Solve four problems total, namely #1, #2, and two out of three from \{#3, #4, #5\}.

1. (MA in PP, 10 points.) Show that $\text{MA} \subseteq \text{PP}$. (You are not required to do it this way, but one way you can show this is to show $\text{MA} \subseteq \text{BPP}_{\text{path}} \subseteq \text{PP}$, where $\text{BPP}_{\text{path}}$ is the “real” name of the class called “RestartingBPP” on Homework 3.)

2. (Collapses, 10 points total.)
   (a) (4 points.) Show that $\text{PSPACE} \subseteq \text{P}/\text{poly}$ implies that $\text{PSPACE} = \Sigma_2 = \Pi_2$.
   (b) (2 points.) Show that $\text{EXP} \subseteq \text{P}/\text{poly}$ implies that $\text{EXP} = \Sigma_2 = \Pi_2$.
   (c) (4 points.) Show that $\text{NP} \subseteq \text{P}/\text{poly}$ implies $\text{AM} = \text{MA}$.

3. (Superiority II, 10 points.) Recall that for Homework #2, Problem 2, you did Exercise 3.4 in Arora–Barak. Now, show that $\text{NTIME}(n^{1.1})$ is in fact “superior” to $\text{NTIME}(n)$. You will need a new proof of the Nondeterministic Time Hierarchy Theorem. What follows are some hints for it; besides completing the proof, please note that even the hints need several details to be filled in.

   Consider a nondeterministic TM $D$ which parses its input into the form $1^i01^j0^y$, where $i, j \in \mathbb{N}$ and $y \in \{0, 1\}^\ast$. This is henceforth written as $(i, j, y)$. The number $i$ encodes an NTM $M$, the number $j$ is for “junk” (to ensure arbitrarily long strings), and $y$ is interpreted as a sequence of nondeterministic “guess bits”. If $|y| < \lceil (i + j + 2)^{1.05} \rceil$ then $D$ accepts iff $M$ accepts both $(i, j, y0)$ and $(i, j, y1)$ in $(i + j + 2 + |y|)^{1.05}$ steps. If $|y| = \lceil (i + j + 2)^{1.05} \rceil$ then $D$ accepts iff $M$ rejects $(i, j, \varepsilon)$ when using $y$ as its nondeterministic guesses.

   In solving this problem, please have a clear section where you give what you feel is/are the “main idea(s)” in the proof. Of course, you must fill in all the details in the other sections.

4. (The Exponential Time Hierarchy collapses, 10 points total.) Throughout this problem, let $N^2$ denote some nondeterministic oracle-TM running in time $kn^k$, let $A$ denote some language in $\text{NEXP}$, and let $U$ denote any fixed $\text{NEXP}$-complete language (under poly-time mapping reductions, as usual).

   (a) (3 points.) Show there is a poly-time deterministic oracle-TM $C'$ with the following property: $C'(x)$ computes the exact number of strings in $A$ of length at most $k|x|^k$.

   (b) (3 points.) Say that a nondeterministic TM $K$ “computes the answer to $y \in A$” if, on input $y$, it runs nondeterministically, accepts on at least one computation branch, and on all branches where it accepts, its working tape contains nothing but the correct answer (0/1) to the question of whether $y \in A$.

   Show that you can extend your machine $C'$ from part (a) so that after $C'(x)$ is finished, it can deterministically construct a nondeterministic TM $K$ which, on input $y$ of length at most $k|x|^k$, runs in time at most $\exp(k'|x|^k')$ (for some constant $k'$) and “computes the answer to $y \in A$.”
(c) (3 points.) Show that $C^U$ can be further extended so that $C^U(x)$ constructs the description of a nondeterministic machine $N^*$ (running in some $\exp(k'|x|^k')$ time) which, on input $x$, has the same overall answer as $N^A(x)$.

(d) (1 point.) Show that $\mathsf{NP}^\mathsf{NEXP} = \mathsf{P}^\mathsf{NEXP}$.

5. (Trading error for advice.) Show that $\mathsf{RSPACE}(n) \subseteq \mathsf{ZPSPACE}(n)/(n+1)$.

(Now to explain carefully the meaning of these complexity classes. First of all, roughly speaking, $\mathsf{RSPACE}(n)$ is to linear space as RP is to polynomial time; similarly $\mathsf{ZPSPACE}(n)$ and ZPP. However, it turns out there is a major subtlety in defining randomized space classes. The issue is whether you require the randomized machines to always halt or just to halt with probability 1. These are actually not the same; a randomized machine that flips coins until it gets a 0 and then halts has the property that it “halts with probability 1”, but it doesn’t “always halt”. It turns out that the distinction is unimportant for time-bounded classes, but quite important for space-bounded classes. To make a long story short, the “better” definition turns out to be the one where you require machines to always halt, meaning that for every input and every possible sequence of random bits they might flip, they halt in finitely many steps. In fact, once you decide on this definition, it is not too hard to show that for space-$s(n)$ machines you can assume that the machine always halts in at most $2^{O(s(n))}$ steps. (This is pleasant, because it’s something we rely on in the deterministic case, too.) Therefore, finally: We say $L \in \mathsf{RSPACE}(s(n))$ if there is a randomized Turing Machine $M$ that, on every input $x$ of length $n$ and every possible sequence of random bits, uses at most $O(s(n))$ space and at most $2^{O(s(n))}$ time, and has the following properties:

$$x \in L \implies \Pr[M(x) \text{ acc.}] \geq 2/3, \quad x \notin L \implies \Pr[M(x) \text{ acc.}] = 0.$$ 

By the way, the most famous example of this kind of class is RL, randomized log-space. With our definition, this class includes the demand that the algorithm runs in polynomial time. If this were eliminated, and the machine were only required to halt with probability 1, then the resulting class would actually equal NL! This is not too hard to prove, and is entertaining to think about.

Next, as for $\mathsf{ZPSPACE}(n)$, please use the following definition: a $\mathsf{ZPSPACE}(n)$ machine is a randomized $O(n)$-space, $2^{O(n)}$-time machine that halts on every input and every sequence of random bits, and has three kinds of final states: “accept”, “reject”, and “?”). We say that $L \in \mathsf{ZPSPACE}(n)$ if such a machine has the following property: on every input $x$, the machine never outputs a “wrong” answer (i.e., accepts when $x \notin L$, or rejects when $x \in L$); and, on every input $x$, the probability the machine outputs “?” is at most $1/3$.

Finally, $\mathsf{ZPSPACE}(n)/(n+1)$ is the same class, but where the machine takes $n+1$ bits of advice on inputs of length $n$. That is, $L \in \mathsf{ZPSPACE}(n)/(n+1)$ if there exists a $\mathsf{ZPSPACE}(n)$ machine $M$ and sequence of advice strings $(a_n)$ with $|a_n| = n+1$ such that, when provided with $a_{|x|}$ on input $x$, the machine $M$ has the aforementioned accept/reject/? properties.)