University of Freiburg
Dept. of Computer Science
Prof. Dr. F. Kuhn
Z. Parsaeian, G. Schmid

Distributed Graph Algorithms
Exercise Sheet 7

Motivation. The *Sinkless Orientation* problem plays a central role in the theory of distributed graph algorithms in the LOCAL model. One way to argue for its importance is to show that it is *easier* than other fundamental problems: if Sinkless Orientation already requires many rounds, then other problems that reduce to it must be hard as well. Another way is to say that other problems are easy, if we know how to solve Sinkless Orientation fast. We will see one example each.

Δ-coloring in low degree graphs

We compare the complexity of 5-coloring and 4-coloring graphs of maximum degree at most 4, and show how the hardness of 4-coloring can be explained via a reduction to Sinkless Orientation. Notice that one is in essence a $\Delta + 1$ coloring, while the other is Δ coloring.

Exercise 1: 5-Coloring Degree-4 Graphs is Easy

Let $G = (V, E)$ be a graph of maximum degree at most 4.

Show that there exists a deterministic distributed algorithm that computes a proper 5-coloring of G in $O(\log^* n)$ rounds in the LOCAL model.

Exercise 2: Why 4-Coloring Degree-4 Graphs is Hard

In contrast to Exercise 1, we now argue that 4-coloring graphs with maximum degree at most 4 is a much harder problem. We do this by relating it to the Sinkless Orientation problem.

To do so, we solve the following two exercises.

Exercise 2.1: From Vertex Coloring to Edge Coloring

Let G be a tree of maximum degree at most 3. Suppose there exists a deterministic distributed algorithm that computes a proper 4-vertex-coloring of any graph of maximum degree at most 4 in T rounds.

Show that this implies the existence of a deterministic $O(T)$-round distributed algorithm that computes a proper 4-edge-coloring of any tree of maximum degree at most 3.

Exercise 2.2: From Edge Coloring to Sinkless Orientation

Assume there is an a deterministic $O(T)$-round distributed algorithm that computes a proper 4-edge-coloring of any tree of maximum degree at most 3.
Let $G = (V, E)$ be a 2-colored tree of maximum degree at most 3. (So we have the same setup as in the $\Omega(\log n)$ Sinkless Orientation lower bound of the lecture.)

Show that the Sinkless Orientation problem on G can be solved in $O(T)$ rounds.

Sink-and Sourceless Orientation

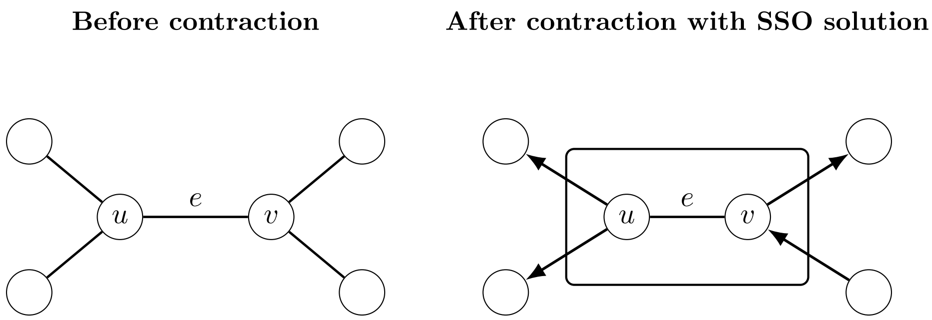
In the problem of Sink-and Sourceless Orientation (SSO), we require an orientation of the edges, such that each node of degree at least 3 has both an incoming and an outgoing edge. In other words, no node is a sink (in the same way as in Sinkless Orientation), but also no node is a source (all edges are outgoing). Clearly the problem is at least as hard as Sinkless Orientation (any solution to Sink-and Sourceless Orientation is a solution to Sinkless Orientation). We will now show, that we can solve Sink-and Sourceless Orientation just as fast as Sinkless Orientation.

Exercise 3: SSO is easy for large degrees

Consider the problem SSO_6, where only nodes of degree at least 6 have to output a correct solution.

Show that if we are given a T round algorithm for SO, we can solve SSO_6 in T rounds.

Dealing with low degree nodes The problem is trivial on nodes of degree 0,1,2 and we have have shown above that we can deal with degree ≥ 6 nodes easily. So the problem are nodes of degree $3, 4$ and 5. Our solution is to *merge* low degree nodes to obtain higher degree nodes. For example take two adjacent nodes u, v of degree 3, by acting as if they were just a single node (so contracting the edge between them), we have created a node of degree 4.



Exercise 4: Merging Nodes

Show that if we have a solution to SSO in the graph, where we treat the nodes u, v as a single node (that now has degree 4), we can always orient the edge $e = \{u, v\}$ to make both u and v happy.

Clusters: We now generalise this to more than just contracting a single edge. For any graph $G = (V, E)$, we say that $C \subset V$ is a cluster of V, if

- For any $u, v \in C$, there exists a path from u to v that is entirely contained in C of length $O(1)$.

We say that the degree of C is the number of edges with one endpoint in C and one endpoint in $V \setminus C$. We denote by E_C the set of edges contained in C, that is,

$$E_C = \big\{\{u, v\} \in E \mid u, v \in C\big\}.$$

If our clusters have degree at least 6, then, using Exercise 3, we can act as if it was a degree ≥ 6 node and obtain an orientation of the edges outside the cluster with an incoming and an outgoing edge.

Exercise 5: Clusters with Incoming and Outgoing Edges

Given a cluster C of G, assume that all edges outside of C are oriented, and that this orientation constitutes a valid solution for SSO, for the graph in which the cluster C is contracted to a single node. In particular, all edges incident to C are oriented, with the guarantee that at least one such edge is oriented towards C and at least one is oriented away from C.
Note that E_C are the only edges in the graph that are not yet oriented.

Show that there is an orientation of E_C, such that all nodes in C have an incoming and an outgoing edge.

Hint: Start with a path (fully contained in C) that connects the incoming edge e_{inc} to the outgoing edge e_{out}.

If we cannot obtain such a cluster with a large enough degree, we will instead find a cluster with a short cycle.

Exercise 6: Clusters with Cycles

1. Let $C \subset V$ be some cluster of G, that contains a cycle that is completely contained in C.
2. Assume that all edges outside of C are oriented (in an completely arbitrary way).

Show that there is an orientation of E_C, such that all nodes in C have an incoming and an outgoing edge.

Exercise 6: Getting good clusters

We now have to obtain suitable clusters for all nodes with degree that is too small.
Let $G = (V, E)$ be an arbitrary graph and let $G' = (V', E')$ be the subgraph of G induced by all nodes of degree 3, 4, or 5 in G.
Your task is to design an $O(\log^* n)$ algorithm that partitions the node set V' into sets $C_1, \ldots, C_k, I$ such that

1. $V' = C_1 \cup \ldots \cup C_k \cup I$ and $C_1, \ldots, C_k, I$ are mutually disjoint.
2. The set I contains all isolated nodes.
3. Each set C_i is a cluster that either has degree ≥ 6, or contains a cycle.

Give a description of such an algorithm and argue that it always produces valid clusters of G'.

Hint: You may try to first solve a different $O(\log^* n)$ problem on G' and then use that to solve the task.
Hint: If you can't increase the degree to ≥ 6 immediately, increase it to just ≥ 4 and try running the procedure again.

Exercise 7: Computing SSO

We now have everything we need to solve SSO.
Combine the results from Exercises 3,5,6 to solve the problem of SSO in $O(\log^* n + T_{SO})$, where T_{SO} is the time to solve Sinkless Orientation.
Hint 1: Note that Exercise 6 does not immediately give you clusters that guarrantee a degree ≥ 6.
Hint 2: You still have to deal with the isolated nodes, that are not put into some cluster in Exercise 6. What can you say about nodes that are isolated in G'?