1 3-Coloring

Protocol: to be repeated $|E|k$ times or until rejection

1. Prover creates and commits to a random permutation of a 3-coloring of $G$ using the colors $\{1,2,3\}$.

2. Verifier randomly selects an edge $(u,v)$ of $G$.

3. Prover reveals the colors of $u$ and $v$.

4. Verifier accepts iff the colors are in $\{1,2,3\}$ and distinct.

Zero-Knowledge: At each round, an honest Prover shows the Verifier two random different colors. Clearly, the Verifier could simulate this by himself, by just picking two colors at random.

Completeness: Obviously the protocol is complete: if the Prover knows a 3-coloring then he can always pass the above test.

Partial Soundness: If the Prover does not actually know a good coloring, then the coloring he commits to must be wrong in at least one edge. Let $m$ be the number of edges in the graph; the Verifier will choose a mis-colored edge at least $1/m$ of the time. Thus, after $km$ rounds, a dishonest Prover will have successfully fooled the Verifier with probability at most $(1 - 1/m)^{mk} \leq 1/e^k$

References: None

2 Edge Coloring

Protocol: to be repeated $|V|k$ times or until rejection

1. Prover creates and commits to a random permutation of the colors of a $d$-coloring of $G$ using the colors $\{1,\ldots,d\}$.

2. Verifier randomly selects a vertex $v$ of $G$.

3. Prover reveals the colors of all incident edges of $v$.

4. Verifier accepts iff the colors are in $\{1,\ldots,d\}$ and distinct.
Zero-Knowledge: At each round, an honest Prover shows the Verifier some random different colors. Clearly, the Verifier could simulate this by himself as in the previous problem.

Completeness: Obviously the protocol is complete.

Partial Soundness: If the Prover does not actually know a good coloring, then the coloring he commits to must be wrong in the neighborhood of at least one vertex. Let \( n \) be the number of vertices in the graph; the Verifier will choose a bad vertex at least \( 1/n \) of the time. Thus, after \( kn \) rounds, a dishonest Prover will have successfully fooled the Verifier with probability at most \((1 - 1/n)^{nk} \leq 1/e^k\)

References: ashen

3 Vertex Cover

Protocol: to be repeated \( k \) times or until rejection.

1. Let \( S \) be the original vertex cover. Prover creates and commits to a random permutation \( \pi \) on the vertices of the graph \( G \), a copy of \( \pi(G) \), and \( S' = \{ \pi(x) | x \in S \} \), the permuted vertex cover.

2. Verifier randomly selects to either view the entire graph with the permutation \([\text{accepts iff the graph is } G]\), or the possible edges between pairs of vertices not in \( S' \) \([\text{accepts iff there are no edges}]\).

Zero-Knowledge: At each round, an honest Prover either shows the Verifier a random permutation of \( G \), or a bunch of non-edges between randomly named vertices. Clearly, the Verifier could simulate this by himself.

Completeness: Obviously the protocol is complete.

Partial Soundness: If \( G \) is correct and there are no edges between vertices not in \( S' \), then all edges in \( G \) must have at least one endpoint in \( S' \), so \( S' \) is a vertex cover for the permuted graph, and \( S \) is a vertex cover for the original graph. Thus, if the Prover does not actually know a good coloring, he can only satisfy one of the options at the time. Thus, in each round the Verifier will catch a dishonest Prover at least \( 1/2 \) of the time. Thus, after \( k \) rounds, a dishonest Prover will have successfully fooled the Verifier with probability at most \( 1/2^k \).

References: agavlova