Homework 2
Due 2017-Jan-23, 11:59pm

Prof. Amit Chakrabarti
Department of Computer Science
Dartmouth College

General Instructions. Same as for Homework 1.

Honor Prinicple. Same as for Homework 1.

---

4. The *extreme points problem* asks whether the convex hull of n given points in the plane has n vertices (i.e., whether all of the n points are "extreme"); note that this is potentially an easier problem than actually computing the convex hull.

   Model this problem as a set recognition problem, i.e., that of recognizing whether or not an input vector $\mathbf{x}$ belongs to W, for an appropriate set $W \subseteq \mathbb{R}^{2n}$. Prove that the number of connected components $\#W \geq (n-1)!$ and conclude that the algebraic computation tree complexity of the problem is $\Omega(n \log n)$.

5. Let $\mathbf{a}_1, \ldots, \mathbf{a}_k$ and $\mathbf{b}$ be fixed nonzero vectors in $\mathbb{R}^n$ such that the system of inequalities

$$\langle \mathbf{a}_1, \mathbf{x} \rangle \geq 0\,,\ \langle \mathbf{a}_2, \mathbf{x} \rangle \geq 0\,,\ \ldots\,,\ \langle \mathbf{a}_k, \mathbf{x} \rangle \geq 0\,,$$

   in the unknown $\mathbf{x} \in \mathbb{R}^n$, is feasible and implies the inequality $\langle \mathbf{b}, \mathbf{x} \rangle \geq 0$. Here $\langle\ ,\ \rangle$ denotes the standard inner product in $\mathbb{R}^n$. Then it can be shown that $\mathbf{b}$ is a non-negative linear combination of the $\mathbf{a}_i$'s, i.e., $\mathbf{b} = \sum_{i=1}^{k} \lambda_i \mathbf{a}_i$ for some non-negative reals $\{\lambda_i\}$. This fact is sometimes known as Farkas's Lemma.

   Using Farkas's Lemma, prove the following two lower bounds in the *linear* decision tree model (i.e., on input $\mathbf{x}$, each internal node gets to ask a question "$\sum_{i=1}^{n} c_i x_i \geq 0$?" where the c_i's are constants).

   5.1. The complexity of finding the largest of n given reals is $n-1$.

   5.2. The complexity of finding the second largest is at least $n - 2 + \log n$.

   Hint: Once you have solved #5.1, use what you learnt along with a leaf counting argument to solve #5.2.