Distributed Graph Algorithms **Spring 2023**

Exercise 2: May 17

Lecturer: Merav Parter

Exercise 1 (Coloring in $O(\log^* n)$ Rounds). In this exercise, we consider slight extensions of the $O(\log^* n)$-round algorithm $\mathcal{A}$ for 3-coloring oriented trees that we saw in class. In the following, it is sufficient to specify the modifications, and explain the correctness. (a) Show that a similar algorithm also works for the n-length cycle (even without orientation) upon minor modifications. (b) Show that for any n-vertex graph with maximum degree Δ, one can modify algorithm $\mathcal{A}$ to provide $2^{O(\Delta)}$-coloring within $O(\log^* n)$ rounds.

Exercise 2 (Color Reduction). In this exercise, we consider general n-vertex graphs with maximum degree Δ. Prove the following two statements. (a) Given a k-coloring $C : V \to [1, k]$ of a graph with $k \geq \Delta + 2$ colors, in a single round one can compute a $(k-1)$-coloring $C' : V \to [1, k-1]$.
(b) Given a k-coloring $C : V \to [1, k]$ of a graph with $k \geq \Delta + 2$ colors, in $O(\Delta \log(k/(\Delta + 1)))$ rounds, one can compute a $(\Delta + 1)$-coloring $C' : V \to [1, \Delta + 1]$. **Hint**: Split the colors $[1, k]$ to several buckets (how many?) and reduce the colors of all the buckets simultaneously (use (a)!). Show first that in $O(\Delta)$ rounds, we can reduce the number of colors to at most $k/2$, and repeat this procedure for $O(\log(k/\Delta + 1))$ rounds.

Exercise 3 (FD of Bounded Arboricity Graphs). The arboricity of a graph $G = (V, E)$, denoted by $a(G)$, is the minimum number a of edge-disjoint forests $F_1, \ldots, F_a$ whose union covers the entire edge set[1] E. Such a decomposition is called a-forest decomposition. Forest decompositions have many applications (e.g., $O(a)$ coloring for graphs with arboricity a). In this exercise, we will provide a local algorithm for computing an approximate forest decomposition with at most $(2+\epsilon) \cdot a(G)$ forests. In the distributed output format of the decomposition algorithm, every vertex is required to know its parent in each of the forests $F_1, \ldots, F_{(2+\epsilon)\cdot a(G)}$ (the union of all these forests should cover $E(G)$). Throughout, assume that all vertices in G are given as input the parameter $a(G)$ and the approximation parameter ϵ.

The first step for computing the forest decomposition is based on computing a vertex *partitioning* of the graph $L_1, \ldots, L_k$ such that each vertex $v \in L_i$ has at most $(2 + \epsilon)a(G)$ neighbors in $G(\bigcup_{j=i}^{k} L_i)$. This partitioning is based on showing the following observation.

(a) A graph G with arboricity $a = a(G)$ has at least $\epsilon/(2 + \epsilon)|V(G)|$ vertices with degree $\leq (2 + \epsilon)a$.

(b) Use claim (a) to define the partitioning $L_1, \ldots, L_k$ for $k = O(1/\epsilon \cdot \log n)$ using $O(k)$ rounds. In the distributed output format, each vertex v should learn its index i such that $v \in L_i$.

(c) Use the vertex partitioning of (b), to orient the edges of G such that the out-degree of each vertex is at most $(2 + \epsilon)a$. Show that this can be done in a single communication round. In the output format, each vertex v is required to learn the orientation of all its edges (and thus in particular, its outgoing edges).

(d) Finally, use the edge orientation of (c) to locally define the forest decomposition $F_1, \ldots, F_{(2+\epsilon)\cdot a(G)}$. Show that in your solution, each F_i is indeed a forest.

[1] Clearly graphs with bounded arboricity $a(G) = O(1)$ are sparse (with at most $O(n)$ edges), however, they might contain high-degree nodes (e.g., the star graph has arboricity of 1). Therefore the maximum degree Δ might be considerably larger than $a(G)$.