①

(a) Recall the procedure of the "reduce" algorithm. In each round, node v_i sends its id to its neighbors. If v_i is a hill, its id is more significant than all neighbors. In this case, the node will be colored with accurate color.

(b) In each round, nodes between node i and node j will color two. There are $j - i + 1$ nodes, and the round will be processed $[j - i + 1 + 1]$ times. Then we have no nodes to color.

(c) One permutation is (i) = i and another is (i) = n − i. Under these permutations, we could color one node exactly in each round. In these cases, the process will be done n times.

②

(a) Suppose we are on a loop with n nodes. Each node has a vertical edge to the next node clockwise. Then we assign the number 1, 2, 3, ..., n on nodes. In this case, n nodes will send a message to the next node in the first round. In the second round, $n-1$ nodes will send the messages expected to the node with id 1. In the $i - th$ round, there are $n - i + 1$ nodes that send messages to the next node. So the total message number is $O(n^2)$.

(b) Consider we assign the number reversely as before. In this case, after the first round, only the node receives id n and will keep on sending the id. After the n round, the id n will be received by its owner, and the algorithm will be terminated. Each round, we send $O(1)$ messages, so the total message number is $O(n)$.

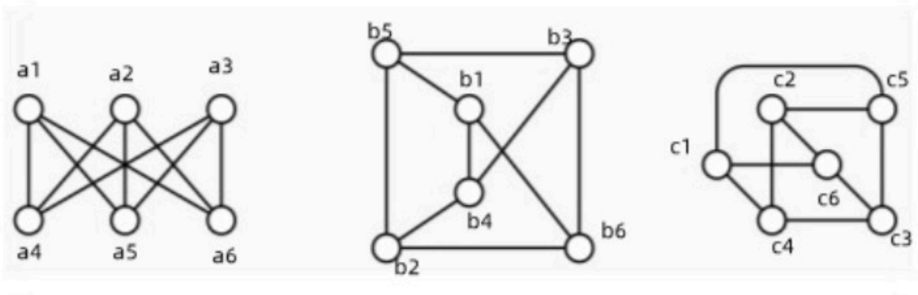
③

(a) We construct a coloring assignment and prove that it's optimal. Suppose each node has a different id number in 1, 2, 3, ..., n. We choose all nodes with even id numbers, and it's evident that we construct an Independent Set. Then we prove this set is maximum. Obviously, for each pair point (i, i+1), we can only choose one of them, which means we will never reach more giant node sets.

(b) We model this question as a CSP question and use the forward check algorithm to solve it. The color we need will not be greater than n, and the time complexity is $O(n^2d^2)$, where we use d to denote the max degree of the graph.

④

We assign the label to nodes on the graph as follows. Notice that under this label assignment, map $f(a_i) = b_i$, $g(b_i) = c_i$, $h(c_i) = a_i$ are all isomorphism.



⑤

(a) After k rounds, the size of the state space of the information sent by u is 2^k, and the same is true for v. Among them, there are only 2^k combinations of the same sample points.

The totally probability is $p = \frac{2^k}{2^k \cdot 2^2} = \frac{1}{2^k}$

(b) In each round, we have a one-half probability of dividing the size, and a one half probability that we need another round of elections. The expectation of the total number of elections can be as:

$$\begin{aligned}
\mathbb{E}(r) &= 1 \cdot \frac{1}{2} + 2 \cdot \frac{1}{4} + 3 \cdot \frac{1}{8} + \cdots \\
2\mathbb{E}(r) &= 1 \cdot 1 + 2 \cdot \frac{1}{2} + 3 \cdot \frac{1}{4} + \cdots \\
\mathbb{E}(r) &= 1 + \frac{1}{2} + \frac{1}{4} + \cdots \\
\mathbb{E}(r) &= 2.
\end{aligned}$$

⑥

We can decompose this problem into four cases:

At case1: r_1 moves clockwise and r_v moves clockwise, and the time spent is (1-d)/(v-1)

At case2: r_1 moves clockwise and r_v moves counterclockwise, and the time spent is d/(v+1)

At case3: r_1 moves counterclockwise and r_v moves clockwise, and the time spent is (1-d)/(v+1)

At cases4: r_1, r_v moves counterclockwise and the time spent is d/(v-1).

The probability of each point moving clockwise and counterclockwise is random, the total mathematical expectation is:

$$\begin{aligned}\mathbb{E}(t) &= \left(\frac{1-d}{v-1}+\frac{d}{v-1}+\frac{1-d}{v+1}+\frac{d}{v+1}\right)\cdot\frac{1}{4} \\ &= \frac{2v}{v^2-1}\end{aligned}$$

⑦

We let the cooperation algorithm designed for the two robots as follows: At the beginning, both robots run towards the center of the circle. This state will have two subsequent states, namely, the fast robot arrives first and the slow robot arrives first. If the faster robot arrives first, the robot sends the treasure directly along the radius to the circle. If the slow robot arrives first, the slow robot immediately moves to the fast robot, and the fast robot continues to move to the center of the circle, which is also the position of the slow robot. When they meet, the slow robot hands the treasure to the fast robot and the fast robot immediately moves in the opposite direction, sending the treasure around the circle. It is easy to prove that this distributed algorithm is optimal.

⑧

(a) The first sequence is periodic and the second sequence is not periodic

(b) assume that n is the minimum period of the sequence and m is the period of the sequence but it is not a multiple of n:

$a_i = a_{i+n}$

$a_i = a_{i+m}$

Further, because of Bezuo's theorem:

$a_i = a_{i+\gcd(m,n)}$

gcd(m, n) is a smaller number than n, which by definition is the period of the sequence, this leads to a contradiction because we assume that n is the smallest period of the sequence

(c) in the first round each node passes its id to its next node. In the following n-1 rounds, the nodes continue to pass the information received from the previous node to the next node. In this way we can make every node on the circle get the whole sequence.

⑨

(a) We solve this problem using a naive distributed algorithm. The first person rides halfway through the distance, leaves the bike behind, and walks to the point. The second person walks the first half of the journey and rides a bicycle for the second half. In this way, everyone does half the distance on the bike and half the distance on foot, so they get to the end at the same time

(b) Let's assume that the time t is for a person riding a bicycle with speed u is t, we can get the following equation and solve this equation.

$$ut + 1 - t = t + v(1 - t)$$

$$(u + v - 2)t = v - 1$$

$$t = \frac{v - 1}{u + v - 2}$$

Symmetrically we can calculate the time t' that a person riding at speed v rides.

$$t' = \frac{u - 1}{u + v - 2}$$