) \) using this data structure in \( O(\log n) \) time.

(b) Next consider the following problem: Let \( C \) be a cone with the origin as apex and for \( p \in P \) let \( C_p := C + p \). For all \( p \in P \) we want to determine the leftmost point \( q \in P \cap C_p \) unequal to \( p \). Give a sweep algorithm for this problem running in \( O(n \log n) \) time.

Hint: You can assume that one side of \( C \) is aligned with a direction of your choice by rotating the point set. And of course you will want to use (a).

(c) For \( \Theta = 2\pi/k \) give an \( O(kn \log n) \)-time algorithm to compute the \( \Theta \)-graph of \( P \). Two or three sentences are sufficient as answer.