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)*

- 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 10)*

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

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.

Chapter 3 - Tree Algorithms

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