Problem 1. Let $T$ be a $(2, 3)$-tree on a set $S$ of integers. Suppose that each node in the tree stores a pointer to its parent. You are given the leftmost leaf $z$ of $T$, and asked to report the $k$ smallest integers in $T$. Describe an algorithm to do so in $O(k)$ time.

Problem 2. Let $G = (V, E)$ be a directed graph. Suppose that we perform BFS starting from a source vertex $s$, and obtain a BFS-tree $T$. For any vertex $v \in V$, denote by $l(v)$ the level of $v$ in the BFS-tree. Prove that BFS enqueues the vertices $v$ of $V$ in non-descending order of $l(v)$.

Problem 3. Let $G = (V, E)$ be a directed graph. Suppose that we perform BFS starting from a source vertex $s$, and obtain a BFS-tree $T$. For any vertex $v \in V$, prove that the path from $s$ to $v$ in $T$ is a shortest path from $s$ to $v$ in $G$.

Problem 4. Let $G = (V, E)$ be an undirected graph. We will denote an edge between vertices $u, v$ as $\{u, v\}$. Next, we define the single source shortest path (SSSP) problem on $G$. Define a path from $s$ to $t$ as a sequence of edges $\{v_1, v_2\}, \{v_2, v_3\}, ..., \{v_t, v_{t+1}\}$, where $t \geq 1$, $v_1 = s$, and $v_{t+1} = t$. The length of the path equals $t$. Then, the SSSP problem gives a source vertex $s$, and asks to find shortest paths from $s$ to all the other vertices in $G$. Adapt BFS to solve this problem in $O(|V| + |E|)$ time. Once again, you need to produce a BFS tree where, for each vertex $v \in V$, the path from the root to $v$ gives a shortest path from $s$ to $v$.

Problem 5 (Connected Components). Let $G = (V, E)$ be an undirected graph. A connected component (CC) of $G$ includes a set $S \subseteq V$ of vertices such that

- For any vertices $u, v \in S$, there is a path from $u$ to $v$, and a path from $v$ to $u$.

- (Maximality) It is not possible to add any vertex into $S$ while still ensuring the previous property.

For example, in the above graph, $\{a, b, c, d, e\}$ is a CC, but $\{a, b, c, d\}$ is not, and neither is $\{g, f, e\}$.

Prove: Let $S_1, S_2$ be two CCs. Then, they must be disjoint, i.e., $S_1 \cap S_2 = \emptyset$.

Problem 6. Let $G = (V, E)$ be an undirected graph. Describe an algorithm to divide $V$ into a set of CCs. For example, in the example of Problem 5, your algorithm should return 3 CCs: $\{a, b, c, d, e\}$, $\{g, f\}$, and $\{h, i, j\}$.
**Problem 7.** Recall that, in the DFS algorithm, after we have grown a DFS-tree, we may need to restart from an arbitrary vertex that remains white. Multiple re-starts may be necessary throughout the algorithm. Describe how to find the re-starting white vertices efficiently so that the overall execution time of the algorithm is $O(|V| + |E|)$. 