(1) Show that the language of Turing machines $M_i$ such that $M_i$ is total and $L(M_i)$ is infinite belongs to $\Pi_2$ by writing a $\Pi_2$ definition for it. You may regard any decidable predicate as “ground” requiring no further quantifiers. (The only ones you need here are the Kleene $T$-predicate $T(i, x, c) \equiv c$ is a valid halting computation of $M_i$ on input $x$, a relation $U(c)$ saying $c$ accepts, and basic numerical comparisons.)

Now how about the language of $i$ such that $M_i$ is total and $L(M_i)$ is finite? Can you prove it is neither in $\Pi_2$ nor in $\Sigma_2$? (One hint: Consider designing a machine that accepts $\langle x, m \rangle$ if and only if $M_i$ accepts $x$ in exactly $m$ steps. $12 + 24 = 36$ pts.)

(2) Give a $\Sigma_3^0$ definition for the language $I_{REC} = \{ i : L(M_i) \text{ is decidable} \}$. Note that $M_i$ itself need not be total. Take it as a fact—we won’t ask for a proof—that $\Sigma_3^0$ is optimal.

Now suppose $F$ is a sound and effective formal system, meaning it has a decidable proof predicate $P(\phi, \pi) \equiv \pi$ is a proof of the sentence $\phi$ in $F$, such that whenever it holds, $\phi$ is actually true. For any TM $M_i$, we can make a sentence $\phi = \tau(i)$ from the formula $\tau(i) = (\forall x)(\exists c)T(i, x, c)$. Then when $P(\tau(i), \pi)$ holds, we call $M_i$ a provably total machine. Use the fact about $\Sigma_3^0$ and $I_{REC}$ to deduce that there are decidable languages that are not accepted by any provably total machine. ($12 + 12 = 24$ pts.)

(3) Define the “anti-index set” of any subclass $C$ of the decidable languages by $R_C = \{ i : \tau(i) \land L(M_i) \notin C \}$. That is, $R_C$ is the set of total machines that accept languages outside the class. Show that for $C = P$, $R_C$ belongs to $\Pi_2$. How about for $C = NP$? Now try it for $C = BPP$ where you don’t have a simple recursive enumeration of total machines representing the class, but... ($30$ pts. total)

(4) Prove that a language $A$ belongs to $NP \cap co-NP$ if and only if $NP^A = NP$. ($24$ pts.)

(5) Define $H$ to be the intersection of $P^A$ over all oracles $A$ such that $NP^A = P^A$. Show the following facts about $H$:

(a) $PH \subseteq H \subseteq PSPACE$.

(b) If $PH = \Sigma_k^P = \Pi_k^P$ for some $k$ (read as saying that the polynomial hierarchy “collapses” to the $k$th level), then $H = \Sigma_k^P = \Pi_k^P$ too.

(c) If $R_H \notin \Pi_2$ then the polynomial hierarchy is infinite and different from $PSPACE$ and $H$ is properly between $PH$ and $PSPACE$.

(d) If $H = PH$, does that prevent the polynomial hierarchy from being infinite? (Actually, I don’t know...)

I invented this idea long ago and used “$H$” to mean a kind of disembodied hierarchy. Except that maybe $H = PH$ can be proved without major consequences, the idea hasn’t gone anywhere except as an illustration of grasping and playing with both the logical and complexity concepts. ($12 + 9 + 9 = 30$ pts. total, possibly more if you do something with (d), for $144$ pts. on the set)