Exercises marked with a ⋄ can be handed in for bonus points. Due date is November 5.

**Exercise 1 (⋆)**
Show that the VC dimension of the set system consisting of all $n$-dimensional axis aligned hyperrectangles is $2^n$.

**Exercise 2**
Let $\mathcal{C}$ be a concept class of boolean formulas (e.g. 3-CNF or 3-term DNF) and let $\mathcal{C}^m$ be the restriction of $\mathcal{C}$ to those boolean formulas which do not contain negative literals. Show that if $\mathcal{C}^m$ is efficiently PAC learnable, then $\mathcal{C}$ is efficiently PAC learnable.

**Exercise 3**
Let $\mathcal{C}_n$ be the set of all algorithms $c$ taking an $n$-bit string $a$ as input and returning a single bit $c(a)$ in time $\text{size}(c)$. Show that if $\mathcal{C}$ is efficiently PAC learnable, then every (polynomial time evaluable) concept class is efficiently PAC learnable.

**Exercise 4**
A 1-decision list $L = [(l_1, b_1), \ldots, (l_k, b_k), b]$ is defined by an ordered sequence of pairs $(l_i, b_i)$, where $l_i$ is a literal over boolean variables $x_1, \ldots, x_n$ and $b_i$ is a bit, and a default bit $b$. Given an $n$-bit string $a \in \{0, 1\}^n$, the value $L(a)$ is defined to be either $b_j$ where $j$ is the smallest index such that $l_j(a) = 1$, or $b$ if no such $j$ exists.

For example, for $L = [(x_1, 1), (\neg x_4, 0), 1]$, we have $L(0001) = 1$ (using the default bit), $L(0000) = 0$ (because $x_1 = 0$ and $\neg x_4 = 1$), and $L(1000) = 1$ (because $x_1 = 1$).

1. Design an efficient Occam algorithm for the concept class of 1-decision lists.

2. Conclude that 1-decision lists are efficiently PAC learnable. How many samples are required by the PAC learning algorithm?

**Exercise 5 (⋆)**
Let $\mathcal{C}$ be efficiently PAC learnable. Show that there is an efficient probabilistic Occam algorithm for $\mathcal{C}$. (It is enough to show that the expected running time of the Occam algorithm is polynomial.)