On properties of DFS and BFS trees
Graph Theory

Al Kafi Muhtasin

July 28, 2024

In this report, we are going to prove lemma 4.6 and 4.7 from the lecture notes which mention some simple properties of DFS and BFS trees.

Lemma 4.6 For an undirected graph, any depth-first search tree has no cross-edges.

Proof. Let T be a tree spanned by depth-first search, and e be a non-tree edge with endpoints $x, y \in T$ and $dfnumber(x) < dfnumber(y)$. Now, the goal of a depth-first search is to explore as far as possible in each branch. We consider the branch containing the vertex x. After reaching vertex x, the exploration would have extended further to y via e if there were no other frontier edges from x. As this is not the case, there were other frontier edges. Let us assume that proceeding along those edges does not reach vertex y. Then, the search would have backtracked to x and extend to y via e. That means, there is another path leading to y from x other than e. Thus, x and y are of the same family i.e. e is not a cross-edge.

□

Lemma 4.7
i) Let x, y be two vertices in a breadth-first search tree, then the property $level(y) > level(x)$ implies $dfnumber(y) > dfnumber(x)$.

Proof. This proof is done by using mathematical induction.
For $level(x) = 0$, the statement is trivial as the only vertex at 0 depth is the root itself with $dfnumber(x) = 0$ and any other vertex will have $dfnumber(y) > 0$.
Let the statement be true for some $level(x) = k$. We take $level(y) > level(x) = k+1$. If x_0 and y_0 are parents of x and y respectively in the tree, then we have $level(y_0) > level(x_0) = k$. By our assumption, $dfnumber(y_0) > dfnumber(x_0)$. That means, the BFS algorithm had operated on x_0 first due to default priority and chose x_0x before y_0y i.e x was discovered before y. Thus, $dfnumber(y) > dfnumber(x)$ and by induction, the proof is done.

□

ii) Any breadth-first search tree provides the shortest path tree of an unoriented graph with a given root.

Proof. A shortest-path tree for a undirected, connected graph G is spanning tree T with root r such that the unique path in T from r to each vertex x is a shortest path in G from r to x.

Let T be a tree spanned by breadth-first search and let us consider a vertex x such that $level(x) = k$ i.e. $d(r, x) = k$ in the tree. Assume the shortest path from r to x in G has length $k' < k$. We notice that the vertices for this shortest path are also included in T as BFS produces spanning tree for undirected graph. Let x_0 be the parent of x in T and $x'_0 \in T$ be the previous vertex of x in the shortest path in G. Then, we have $d(r, x'_0) = level(x'_0) = k' - 1 < k - 1 = level(x_0)$ which implies $dfnumber(x'_0) < dfnumber(x_0)$. Then, the BFS algorithm would have chosen the frontier edge x'_0x even before x_0x could become a frontier edge resulting in $level(x) = k'$. As that is not the case, no such shortest path exist i.e. the path from r to x in T is the shortest. Thus, T is shortest-path tree. □