- You know the drill now: Find students around you to form a ***small group***; use ***all resources*** to help to solve the problems; ***discuss*** your idea with other group member and ***write down*** your own solutions; raise your hand and pull the ***course staffs*** to help.
- This worksheet is *optional*; ***no submissions required***, although we encourage to try it out and test your understanding.

Our topic for this working session is on *zero-knowledge proofs*.

Recall an ***interactive proof*** is between a *prover* and a *verifier*.

- The ***prover P*** is all-mighty, with unlimited computational power. The prover only has access to the input at the start.
- The ***verifier V*** is a randomized polynomial-time Turing machine, with access to a private coin and the input at the start.

Now (P, V) performs a ***k-round interaction*** on input x:

- The verifier, after tossing the coin, sends a message m_1 to P. Formally, $m_1 := V(x, r)$, where x is the input and r is a sequence of random bits. Notice that m_1 has to be computed in polynomial-time.
- The prover, on receving the message m_1 from V, device a response m_2 based on x and m_1. Formally, $m_2 := V(x, r; m_1)$. Remember that there is *no restriction* for m_2 to be computed in polynomial-time, as P is all-mighty.
- Now repeatedly, the verifier will send a message m_i to P using the input x, the random string r, and partial message history $m_1, \dots, m_{i-1}$; and the prover will respond with another messgae m_{i+1} to V using the input x and partial message history $m_1, \dots, m_i$.
- After k messages sent, the verifier V has to decide whether to accept or reject, based on x, r, and the whole message history.

We say that a language L is in ***IP*** (the class of interactive proofs) if there exists some BPP machine V, such that for every input x of length n, there is a polynomial bound k in n on the number of rounds, where

- *[Completeness]* If $x \in L$, then there is a prover P that convinces V to accept using a k-round interaction with at least $2/3$ chance.
- *[Soundness]* If $x \notin L$, then every prover $\hat{P}$ can only convince V to accept using a k-round interaction with at most $1/3$ chance.

By repeating the interactive protocol we can safely boost the success rate to $1 - 1/\operatorname{poly} n$. (Why?)

Now, a ***zero-knowledge proof*** for an NP-problem L is an interactive proof (P, V) that has the extra requirement that no malicious verifier $\hat{V}$ can learn anything more than:

- *[Zero-knowledge]* For every verifier $\hat{V}$ and every k-round interaction between P and $\hat{V}$, there exists a BPP maichine S (simulator for P), such that for every $x \in L$ and certificate u (recall L is in NP), the output of $\hat{V}$ after the interaction is *computational indistinguishable* from running S standalone without access to P. In other words, malicious $\hat{V}$ could have generated the "same" response without talking to P.

That was a mouthful. Let's instead play with some hands-on examples.

Luke and Lucy have some leftover Halloween candies (yes, they haven't finished them because their Dad hid a good portion of the candies). They want to compare the number of candies they have, but don't want to reveal the actual numbers (because they don't want to share). We can safely assume that the number of candies is no more than 100.

Can you help them to decide *if they have the same number of candies*? The method you describe has to be convincing, as in both Luke and Lucy will believe the end result beyond reasonable doubt. Finally, we have to assume that the kids are perfectly reasonable.Did I mention that Luke and Lucy are five?

1. Supposedly you count both numbers and tell them they are different. But why would they trust you? Come up with a protocol so that Luke and Lucy will be convinced.

2. Supposedly you count both numbers and tell them they are *the same*. Again why would they trust you? Come up with a protocol so that Luke and Lucy will be convinced. *[Hint: Go read up the zero-knowledge proof for graph isomorphism, and adapt to this setting.]*

3. Now there are no adults around. How can Luke and Lucy check if the number of candies are the same between themselves? *[Hint: You may assume certain crypographic primitives.]*

To think about later: (No submissions needed)

4. Decide if they have any candy in common.

5. Decide who has the more candies. *[Hint: Can you use the method for testing intersection?]*

Conceptual question: In class we have shown that the NP-complete Hamiltonian cycle problem indeed has a zero-knowledge proof. Does this imply that all NP-complete problems have zero-knowledge proofs?