



Master2 Informatique

UE: DISTRIBUTED ALGORITHMS – 4TIN913U

Responsable : C. Gavaille

Date : 7th of January 2026

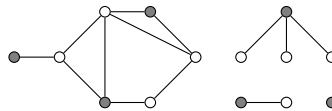
Durée : 1h30

Document: one A4-paper sheet allowed.

The language for your answers can be either french or english, but not both!

Small Dominating Set

We consider the problem of constructing a dominating set S of size $|S| \leq |V(G)|/2$ in a graph G . Such a set is denoted as an SDS. Recall that a set S is dominating for G if every vertex that is not in S has at least one neighbor in S . A dominating set S is *minimum* if its size $|S|$ is the smallest possible among all dominating sets for the graph. Such a set is denoted as an MDS. Below is an example of an SDS for a disconnected graph with 14 vertices.



Question 1. Determine an MDS for the graph above and specify its size. A drawing with the MDS highlighted is sufficient. Justify why your solution is of minimum size.

Question 2. What is, as a function of n , the size of an MDS for C_n , the cycle with n vertices? Justify your answer.

We wish to compute an SDS for C_n using a distributed algorithm in the LOCAL model. For this, we assume that C_n possesses an arbitrary 3-coloring c , and we denote by $c(u) \in \{0, 1, 2\}$ the color of vertex u . Thus, at the start, each vertex u has access to $c(u)$ as well as its identifier $\text{ID}(u) \in \{0, \dots, n-1\}$. Note that without communicating, u can determine neither n nor the most frequent color, for example. You cannot make any particular assumption about c . According to the course, it could have been computed, for example, in $\log^*(n) + O(1)$ rounds.

Question 3. Recall the definition of the function $\log^*(n)$. What is the value of $\log^*(16)$?

Question 4. Show that for any integers n, b such that $1 \leq b \leq n/2$, there exists a 3-coloring of C_n where one of the colors appears exactly b times.

Question 5. Why can such a 3-coloring not exist for C_n if $b = 0$ or $b > n/2$?

We propose a first distributed algorithm $A1$ to compute an SDS for C_n . We assume the solution S is encoded by a variable $s(u) \in \{\text{TRUE}, \text{FALSE}\}$ indicating whether $u \in S$ or not.

Algorithm $A1$, without any communication rounds, selects u if and only if $c(u) = 0$. In other words :

A1 : if $c(u) = 0$, then $s(u) := \text{TRUE}$ else $s(u) := \text{FALSE}$

Question 6. Based on the previous questions, prove that $A1$ is incorrect.

We then propose to analyze a second algorithm $A2$, this time probabilistic, which uses the function $\text{RANDOM}(3)$ returning a uniform random integer from $\{0, 1, 2\}$. We assume the RANDOM function is implemented in the same way on all vertices, specifically that the initial seed at startup is the same. Thus, the first call to $\text{RANDOM}(3)$ will yield the same value for all vertices executing it.

Algorithm $A2$, also requiring zero rounds, consists of the following :

A2 : if $c(u) = \text{RANDOM}(3)$, then $s(u) := \text{TRUE}$ else $s(u) := \text{FALSE}$

Question 7. In your opinion, is the probability that the solution S computed by A2 is an SDS for C_n at least $1/2$? Justify.

For your upcoming algorithms, you may use the following two communication primitives :

- **SENDALL(m)** : sends the message m to all neighbors of u .
- **RECEIVEALL()** : returns the list of messages received by u that were sent during the current round.

You may use **NEWROUND** to indicate the start of a new round.

Question 8. Propose a (deterministic) algorithm, avoiding the pitfalls of A1 and A2, to compute an SDS for C_n assuming it is 3-colored. Specify the number of rounds. [Hint : There exists an algorithm that runs in only $O(1)$ rounds.]

We now propose the opposite : compute a 3-coloring for C_n given a dominating set assumed to be minimum.

Question 9. Propose an algorithm to compute a 3-coloring for C_n assuming an MDS for C_n is known. Clearly specify the number of rounds.

Question 10. Deduce that it is not possible to compute an MDS for C_n in $o(\log^*(n))$ rounds.

Communication Complexity

Question 11. Explain what a communication protocol is for computing a boolean function f . What are its main rules? What is the intended objective?

Question 12. To prove which distributed computing result seen in class did communication complexity serve us?

Question 13. Recall the definition of a fooling set (or discriminant set) F_v for a boolean function f and a value $v \in \{0, 1\}$.

Let $\text{CC}(f)$ denote the communication complexity of f .

Question 14. Prove that if f has a fooling set F_v such that $|F_v| \geq 2^k$ for some integer $k \geq 1$ and a value $v \in \{0, 1\}$, then $\text{CC}(f) \geq k + 1$. [Hint : Pay attention to the “+1”! Use the connections established in the course between $\text{CC}(f)$, the number of transcripts (i.e., scripts), and the size of fooling sets.]

Let Π be a communication protocol for a certain function f such that Π generates only two possible transcripts over all inputs $(x, y) \in \{0, 1\}^{2n}$ of f .

Question 15. Show that Alice and Bob can internally use Π to compute $f(x, y)$ by exchanging at most 2 bits in total, regardless of the input (x, y) . This new protocol must be prefix-free.

Consider the function $\text{lt}_n(x, y) \in \{0, 1\}$ (for less-than) which, given two binary strings x, y , each of n bits, returns 1 if and only if “ $x < y$ ”, i.e., if the integers represented in binary by the strings x and y satisfy $x < y$. For example, $\text{lt}_2(11, 10) = 0$. However, $\text{lt}_2(01, 10) = 1$. Obviously, $\text{lt}_n(x, x) = 0$ for all $x \in \{0, 1\}^n$.

Question 16. Calculate $\text{CC}(\text{lt}_n)$. Justify your answer. [Hint : Note that $\text{CC}(\text{lt}_n) \leq n + 1$.]

END.