Problem 1 Solution

There's a few ways to get this (even just pumping it through a script) but one way is to realize $\mathbb{Z}_{77} \cong \not\Vdash \times \not\Vdash \not\Vdash$. So an element of $\mathbb{Z}_{77}$ is a quadratic residue if and only if it is one for both $\mathbb{Z}_7$ and $\mathbb{Z}_{11}$. Furthermore, the number of roots it will have is the number of roots it has in $\mathbb{Z}_7$ times the number of roots it has in $\mathbb{Z}_{11}$. So $(a, b) \in \mathbb{Z}_7 \times \mathbb{Z}_{11}$ is a quadratic residue if and only if a or b (but not both) is 0 and the other is quadratic residue in its group.

So only multiples of 7 or 11 can be a quadratic residue with exactly two roots. Consider squaring the following multiples of 7: $7^2 = 49,\ 14^2 = 42,\ 21^2 = 56,\ 28^2 = 14,\ 35^2 = 70$. The remaining multiples of 7 in $\mathbb{Z}_{77}$ are just the negatives of what we've already seen and so would repeat the same squares.

Similarly, the following multiples of 11 squared fill in the remaining possible squares: $11^2 = 44,\ 22^2 = 22,\ 33^2 = 11$. And so 49, 42, 56, 14, 70, 44, 22, and 11 comprise all eight quadratic residues with exactly two roots in $\mathbb{Z}_{77}$.

And these don't affect our scheme since our scheme only considers elements of $\mathbb{Z}_{77}^*$ which these are not.

Problem 2 Solution

a) $IP \subseteq IP'$ is trivial since a deterministic prover is still considered a probabilistic one.

To show $IP' \subseteq IP$, consider $L \in IP'$. This means there exists a prover/verifier pair (P', V') that is complete and sound with respect to L. We want to create (P, V) as an IP scheme for L with P being deterministic now. Define P to behave exactly as P' excet hard-coding value in where P' would otherwise flip coins. Letting $V' = V$ we see that this scheme is

- Complete since (P', V') was complete, meaning V' accepted with probability exactly 1 when interacting with P' on valid $x \in L$. That is, any coin flip P' could make would *always* convince $V' = V$ when $x \in L$ and so hard-coding one of those value into P still maintains that completeness
- Sound since soundness is simply a measure of how resilient a verifier is against cheating provers and since the verifier is the same, it is just as resilient and sound as before. That is, since we are quantifying over the same cheating provers as we did for (P', V') to be sound in IP' (actually we're quantifying over even less provers since now they're all deterministic which is just a special case of probabilistic), then V being the same as V' doesn't accept any more instances than it did before and so the soundness bound holds.

b) NP$\subseteq IP'$ is trivial since a language being in NP is to have a polynomially verifiable certificate (proof) that can be found by a nondeterministic Turing machine and the prover P can be defined as that Turing machine in which it just shows the easily verifiable certificate to the NP verifier V. This corresponds to a trivial single round interactive proof with no interaction. (IP generalizes NP)

To show $IP' \subseteq$NP, consider $L \in IP'$. This means there exists a prover/verifier pair (P', V') that is complete and perfectly sound with respect to L. To show L is NP, we must find a polynomially verifiable certificate for every $x \in L$. Consider the polynomial size certificate of messages sent back and forth between P' and V' on input x along with the random bits V' used in that given interaction. Since V' completes this interaction in polynomial time, we can simulate V' and verify that this transcript encodes a valid interaction between P' and V'. If V' accepts in this interaction, we say that this transcript is a witness for x being in L.

Note that if $x \in L$, then there is such a transcript that makes V' accept since (P', V') is complete. That is, there is a witness for $x \in L$. However, if $x \notin L$, then it is impossible for any prover to make V' accept since our perfect soundness condition says that the probability of acceptance is absolutely 0 when $x \notin L$. So a witness can't exist since a witness is a valid transcript of V' accepting on input x. Therefore, $x \in L \leftrightarrow$ there exists a polynomially verifiable witness and so, by definition, $L \in$NP.

Problem 3 Solution

- **Completeness:** If $x \in QNR(N)$, then $\forall s \in \mathbb{Z}_N^*$, $s^2 x \in QNR(N)$. That is, if $b = 0$ then $m = s^2 x \in QNR(N)$ while if $b = 1$ then $m = s^2$ certainly is a quadratic residue and so $m \notin QNR(N)$. Since the prover can easily tell if $m \in QNR(N)$, it can distinguish between the cases of what b was and can set b' accordingly per it's protocol. Thus, b' will always equal b and the verifier will always accept.

- **Soundness:** If $x \notin QNR(N)$, i.e. $x \in QR(N)$, then $\forall s \in \mathbb{Z}_N^*$, $m = s^2 x \in QR(N)$. And since $m = s^2$ is also in $QR(N)$, whether $b = 0$ or 1 m will always be $QR(N)$. Moreover, since s is a random element of $\mathbb{Z}_N^*$, m is always just a random element of $QR(N)$. Thus, what the prover receives, m, is identically distributed whether $b = 0$ or 1 and so the prover can't distinguish which case it was sent even if it cheats and even with unlimited computational power. So the prover can't do better than guessing b' for which case they were given, giving them a probability of $1/2$ of fooling the verifier.

- **Honest Verifier Zero Knowledge:** The view of the verifier in this protocol if it is followed as described, is (m, b'). It is the case that m is either a random element of $QNR(N)$ or $QR(N)$ (with 50/50 chance of being in either) and b' is a bit equal to 0 or 1 accurately indicating which of those sets it's in, respectively. We want to create a simulator that will efficiently sample for this distribution. Define the simulator as follows. Sample a b' as a random bit and then set m accordingly to be either $s^2 x$ or s^2 for a randomly sampled $s \in \mathbb{Z}_N^*$ as the usual scheme does. Now we have an m that is either a random element of $QNR(N)$ or $QR(N)$ (with 50/50 chance of being in either) and b' is a bit equal to 0 or 1 accurately indicating which of those sets it's in, respectively. This distribution is identical to that or the honest verifier's view and so our simulator (which is certainly polynomial in time) satisfies the conditions for honest verifier zero knowledge.

  Finally, unless the quadratic residuosity problem is in polynomial time, the scheme is not perfect zero knowledge. Equivalently, we want to show that if the scheme is perfect zero knowledge, then there is a polynomial time algorithm for the QR problem.

  So let's assume that our scheme is perfect zero knowledge. This means, by definition, that every possible (cheating) verifier has a polynomial time simulator that can simulate that verifier's view. Specifically, let's consider the verifier that deviate's from the protocol by always sending $m = x$. The view of this specific verifier is x along with b' which indicates whether or not $x \in QR(N)$ (and accurately since the prover has p, q and will answer accordingly based on its protocol). Since our scheme is perfect zero knowledge, we must have a simulator that recreate the view without the prover's help. That is, it will take in x, N and return x and b'. However, since it simulates the verifier's view, this b' will indicate whether or not x is a quadratic residue. But then this simulator solves the QR problem in polynomial time.