Note: Please attempt all problems. You can ask me for hints, but please avoid the temptation to just search the internet for solutions! Working through the problems by yourself will help you understand the material better, and help prepare you for the quizzes. The problem marked by (*) is extra challenging; it is optional.

1. Prove that any deterministic communication protocol that allows Alice and Bob to check if their inputs $a_1 \ldots a_n \in \{0, 1\}^n$ and $b_1 \ldots b_n \in \{0, 1\}^n$, respectively, are equal must use at least $n$ bits of communication.

You may follow the proof structure outlined below. View the protocol of communication between Alice and Bob as a rooted binary tree, where at each node either Alice or Bob speaks by sending one bit to the other party (where what they say depends just on their input as well as the earlier communication that took place).

The protocol starts at the root of the tree, and then in each step of communication, the protocol moves either to the left child (if the sent bit was 0) or to the right child (if the sent bit was 1) of the current tree node. The leaves are labeled by the outputs of the protocol: once the protocol has reached a leaf $\ell$ labeled with a bit $b \in \{0, 1\}$, it outputs $b$. The communication cost of the protocol is just the height of this tree.

Argue that the protocol tree has the following rectangle property: for any leaf $\ell$ of the tree, if for some strings $x_1, x_2, y_1, y_2 \in \{0, 1\}^n$, we have that the protocol reaches the leaf $\ell$ when Alice has $x_1$ and Bob has $y_1$, and also the protocol reaches $\ell$ when Alice has $x_2$ and Bob has $y_2$, then the protocol must reach $\ell$ when Alice has $x_1$ and Bob has $y_2$.

Consider the pairs $(x_1, y_1), \ldots, (x_N, y_N)$ of $n$-bit strings, where $N = 2^n$, and for each $1 \leq i \leq N$, $x_i = y_i$ is the $i$th binary string of length $n$. Observe that the protocol must output 1 on each pair $(x_i, y_i)$. Use the rectangle property to argue that the protocol must have at least $2^n$ leaves, and hence have the communication cost at least $n$.

2. Show that if $\text{NP} \subseteq \text{BPP}$, then $\text{NP} = \text{RP}$. (Hint: Search-to-Decision.)

3. Show that $\text{AM} \subset \text{NP/poly}$. (Hint: Recall the proof that $\text{BPP} \subset \text{P/poly}$.)
4. Show that $\text{MA} \subseteq \Sigma_p^2 \cap \Pi_p^2$. (Hint: First, show that $\text{MA} \subseteq \Sigma_p^2$, using the ideas from the proof $\text{BPP} \subseteq \Sigma_p^2$ you saw in class. Then try to extend your proof to the case of $\text{MA} \subseteq \Pi_p^2$. Note that the inclusion $\text{MA} \subseteq \Pi_p^2$ does not immediately follow from $\text{MA} \subseteq \Sigma_p^2$, since it is not known (and in fact, not believed) that $\text{MA} = \text{coMA}$!)

5. Recall that a language $L \in \text{IP}$ if there is a probabilistic polytime verifier $V$ such that, for every input $x$, if $x \in L$, then there is a prover $P$ such that $V$, after interacting with the prover $P$ for up to a polynomial number of rounds, accepts with probability at least $2/3$, and if $x \notin L$, then every prover $P$ is rejected by $V$ with probability at least $2/3$. (Recall that the verifier $V$ uses private randomness, i.e., $V$ is not required to reveal to the prover $P$ the random bits used by $V$.)

Consider a variation on the definition of $\text{IP}$: Define the class $\text{IP}'$ where for $x \in L$, there is a prover that makes the verifier accept with probability at least $2/3$, but for $x \notin L$, every prover is rejected by the verifier with probability 1.

Show that $\text{IP}' = \text{NP}$.

6. (*) The class $\text{MIP}[m]$ is defined to have $m$ provers. That is, a language $L \in \text{MIP}[m]$ if there exists a probabilistic polytime verifier $V$ such that, for every $x \in L$, there exist $m$ provers $P_1, \ldots, P_m$ that make $V$ accept with probability at least $2/3$, and for every $x \notin L$, every collection of $m$ provers are rejected by $V$ with probability at least $2/3$. The protocol involves at most a polynomial number of rounds, and, in each round, the verifier $V$ asks $m$ questions in parallel to the $m$ provers. The provers are not allowed to communicate with each other during the run of the protocol.

Show that for any $m(n) \in \text{poly}(n)$, $\text{MIP}[m] = \text{MIP}[2]$. That is, the definition of $\text{MIP}$ doesn’t change if we allow up to a polynomial number of provers.

(HINT: Use one prover to simulate all $m$ provers, and use the second prover to simulate a randomly chosen prover from among the $m$ provers. Repeat this a few times.)