CS330HW5

ras70

October 2017

Problem 2A

Assume towards contradiction that (u, v) is a cross edge in the DFS tree corresponding to an edge in the undirected graph. Since the graph is undirected it can be established that since there is an edge (u, v) there must also be an edge (v, u) given the symmetry of an undirected edge.

By the definition of a cross edge, the following must have happened in sequential order during graph traversal:

1. Vertex v enters the stack.
2. Vertex v leaves the stack
3. Vertex u enters the stack
4. Vertex u leaves the stack

Said differently since (u, v) is a cross edge, $pre(v) < post(v) < pre(u) < post(u)$ must hold.

Given the condition $pre(v) < pre(u)$ Vertex v must have been visited in the traversal algorithm and added to the stack before Vertex u. By the design of the DFS algorithm all neighboring vertices (and children) of a vertex will be added before that given vertex leaves the stack. More formally, given the design of the DFS traversal all descendants of a Vertex A will be enter the stack after A enters the stack and leave the stack before Vertex A leaves the stack. Therefore, since the edge (v, u) exists Vertex u will be added to the stack before Vertex v is removed, or $pre(u) < post(v)$. However, in the definition of a cross edge $post(v) < pre(u)$. Thus, by contradiction (u, v) cannot be a cross-edge in an a DFS tree generated from an undirected graph.

Problem 2B

Assume towards contradiction that a forward edge (u, v) exists in the BFS tree. The existence of this edge establishes that u was examined before v in the BFS traversal. Additionally, since (u, v) is a forward edge $pre(u) < pre(v) < post(v) < post(u)$ where *pre* and *post* indicate the timing of a vertex entering and leaving the queue respectively. In other words, since (u, v) is a forward edge u must enter the queue before v. This means when u is being examined v will yet to have been visited.

Let $d(a, b)$ denote the distance between Vertices a and b in the BFS tree where b is the descendant of a and each edge has weight 1. By this convention $d(c, d) = 1$ must hold for any tree edge (c, d). By definition a forward edge is *not* a tree edge. Therefore, since (u, v) is a forward edge, not a tree edge, the distance between u and v cannot be 1 in the BFS tree (otherwise it would be a tree not a forward edge). Since distances are positive and $d(u, v) \neq 1$, then

$$d(u, v) > 1 \tag{1}$$

During the BFS traversal an edge (x, y) is added to the BFS tree if Vertex y is added to the queue while examining x. Since (u, v) is an edge in the graph, Vertex v must be a neighbor of Vertex u. As explained in the first paragraph while u is being examined v must be unvisited and therefore by BFS design will be added to the queue. Thus, since v is added to the queue while u is being examined (u, v) is an edge in the BFS tree. Thus, (u, v) is a tree edge in the BFS tree and,

$$d(u, v) = 1 \tag{2}$$

It has been established that if (u, v) is a forward edge $d(u, v) > 1$ and $d(u, v) = 1$. Thus, equations 1 and 2 come in direct contradiction and (u, v) cannot be a forward edge.