Exercises Zero Knowledge (1)

1 Interactive Proof System

Recall the example where Eve proved to Wall-E that she can tell the difference between Pepsi and Coca Cola. She did this by having Wall-E pour her some cola, without her knowing which brand it was. Then she would tell him whether it was Pepsi or Coca Cola.

1. Explain why this protocol could be implemented as an interactive proof system by arguing why the completeness and soundness bounds are not exceeded.
   Hint: By repetition of this protocol, the probabilities of making a mistake might change.
2. Is this a zero-knowledge proof?

2 Where's Waldo

In the lecture we discussed a zero-knowledge way of showing you found Waldo. We did this by copying the page, going into another room and cutting out Waldo.

1. How can someone, who does not know where Waldo is, and does not have access to the prover, simulate the interaction between the prover and this defined verifier? I.e. How would the simulator work for this proof, where the verifier only answers 'yes' if he is convinced, and 'no' if he isn't?
2. Can you think of another way of showing that you found Waldo without revealing his whereabouts?

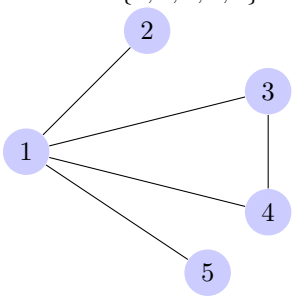
3 Simulators

Why do simulators have to be polynomial time?.

4 Graph Nonisomorphism

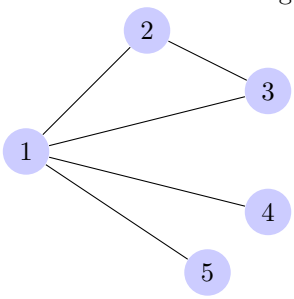
A graph is a mathematical structure that consists of a set of vertices (V) and a set of edges (E). Edges always connect two vertices. An example of a graph

with $V = \{1, 2, 3, 4, 5\}$ and $E = \{(1, 2), (1, 3), (1, 4), (1, 5), (3, 4)\}$ looks like this:



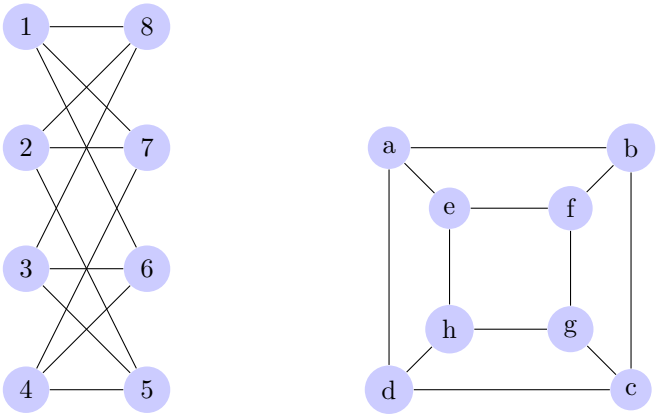
If we now, after this see another graph, we may wonder whether the two graphs are in fact the same up to relabeling. This means that there should be a function φ that will map all vertices of the one graph to those of the other such that the edges match.

So consider the following graph:



It is quite clear that if we define φ as $\varphi(1) = 1, \varphi(2) = 4, \varphi(3) = 3, \varphi(4) = 2, \varphi(5) = 5$, we end up with virtually the same graph. The existence of such a function is called **graph isomorphism**.

In this exercise we will work with a protocol for proving graph nonisomorphism: proving that such a function does not exist. This is fairly clear in some cases, like the one above, but in other cases it is quite difficult to see whether or not two graphs are isomorphic. For example, these two:



An interactive proof system for Graph Nonisomorphim

We will now define a protocol for an interactive proof system for graph nonisomorphism.

As always, we have two interactive Turing machines, the prover (from here on

called P) and the verifier (from here on called V). The common input they receive are two graphs, G_1 and G_2, which are the subject of the proof. The fact that they are nonisomorphic is only known to P, not to V. It is now up to P to prove that they are nonisomorphic.

The Protocol

1: V randomly chooses G_1 or G_2. Then he computes a random function ψ that 'shuffles' the vertices as we did above. In other words, he creates a random isomorphic copy of the graph he chose. Let's call that one H. He sends H to P.

2: P receives H and determines whether it is isomorphic to G_1 or G_2, and sends 1 (2) to V.

3: If the number V received from P is corresponds to the number of the graph V originally chose, V accepts. Otherwise he rejects.

The Exercise Show that the protocol is an interactive proof system. I.e. show the following:

1. The Verifier is polynomial time
2. The completeness bound is met
3. The soundness bound is not exceeded.