Principles of Distributed Computing
Exercise 5: Sample Solution

1 Coloring Rings

a) Let $n \geq 4$ be even, and let R_n be the set of all labeled rings on n vertices (there are $(n-1)!/2$ of those); note that we need to consider all these graphs since a correct algorithm has to produce a valid coloring for each of them. Consider the r-neighborhood graph $\mathcal{N}_r(R_n)$ of R_n. Note that for $r = n/2-2$ the r-neighborhood of a node contains all but three identifiers, ordered according to their occurrence.

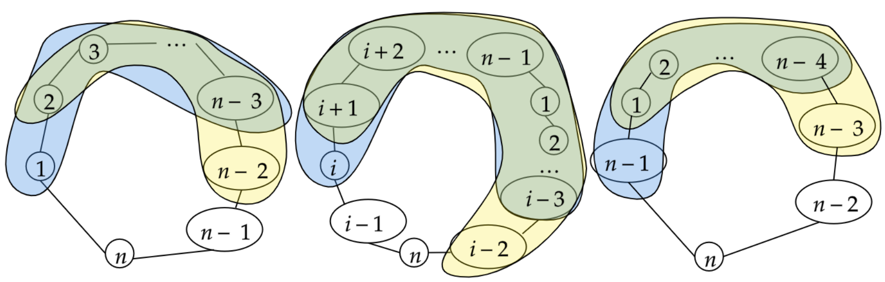
There exists a correct algorithm to legally color an n-vertex ring with two colors in r rounds if and only if $\mathcal{N}_r(R_n)$ is bipartite, i.e., the r-neighborhood graph contains no odd cycle. However, the following r-neighborhoods constitute a cycle of length $n-1$, where "…" signifies labels in ascending order:

$$(1, \ldots, n-3), (2, \ldots, n-2), (3, \ldots, n-1), (4, \ldots, n-1, 1), (5, \ldots, n-1, 1, 2), \ldots,$$
$$(n-1, 1, 2 \ldots, n-4), (1, \ldots, n-3)$$

This means that at least two of these adjacent r-neighborhoods would receive the same color from a deterministic algorithm $\mathcal{A}$, and any ring in R_n containing them adjacently would not be colored properly.

Note that, although we defined an odd cycle, we are not done: roughly, we still need to show that these r-neighborhoods are *valid*. We use the notation N_i to denote the r-neighborhood above that starts with label i (i.e. $N_4 = (4, \ldots, n-1, 1)$). Concretely, we define $N_i = (i, i+1, \ldots, n-4+i)$ for $i \leq 3$, and, for $i > 3$, we define $N_i = (i, \ldots, n-1, 1, \ldots, i-3)$. We need to show that there is a family of n-vertex rings containing:

- for every $i < n$, a ring $R_{i,i+1}$ where two adjacent nodes have N_i and N_{i+1} as their r-neighborhoods.
- a ring $R_{n-1,1}$ where two adjacent nodes have N_{n-1} and N_1 as adjacent neighborhoods.



For $i = 1$ and $i = 2$, $R_{i,i+1}$ is simply the n-vertex ring having labels in ascending order: $(1, 2, \ldots, n)$. For $i \geq 3$, we intuitively have to move the label n *outside* the r-neighborhoods N_i and N_{i+1} (or N_1, if $i = n-1$). Hence, for $3 \leq i < n-1$, $R_{i,i+1}$ is the n-vertex rings having labels $(i, \ldots, n-1, 1, \ldots i-3, (i+1)-3) = i-2, n, i-1)$. Finally, $R_{n-1,1}$ is the

n-vertex ring having labels $(n-1, 1, 2 \ldots, n-4, n-3, n, n-2)$. The rings are shown in the figure above.

Since $\mathcal{N}_r(R_n)$ has such an odd cycle for all $r \leq n/2 - 2$, there exists no algorithm that correctly colors every even ring of length n with 2 colors in at most $n/2 - 2$ rounds.

b) Each node informs its two neighbors whether it is in the MIS or not and additionally sends its identifier. If node v is in the MIS, it sets its color to 1. If v is not in the MIS but both of its neighbors are, then v sets its color to 2. If v has a neighbor w not in the MIS, v chooses color 2 if its identifier is larger than w's identifier, otherwise v chooses the color 3.

The algorithm only needs one communication round. Correctness follows from the fact that either a node v is in the MIS or at least one of its neighbors is. Thus, a MIS can at best be computed one round faster than a 3-coloring, which implies that computing a MIS costs at least $(\log^* n)/2 - 2$ rounds.

2 Ramsey theory

Let us fix the edge-color blue for knowing each other and the edge-color red for not knowing each other.

a) Figure 1 shows a valid edge-coloring for K_5.

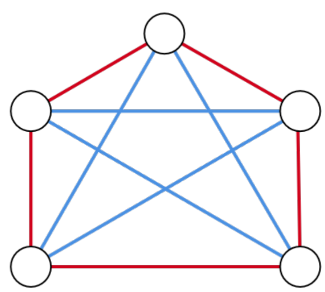


Figure 1: Valid edge-coloring for K_5.

Assume that there is a valid edge-coloring for K_6, and choose some node v. Out of the five edges incident to v, at least three are assigned the same color, and we may assume without loss of generality that this is red. Then, let u_1, u_2, u_3 be three nodes such that the edges (v, u_1), (v, u_2), and (v, u_3) are red. As the triangle induced by (v, u_1, u_2) cannot be red, the edge (u_1, u_2) must be blue. Similarly, the edges (u_2, u_3) and (u_1, u_3) must be blue. However, this means that the triangle induced by (u_1, u_2, u_3) is blue, which contradicts that the edge-coloring is valid.

b) This is a trick question: for any n, there is an edge-coloring on K_n satisfying the given constraints. Namely, just color all the edges in red. This way, any triangle of K_n has at least two red edges.

c) We first show that any edge-coloring satisfying our constraints contains at most $\lfloor n/2 \rfloor$ blue edges. Hence, assume an edge-coloring on K_n satisfying the given constraints. Consider the blue subgraph (obtained by removing all red edges). Note that this subgraph does not contain any path of length larger than 1, i.e., it is a collection of isolated edges and vertices. This can be proven by contradiction: if the blue subgraph contains a path of length two, then our colored K_n contains a triangle with two blue edges. Then, the maximum number of blue edges is at most the size of a maximum matching in K_n, which has size $\lfloor n/2 \rfloor$.

We still need to show that our upper bound is tight, i.e., that a coloring with $\lfloor n/2 \rfloor$ blue edges exists. Hence, we choose a maximum matching M of K_n, we color the $\lfloor n/2 \rfloor$ edges in M in blue, and we color the remaining edges in red. Any triangle of K_n has at most one blue edge, as at most two of the three nodes in the triangle are adjacent in M. Therefore, our edge-coloring satisfies the constraints.

d) We may assume $p \geq 2$, otherwise one node would already violate the condition.

In task c), we have seen that any edge-coloring on K_n satisfying our constraints contains at most $\lfloor n/2 \rfloor$ blue edges. We have seen how an edge-coloring where each triangle has at most one blue edge can be obtained from a maximum matching on K_n. However, for $n = 2p$, our edge-coloring already leads to a red K_p, and therefore violates our new constraint. In the following, we show that $n = 2p - 2$ is, in fact, the largest n for which we can find a valid edge-coloring.

We first show that such an edge-coloring exists for $n = 2p - 2$ nodes. Similarly to task c), we consider a perfect matching M of K_n. We color the $p-1$ edges in M in blue, and all the remaining edges in red, which ensures that there is no triangle with more than one blue edge. Every subgraph K_p of K_n contains a pair of nodes matched in M, and therefore contains one blue edge. Hence, our second new constraint holds as well.

We now show that no such edge-coloring exists for $n = 2p - 1$ nodes. To obtain a contradiction, we assume that there is such an edge-coloring for $n = 2p - 1$. As shown in task c), at most $p - 1$ edges are blue and the blue subgraph consists of isolated edges and vertices, hence there is a red subgraph K_{p-1}. However, each of the p nodes outside this red K_{p-1} must be incident to one blue edge towards the red K_{p-1}. Otherwise, we can extend the red K_{p-1} to a red K_p. However, this requires $p > p - 1$ blue edges.

In summary, this means that there is a solution for $2p - 2$ nodes but not for $2p - 1$ nodes, giving us a sharp bound.