BFS and DFS Trees

Kelin Luo

Lemma 1. *Consider the case where the BFS tree and DFS tree both start from the same vertex. If BFS tree = DFS tree, then the graph G is a tree.*

Proof. Since the BFS tree equals the DFS tree, both trees span all vertices and contain no cycles.

We will prove this lemma by contradiction. Assumption: Suppose there exists a graph G that is not a tree, and its BFS tree equals its DFS tree. By the definition of a tree (a connected, acyclic graph), this implies that G has at least one cycle.

Without loss of generality, suppose the cycle contains an edge $\{u, v\}$ connecting two nodes of the BFS (and DFS) tree, u and v, where $\{u, v\}$ is not an edge of the BFS (and DFS) tree.

We could distinguish two cases based on the starting point.

- Both the BFS tree and DFS tree are constructed starting from either vertex u or v. In this case, the edge $\{u, v\}$ must be included in the BFS tree, as u and v are adjacent vertices.
- Both the BFS tree and DFS tree are constructed starting from a vertex that is neither u nor v. In this case, the edge $\{u, v\}$ must be included in the DFS tree because the DFS algorithm will explore unvisited vertices, either u nor v, before backtracking.

Therefore, our initial assumption that the graph G is not a tree when the BFS tree equals the DFS tree leads to a contradiction. Hence, if the BFS tree equals the DFS tree, then the graph G is indeed a tree. This completes the proof by contradiction. □

Note: The above proof applies to the following statement: If the BFS tree equals the DFS tree when starting from the same (arbitrary) vertex, then the graph G is a tree.

However, if the BFS tree equals the DFS tree when starting from different arbitrary vertices, the graph G is not necessarily a tree. Consider the following example proposed by some students during the lecture: Given a graph $G = (V, E)$ where $V = \{a, b, c\}$ and $E = \{\{a, b\}, \{b, c\}, \{c, a\}\}$, if BFS starts from vertex b and DFS starts from vertex a, then the BFS tree equals the DFS tree. However, in this case, graph G is not a tree.