## Solution to Question 1

By definition, $|C_p| \leq D - 1$ and $|C_q| \leq D - 1$. Thus, $|S| \geq (2D - 1) - (D - 1) - (D - 1) = 1$.

## Solution to Question 2

*Liveness proof:*

Let $p$ be a node that does not satisfy property 1. Then, as p is eventually activated, $p$ executes part 1 of the algorithm and attributes distinct colors to its edges. Thus, $p$ satisfies property 1.

*Safety proof:*

Let $p$ be a node satisfying property 1. Let us show that $p$ always satisfies property 1. Suppose the opposite: $p$ satisfies property 1, and after some activation of $p$, $p$ does not satisfy property 1 anymore.

During this activation, as $p$ initially satifies property 1, $p$ only executes part 2 of the algorithm.

- If $p$ executes 2.(a), then $C(p, q) := C(q, p)$ with $C(q, p) \notin C_p$, and property 1 is still satisfied: contradiction. Thus, the result.
- If $p$ executes 2.(b), then $C(p, q) := C$ with $C \in \{1, \ldots, 2D - 1\} - C_p - C_q$. Thus, $C \notin C_p$, and property 1 is still satisfied: contradiction. Thus, the result.

## Solution to Question 3

*Liveness proof:*

Let $\{p, q\}$ be an edge that does not satisfy property 2. Then, $p$ is eventually activated and executes part 2 of the algorithm (it does not execute part 1 because all nodes satisfy property 1). If $C(q, p) \notin C_p$, then we have $C(p, q) = C(q, p)$, and the property is satisfied.

Otherwise, if $C(q, p) \in C_p$, then $C(p, q) := C$ with $C \in \{1, \ldots, 2D - 1\} - C_p - C_q$. Then, when $q$ is eventually activated, we have $C(p, q) = C \notin C_q$. Thus, according to part 2.(a) of the algorithm, we have $C(p, q) = C(q, p)$, and the property is satisfied.

*Safety proof:*

Let $\{p, q\}$ be an edge satisfying property 2. Let us show that $\{p, q\}$ always satisfies property 2. Suppose the opposite: $\{p, q\}$ satisfies property 2, and after some activation of either $p$ or $q$, $\{p, q\}$ does not satisfy property 2 anymore.

Suppose that this corresponds to an activation of $p$ (thus, $q$ is not activated at the same time). Then, $p$ does not execute part 1 of the algorithm because all nodes satisfy property 1. Thus, $p$ executes part 2 of the algorithm. However, as $C(p, q) = C(q, p)$, $p$ does not modify $C(p, q)$. Thus, we still have $C(p, q) = C(q, p)$, and property 2 is still satisfied: contradiction.

Same reasoning if this corresponds to an activation of $q$.