Chapter 1 - Vertex Coloring

Problem 1.1 - Vertex Coloring
Given an undirected graph $G = (V, E)$, assign a color cu to each vertex $u \in V$ such that the following holds: $e = (v, w) \in E \Rightarrow cv \neq cw$.

Definition 1.2 - Node Identifiers
Each node has a unique identifier, e.g., its IP address. We usually assume that each identifier consists of only log n bits if the system has n nodes.

Definition 1.3 - Chromatic Number
Given an undirected Graph $G = (V, E)$, the chromatic number $\chi(G)$ is the minimum number of colors to solve Problem 1.1.

Algorithm 1 - Greedy Sequential *(see page 6)*

- **Vertex Coloring**
- Non-distributed
- Centralized
- **Theorem 1.5:** is correct and terminates in n "steps". The algorithm uses at most $\Delta + 1$ colors.

Definition 1.4 - Degree
The number of neighbors of a vertex v, denoted by $\delta(v)$, is called the degree of v. The maximum degree vertex in a graph G defines the graph degree $\Delta(G) = \Delta$.

Definition 1.6 - Synchronous Distributed Algorithm
n a synchronous al- gorithm, nodes operate in synchronous rounds. In each round, each processor executes the following steps:

1. Do some local computation (of reasonable complexity).
2. Send messages to neighbors in graph (of reasonable size).
3. Receive messages (that were sent by neighbors in step 2 of the same round).

Algorithm 3 - Reduce *(see page 7)*

- **Theorem 1.8:** is correct and has time complexity n. The algorithm uses at most $\Delta + 1$ colors.

Definition 1.7 - Time Complexity
For synchronous algorithms (as defined in 1.6) the time complexity is the number of rounds until the algorithm terminates.

Lemma 1.9
$\chi(Tree) \leq 2$

Algorithm 4 - Slow Tree Coloring *(see page 8)*

- Time Complexity: Height of tree (up to n)
- Does not need to be synchronous

Definition 1.10 - Asynchronous Distributed Algorithm
In the asynchronous model, algorithms are event driven ("upon receiving message ..., do..."). Processors cannot access a global clock. A message sent from one processor to another will arrive in finite but unbounded time.

Definition 1.11 - Time Complexity
For asynchronous algorithms (as defined in 1.6) the time complexity is the number of time units from the start of the execution to its completion in the worst case (every legal input, every execution scenario), assuming that each message has a delay of at most one time unit.

Definition 1.12 - Message Complexity
The message complexity of a syn- chronous or asynchronous algorithm is determined by the number of messages exchanged (again every legal input, every execution scenario).

Theorem 1.13 - Slow Tree Coloring
Algorithm 4 (Slow Tree Coloring) is correct. If each node knows its parent and its children, the (asynchronous) time complexity is the tree height which is bounded by the diameter of the tree; the message complexity is $n - 1$ in a tree with n nodes.

Definition 1.14 - Log-Star
$\forall x \leq 2 : log * x := 1 \forall x > 2 : log * x := 1 + log * (log(x))$

Algorithm 5 - "6-Color" *(see page*

- *Time Complexity:* $O(log^*(n))$

) 10

Theorem 1.15 - 6-Color
Algorithm 5 ("6-Color") terminates in $log^*(n)$ time.

Algorithm 6 - Shift Down *(see page 11)*

- **Lemma 1.16:** Preserves coloring legality: also siblings are monochromatic

Algorithm 7 - Six-2-Three *(see page 11)*

- **Theorem 1.17:** colors a tree with three colors in $O(log^*(n))$.

Chapter 2 - Leader Election

Setup:

- Ring topology

Problem 2.1 - Leader Election
Each node eventually decides whether it is a leader or not, subject to the constraint that there is exactly one leader.

Definition 2.2 - Anonymous
A system is anonymous if nodes do not have unique identifiers.

Definition 2.3 - Uniform
An algorithm is called uniform if the number of nodes n is not known to the algorithm (to the nodes, if you wish). If n is known, the algorithm is called non-uniform.

Lemma 2.4
After round k of any deterministic algorithm on an anonymous ring, each node is in the same state s_k.

Theorem 2.5 - Anonymous Leader Election
Deterministic leader election in an anonymous ring is impossible.
(Also holds for other symmetric network topologies (e.g., com- plete graphs, complete bipartite graphs), but does not hold for randomized algorithms.)

Algorithm 8 - Clockwise *(see page 17)*

- **Theorem 2.6:** is correct. The **time complexity** is $O(n)$. The **message complexity** is $O(n2)$.

Algorithm 9 - Radius Growth *(see page 18)*

- **Theorem 2.7:** is correct. The **time complexity** is $O(n)$. The **message complexity** is $O(nlogn)$.
- Asynchronous
- Uniform

Definition 2.8 - Execution
Definition 2.8 (Execution). An execution of a distributed algorithm is a list of events, sorted by time. An event is a record (time, node, type, message), where type is “send” or “receive”.

Definition 2.9 - Open Schedule
A schedule is an execution chosen by the scheduler. An open (undirected) edge is an edge where no message traversing the edge has been received so far. A schedule for a ring is open if there is an open edge in the ring.

Lemma 2.10
Given a ring R with two nodes, we can construct an open schedule in which at least one message is received. The nodes cannot distinguish this schedule from one on a larger ring with all other nodes being where the open edge is.

Lemma 2.11
By gluing together two rings of size $n/2$ for which we have open schedules, we can construct an open schedule on a ring of size n. If $M(n/2)$ denotes the number of messages already received in each of these schedules, at least $2M(n/2) + n/4$ messages have to be exchanged in order to solve leader election.

Lemma 2.12
Any uniform leader election algorithm for asynchronous rings has at least message complexity $M(n) \geq n(log(n) + 1)$.

Theorem 2.13 - Asynchronous Leader Election Lower Bound
Any uniform leader election algorithm for asynchronous rings has $\Omega(n * log(n))$ message complexity.

Algorithm 10 - Synchronous Leader Election *(see page 21)*

- Non-uniform (i.e. the ring size n is known)
- Every node starts at same time
- Mimimum identifies (integers) becomes the leader
- **Time Complexity:** $m * n$, where m is the minimum identifier
- **Message Complexity:** n.

Chapter 3 - Tree Algorithms

Definition 3.1 - Broadcast
A broadcast operation is initiated by a single processor, the source. The source wants to send a message to all other nodes in the system.

Definition 3.2 - Distance, Radius, Diameter
The **distance** between two nodes u and v in an undirected graph G is the number of hops of a minimum path between u and v. The **radius of a node** u is the maximum distance between u and any other node in the graph. The **radius of a graph** is the minimum radius of any node in the graph. The **diameter** of a graph is the maximum distance between two arbitrary nodes.
Relation between radius and diameter: $R \leq D \leq 2R$

Theorem 3.3 - Broadcast Lower Bound
The message complexity of broadcast is at least $n - 1$. The source's radius is a lower bound for the time complexity.

Definition 3.4 - Clean
A graph (network) is clean if the nodes do not know the topology of the graph.

Theorem 3.5 - Clean Broadcast Lower Bound
For a clean network, the num- ber of edges is a lower bound for the broadcast message complexity.

Algorithm 11 - Flooding *(see page 24)*

-

Algorithm 12 - Echo *(see page 25)*

- **Message Complexity:** $n - 1$
- Together with flooding the message complexity is $O(m)$, where $m = |E|$ is the number of edges in the graph.
- **Time Complexity:** Depth of the spanning tree.

Algorithm 13 - Dijkstra BFS *(see page 26)*

- **Theorem 3.6:** The **time complexity** is $O(D2)$, the **message complexity** is $O(m + nD)$, where D is the diameter of the graph, n the number of nodes, and m the number of edges.

Chapter 4 - Distributed Sorting

Chapter 5 - Maximal Independent Set

Chapter 6 - Locality Lower Bounds

Chapter 7 - All-to-All Communication

Chapter 8 - Social Networks

Chapter 9 - Shared Memory

Chapter 10 - Shared Objects

Chapter 11 - Wireless Protocols

Chapter 12 - Synchronization

Chapter 13 - Peer-to-Peer Computing