Distributed Algorithms

25 novembre 2021 - v1

Master2 Informatique

---

*This **individual** homework must be uploaded on the moodle page of the lecture no later than **December 13th, 2021**. The deposit must be composed of a unique file, either in the plain text format or in pdf format (no .zip of .docx files). Important: all questions must fully argued, and this will take into account for the scoring of your homework. The langage for your answers can be either French or English. Please avoid to mix both langages.*

Layer Coloring

We shall consider only deterministic and distributed algorithms in the LOCAL model. We use the same notations as in the lecture notes. In particular, for a graph G, let $\Delta = \max\{\deg(u) : u \in V(G)\}$ denote the maximum degree of the graph G.

Question 1. *For the $(\Delta+1)$-coloring problem, what is the motivation for the choice of this number of colors? That is, why do we choose $\Delta+1$ colors rather than another number of colors, say $\lceil \Delta/2 \rceil$ or n colors for instance?*

Let $f(\Delta, n)$ be the time complexity of some fast distributed algorithm providing, for every graph G of n vertices, an $(\Delta+1)$-coloring.

Question 2. *Without any proof, give an upper bound on $f(\Delta, n)$ that is described in the lecture notes.*

We will introduce now a new coloring method that applies to graphs having some nice decomposition. More precisely, a (k,t)-*decomposition* of a graph G is a sequence $(G_1, G_2, \ldots, G_t)$ of t induced subgraphs of G such that:

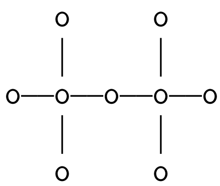
each vertex u of G

- appears in a unique subgraph G_i of the decomposition, and we will say that u is of *level* i.
- has at most k neighbors of level that is greater or equal to its own level.

For convenience, we denote by G_i^+ the subgraph of G induced by all the vertices of level $\geqslant i$. Of course, $G_1^+ = G$.

A (sequential) way to obtain a (k,t)-decomposition for G, if it exists, is to place in G_1 all the vertices of G_1^+ (i.e., G) with degree $\leqslant k$, and to remove them from G in order to obtain the subgraph G_2^+. Then, we continue similarly by removing all the vertices of degree $\leqslant k$ from G_2^+, and move them into G_2, in order to obtain G_3^+. And so on, t times, until to get the complete sequence $(G_1, \ldots, G_t)$. In practice one can stop whenever one get an empty subgraph G_i^+ because if G has a (k,t)-decomposition[1] $(G_1, \ldots, G_{t-1}, \varnothing)$, then $(G_1, \ldots, G_{t-1})$ is a $(k, t-1)$-decomposition of G as well.

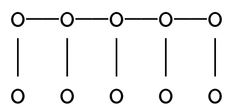
Question 3. *Give a $(1,3)$-decomposition of the following tree, by simply reporting the level of each vertex.*



[1] By denoting by $\varnothing$ the empty graph, so the graph with no vertices.

Question 4. *What is the condition that must fulfill a graph G with n vertices in order to have a (k, n)-decomposition?*

Let consider the *comb* graph with n vertices, assuming n is even, be the graph obtained from a path of $n/2$ vertices each one having a pending vertex (a vertex of degree 1). Hereafter is a comb with 10 vertices.



Question 5. *What is the minimum number t of levels in a $(1, t)$-decomposition of a comb with 10 vertices?*

Question 6. *In general, what is the minimum number t of levels in a $(1, t)$-decomposition of a comb with n vertices?*

Question 7. *What is the minimum number t of levels in a $(2, t)$-decomposition of a comb with n vertices?*

Question 8. *Assume that G is a graph with n vertices and m edges, and that there exist two integers $k, b \geqslant 1$ such that $m \leqslant (k+1) \cdot (1 - 1/b) \cdot n/2$. Show that G has at least n/b vertices of degree $\leqslant k$.*

In the remaining, you may assume that the property stated in Question 8 is indeed true.

Question 9. *Show that every tree with n vertices has a $(3, \lceil \log_2 n \rceil - 1)$-decomposition and also a $(2, \lceil \log_{3/2} n \rceil - 1)$-decomposition, assuming n is large enough. [Hint: Think that the decomposition may stop whenever it remains at most $k+1$ vertices. Why?]*

We shall present now a method to construct a $(k+1)$-coloring of a graph G having a (k, t)-decomposition $(G_1, \ldots, G_t)$. We assume that k and t are parameters known by all the vertices, and that colors are integers taken from $[0, k]$. We will make the use of the function FirstFree seen in the lecture and defined by $\text{FirstFree}(X) = \min(\mathbb{N} \setminus X)$, for every subset $X \subseteq \mathbb{N}$.

The three main steps of the method are the following:

1. Compute the level of every vertex of G.
2. Compute (in parallel) a $(k+1)$-coloring of every G_i.
3. For each $i = t-1, t-2, \ldots, 1$ in this order, update the coloring of vertices only in G_i thanks to FirstFree applied by colors and for the graph G_i^+.

Question 10. *Show that this method indeed provides a valid $(k+1)$-coloring for G.*

The goal now is to give a full description of a distributed algorithm for this method. We will assume that a vertex does not know its degree in G when it starts its computation. But note that it can learn it within a single round of communication with its neighbors.

Question 11. *Describe a distributed algorithm for Step 1 of the medthod that is in charge of computing the level of each vertex u of G, hereafter denoted by $\ell(u)$. Give an analysis of the number of rounds.*

Question 12. *Describe a distributed algorithm for Step 2 of the medthod. Using fonction f defined above in Question 2, give an analysis of the number of rounds.*

Question 13. *Describe a distributed algorithm for Step 3 of the medthod. Give an analysis of the number of rounds.*

Question 14. *Give the total number of rounds of your distributed algorithm for computing a $(k+1)$-coloring for G.*

Question 15. *Explain and justify how to compute in* $O(\log n)$ *rounds a* 3*-coloring of a tree with* n *vertices.*

Question 16. *Discuss whether this coloring is a progress or not with respect to the* 3*-coloring distributed algorithm for trees and cycles we have seen in the lecture and that runs in* $O(\log^* n)$ *rounds.*

Question 17. *Consider a square grid* G *with a total of* n *vertices. Give a* $(3,t)$*-decomposition for* G *with the minimum possible* t*. What is this minimum* t*. Justify. Same question for a* $(2,t)$*-decomposition.*

Question 18. *Show that every planar graph with* n *vertices has a distributed algorithm for computing a* 7*-coloring in* $O(\log n)$ *rounds.*

FIN.