The following exercises will be discussed in the exercise class on October 19, 2016. Since we expect you to be working on the special assignment, all exercises are in-class.

**In-Class Exercise 1: Number of Cells in Arrangements**

We prove lemma 2.5:

Recall why an arrangement of \( n \) lines has at most \( \binom{n}{2} \) vertices and at most \( n^2 \) edges. Show that it has at most \( \frac{n(n+1)}{2} + 1 \) cells, and show that all of these bounds are attained if no three lines intersect in a common point and no two lines are parallel.

**In-Class Exercise 2: Locating a Point in a Line Arrangement**

Recall the algorithm for counting the points below a query line: We interpreted the question in the dual setting, so we were looking for the number of lines above a query point. Prove the space bound in lemma 2.6, which was left as an exercise:

\[
\sum_{v \text{ inner node}} |\hat{S}_v| \leq 2n^2
\]

where, as you may recall,

- \( n \) is the number of lines,
- \( v \) ranges over the nodes of the data structure used in the algorithm, which has one node for every level of the line arrangement;
- \( S_v \) is the set of \( x \)-coordinates of the corresponding level;
- \( \hat{S}_v \) is the ‘enhanced’ set: If a node \( v \) has no child which is an inner node, \( \hat{S}_v = S_v \). Otherwise, \( \hat{S}_v \) is obtained from \( S_v \) by adding every other value from each of the sets \( \hat{S}_u \), \( u \) a non-leaf child of \( v \).

**In-Class Exercise 3: Nearest Neighbor Changes**

We are given a set \( P \) of \( n \) points in \( \mathbb{R}^2 \) and a point \( q \) which has distinct distances to all points in \( P \). We add the points of \( P \) in random order (starting with the empty set), and observe the nearest neighbor of \( q \) in the set of points inserted so far. What is the expected number of distinct nearest neighbors that appear during the process?
In-Class Exercise 4: Linear Separability in Linear Time

Suppose we are given a set $S$ of $n$ closed halfspaces in the plane. For each $H \in S$, let $\ell_H \subset H$ denote its boundary line. We assume that the halfspaces are in general position such that no two boundary lines are parallel and no three boundary lines meet in a single point. Consider the input to be given in the form of linear inequalities, say.

In this task we are interested in a randomized algorithm to decide whether the intersection of the given halfspaces is non-empty, that is whether $R(S) = \emptyset$, for $R(S) := \bigcap_{H \in S} H$, or not. If $S$ has a non-empty intersection, we would also be interested in a certificate point, that is in a point $x \in \bigcap_{H \in S} H$ to demonstrate non-emptiness. To make your calculations simpler, we want to make certificate points unique. To this end, we assume $|S| \geq 2$ and fix, arbitrarily, two halfspaces $H_1, H_2, \in S$. The region $R(S)$ is obviously contained in a wedge formed by the lines $\ell_{H_1}$ and $\ell_{H_2}$ (see figure). Before starting any algorithm, you may assume that the input is rotated first in such a way that this wedge opens to the right and the intersection point $g \in \ell_{H_1} \cap \ell_{H_2}$ acts as a guard that no point in $R(S)$ can have a smaller $x$-coordinate than $g$ (see figure). We then define for any $S' \subseteq S$ with $H_1, H_2 \in S'$ the unique certificate point $c(S')$ as the point in $R(S')$ that has the smallest $x$-coordinate. You may assume that $H_1$ and $H_2$ are fixed before and known to all your algorithms below.

Following are your tasks:

(a) Let $|S| \geq 3$ (with $H_1$ and $H_2$ as described above) and let $H \in S \setminus \{H_1, H_2\}$ be an arbitrary one of the halfspaces. Prove: if $R(S) \neq \emptyset$, then either $c(S) = c(S \setminus \{H\})$ or $c(S) \in \ell_H$.

(b) Let $|S| \geq 3$ (with $H_1$ and $H_2$ as described above) and let $H \in S \setminus \{H_1, H_2\}$ be an arbitrary one of the halfspaces. Assume that $R(S \setminus \{H\}) \neq \emptyset$. Write down a deterministic algorithm that runs in time linear in $n = |S|$ and that on input $(S, H, c(S \setminus \{H\}))$ determines whether $R(S) \neq \emptyset$ and if so outputs $c(S)$.

(c) Let again $|S| \geq 3$ (with $H_1$ and $H_2$ as described above). Using (b), write down a randomized algorithm which, given $S$, determines whether $R(S) \neq \emptyset$ and if so outputs $c(S)$. Your algorithm should run in expected time linear in $n = |S|$.

\footnote{this rotation can always be done such that we also do not have vertical or horizontal lines, which you may assume}
In-Class Exercise 5: Minimum Cut

(a) Let $G$ be the multigraph with vertices $1, \ldots, 12$ depicted below, and let $e$ be an edge of $G$ chosen uniformly at random. Calculate the value of $\Pr[\mu(G) = \mu(G/e)]$.

(b) Let $A$ be a randomized algorithm which, when given any multigraph $G$ on $n$ vertices as input, outputs a certain (random) value $A$, where $A = \mu(G)$ with probability $1/n$ and $A < \mu(G)$ in all other cases. Let $A_1, \ldots, A_k$ be the values returned by $k$ independent invocations of $A$ with input $G$. Choose a specific value for $k$ and describe another algorithm $B$ which takes $A_1, \ldots, A_k$ as input and which outputs a value $B$ such that $B = \mu(G)$ with probability at least $1 - n^{-42}$. 