Problem 1
(a) You are given a coin which has an unknown probability $p \in (0, 1)$ of coming up “heads” when it is tossed. Give a simple procedure for obtaining a coarse multiplicative estimate of $p$. More precisely, give a procedure and show that it has the following property: with probability at least 89%, your procedure performs $\Theta(1/p)$ tosses and outputs a value $\hat{p}$ such that

$$\frac{\hat{p}}{10} \leq p \leq 10\hat{p}.$$ 

(b) Now give a procedure which is given a parameter $\delta > 0$ and with probability at least $1 - \delta$ performs $\text{poly}(\log(1/\delta)) \cdot \frac{1}{p}$ many tosses and outputs a value $\hat{p}$ as described above.

Problem 2
In class we saw an algorithm for PAC learning monotone disjunctions which had the following property: if the algorithm is run on a target concept that is a monotone disjunction of length at most $k$, it outputs a hypothesis which is a monotone disjunction of length at most only slightly longer than $k$. In this problem you’ll show that that it is a computationally hard problem to PAC learn using a hypothesis whose length is at most exactly $k$.

More precisely, suppose that there is a PAC learning algorithm $A$ for monotone disjunctions that runs in time $\text{poly}(n, 1/\epsilon, 1/\delta)$ and has the following property: for all $k$, if $A$ is run on a monotone disjunction of length $k$, it outputs a hypothesis that is a monotone disjunction of length at most $k$.

Show that then there is a randomized $\text{poly}(n)$-time algorithm which optimally solves any instance of SET COVER with high probability. (Since SET COVER is NP-complete, this would mean that NP is contained in RP, which is viewed as being very unlikely.)

Problem 3
a) Consider the instance space $X = \{1, 2, \ldots, 999\}$. Let $C$ be a concept class consisting of 10 concepts, $c_0$ through $c_9$. A number $n$ in $X$ is an element of $c_i$ if the normal decimal representation of $n$ contains the digit $i$. So, for example, the number “778” is an element of $c_7$ and $c_8$.

What is the VC dimension of $C$? Justify your answer.

b) Now consider the instance space $X$ consisting of all words of at most 7 letters that have a dictionary entry at www.dictionary.com. Let $C$ be a concept class consisting of 26 concepts, $c_a$
through $c_z$. A word $w$ in $X$ is an element of $c_\ell$ if the letter $\ell$ is present in $w$. So, for example, the word “book” is an element of $c_b$, $c_o$ and $c_k$.

What is the VC dimension of $C$? Justify your answer.

**Problem 4**  Show that there is a domain $X$ such that for any integer $d > 0$ there is a concept class $C$ over $X$ of VC dimension $d$ such that for any $m > 0$ there is a set $S \subset X$ of $m$ points such that $|\Pi_C(S)| = \Phi_d(m)$.

**Problem 5**  As we mentioned in class, under a suitable cryptographic assumption, there is a concept class $C$ over $\{0, 1\}^n$ which has the following property: there is a PAC learning algorithm for $C$ that runs in time $\text{poly}(n, \frac{1}{\epsilon}, \frac{1}{\delta})$, but there is no online mistake-bound learning algorithm for $C$ that runs in $\text{poly}(n)$ time per update and has a $\text{poly}(n)$ mistake bound. In this problem you will show that if we do not take computation time into account but only consider sample complexity, no similar separation exists between PAC learning and online mistake-bound learning.

Let $C$ be a concept class over $\{0, 1\}^n$ that is PAC learnable using $\text{poly}(n, \frac{1}{\epsilon}, \frac{1}{\delta})$ examples. Show that $C$ is learnable in the online mistake-bound model by an algorithm that makes at most $\text{poly}(n)$ mistakes. (You may use any results proved in class, and your online learning algorithm need not be computationally efficient.)

**Problem 6**  Let $C$ be a concept class whose VC dimension is $d$, and for $s \geq 1$ denote by $C_s$ the class $C_s = \{c = c_1 \cup \ldots \cup c_s \mid c_i \in C\}$. Show that for all $s \geq 1$, the VC dimension of $C_s$ is at most $2ds \log(3s)$. (Hint: Think about the growth function $\Pi_C(m)$.)