Vladimir Hugec

Question 1:

1.1 A

A polynomial time algorithm A for $COLOR_1$ operates as folllows.

A = "On input $\langle G \rangle$ where G is a graph that can be colored with one color:
1. Select an arbitrary node x to begin and color it one color.
2. For each neighbor of x, $y_1, y_2, ...y_n$ color them the same color as x.
3. Repeat step 2. for each neighbor of $y_1, y_2, ..., y_n$.
4. Continue until all nodes in G are colored. "

NOTE: This is just a BFS

1.2 B

A polynomial time algorithm B for $COLOR_2$ operates as folllows.

B = "On input $\langle G \rangle$ where G is a graph that can be colored with two colors:
1. Select an arbitrary node x to begin and color it one color.
2. For each neighbor of x, $y_1, y_2, ...y_n$ color them with the other color.
3. Repeat step 2. for each neighbor of $y_1, y_2, ..., y_n$, coloring them the opposite color as their predecessors.
4. Continue until all nodes in G are colored. "
5. Verify that all for nodes that are connected by an edge, then those vertices are assigned different colors. If test passes then Graph is 2-colorable.

NOTE: As above, just a BFS

Question 2:

To show that $COLOR_k$ is polynomial time reducible to $COLOR_{k+1}$, assume we have a graph G that is k-colored. Now we create a new node of an arbitrary color that is not present in G. Now suppose we duplicate G as G' with the caveat that the new node we created will be attached to every other original node in G. This way we know that since G is k-colored, then G' must be $k+1$-colored. And similarly in reverse, if we know G' is $k+1$-colored, since we node our arbitrarily colored node is not present in G but G is exactly the same otherwise to G', we know G must be k-colorable. And so $COLOR_k \leq_P COLOR_{k+1}$.

Since we know that $COLOR_1 \in P$, $COLOR_2 \in P$ and $COLOR_3$ is NP-complete. $\forall k \geq 3$ we can reduce $COLOR_k$ to $COLOR_{k+1}$ and so we know that $\forall k \geq 3$, $COLOR_k$ must be NP-complete.

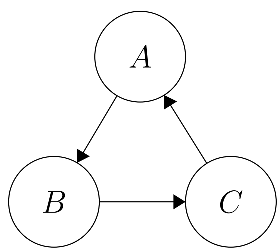
Question 3:

3.1 A

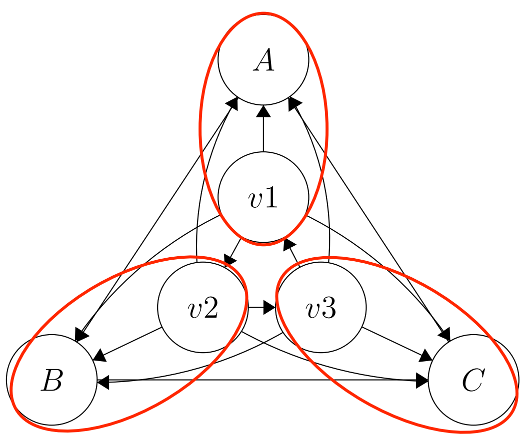
The certificate would be the 3 sets of vertices.

3.2 B

To show that $G \in COLOR_3 \iff G' \in FOREST_3$. Assume that G is 3-colorable. We know that since G is 3-colorable, a particular node can only be connected to a maximum of 2 other nodes. Any more than that and the graph is not 3 colorable. Assume we have a simple graph, G, that is 3-colorable as shown below:



Now, adding in v_1, v_2, v_3 and connecting it to every other node in the graph including themselves:



We can see that the set of 3 subgraphs is $\{A, v_1\}$, $\{B, v_2\}$, $\{C, v_3\}$. Adding in more nodes to G, we notice that it is always possible to split the new G'

into 3 subgraphs. If we add node D connected to each A, B, the subgraphs would become $\{A, v_1, C\}$, $\{B, v_1, v_2\}$, $\{D\}$. If we add node E connected to each C, D, the subgraphs would become $\{A, v_1, C\}$, $\{B, v_1, v_2\}$, $\{D, E\}$. The pattern continues. And so we know that if G is 3 colorable then $G' \in FOREST_3$. Similarly in reverse, since we know $G' \in FOREST_3$, we know that v_1, v_2, v_3 are the only nodes that violate the connection to a max of two other nodes rule. So if those nodes are removed then it must be possible to 3-color the resultant graph.