COMP523 Tutorial 3 - Solutions

Coordinator: Aris Filos-Ratsikas Demonstrator: Michail Theofilatos

October 10, 2019

Problem 1

Show that if a graph G has no cycles of odd length, then it is bipartite.

Solution

We will make the argument for a graph G that is connected; for general graphs we can apply the same argument for each connected component separately. Consider the BFS tree, which is the spanning tree obtained from running BFS on G, starting from some node s. Recall that BFS puts the nodes of the tree in *levels*, with the root (i.e., the node s) being at level 0, the nodes which are adjacent to the root are in level 1 and so on. In particular, for any node u of level i, the shortest path from s to u in G has length i, and it *odd* if i is odd and *even* if i is even. Therefore, we can partition the nodes of the graph into two sets

- $S_{\text{even}} = \{v : |\text{sp}(s, v)| \text{ is even.}\}$
- $S_{\text{odd}} = \{v : |\text{sp}(s, v)| \text{ is odd.}\}$

where $|\text{sp}(u, v)|$ denotes the length of the shortest path between s and v in G. In other words, S_{even} contains the nodes for which the shortest path to s has even length and S_{odd} contains the nodes for which the shortest path to s has odd length. By definition, it holds that $S_{\text{even}} \cup S_{\text{odd}} = \emptyset$ which implies that $S_{\text{even}} \cup S_{\text{odd}} = V$.

Now consider two vertices $u_1, u_2 \in S_{\text{even}}$ and consider the path $p(u_1, u_2)$ in the BFS tree between them - this path is unique since the BFS tree is a spanning tree. Since edges on this path alternate between levels (e.g., they start from an odd level, then go to an even level, then to an odd level and so on), a path that starts and ends with a vertex in S_{even} must have even length. If there was an edge $e = (u_1, u_2)$ in the graph G, that would mean that $p(u_1, u_2)$ together with e would form a cycle of odd length in the graph G. Therefore, there can not be an edge between any two vertices in S_{even}. With a completely symmetric argument, we can establish that there can not be an edge between any two vertices in S_{odd} either. But then the graph $G = (V, E) = (S_{\text{even}} \cup S_{\text{odd}}, E)$ is bipartite.

Problem 2

Prove that a connected graph with n nodes has at least $n - 1$ edges.

Solution

We will prove the statement by induction on the number of nodes n of the graph.

Base Case: For $n = 2$, the graph is only connected if there is an edge between the two nodes, therefore there exists at least 1 edge.

Induction Step: Assume that any connected graph with $n - 1$ nodes ($n > 1$) have at least $n - 2$ edges (**Induction Hypothesis**). We will prove that all connected graphs G of n nodes have at least $n - 1$ edges.

Since G is connected, then every vertex u must be incident to some edge $e = (u, v)$ for some node $v \neq u$. Assume by contradiction that the number of edges m of G is at most $n-2$. This means that the total degree $\sum_{i=1}^{n} \deg(u_i)$ is at most $2(n-2) < 2n$, which follows from the fact that every edge can be incident to at most two nodes. Therefore, we obtain that there exists a node $v \in V$ that has degree $\deg(v) = 1$.

Now assume that we remove v from the graph G, together with the edge that v is incident to. The graph is still connected and it now has $n-1$ nodes and $n-3$ edges. However, this contradicts the Induction Hypothesis, that every connected graph with $n-1$ nodes has at least $n-2$ edges.

Problem 3

Prove the following property for the layers produced by BFS: For any edge (u, v), either u and v are in the same layer, or $|L(u) - L(v)| = 1$, where $L(x)$ is the layer of node x.

Solution

We will use proof by contradiction. Suppose that there exists any edge $e = (u, v)$ such that u is in some layer i and v is in some layer j, such that $j > i + 1$. Since u is in a layer with a smaller index, it was obviously explored first in the operation of BFS, at step i. In the next step, BFS explores all the neighbours of u that have not been explored in previous steps. Since v is a neighbour of u and it was not explored in step $i+1$, it must have been explored before step $i+1$. However, since v has a label $j > i+1$, it must have been explored in step j, and we obtain a contradiction.

Problem 4

Let $G = (V, E)$ be a connected graph and let $s \in V$ be a node of G. Suppose that we run $\text{DFS}(G, s)$ and obtain a DFS spanning tree T and that we also run $\text{BFS}(G, s)$ and obtain the same BFS spanning tree T. Prove that $G = T$.

Solution

We will use proof by contradiction. Suppose that $G \neq T$. This means that there exists an edge $e = (u, v)$ in G, which is not an edge of T. Since e is not part of the DFS tree, it must be a *back edge*. This means that one of the two must be an ancestor of the other in the way the DFS tree is generated (see also Statement 3.7., page 85 of Kleinberg and Tardos - Algorithm Design). Since $e = (u, v)$ does not belong to T, this implies that the $|d(s, v) - d(s, u)| \geq 2$. In turn, this means that v and u are in levels of the BFS tree that are neither consecutive, nor the same. However, this is not possible by the statement of Problem 3, a contradiction.

Problem 5

A *Hamiltonian path* in a DAG G, is a path that visits all the nodes of the graph *exactly once*. Prove that a Hamiltonian path in a DAG G exists if any only if G has a *unique* topological order.

Solution

First we prove the forward direction. Assume that for graph G, there exists a Hamiltonian path $(u_1, u_2, \ldots, u_n)$. Obviously, $T_1 = u_1, u_2, \ldots, u_n$ is a valid topological ordering, since by the existence of the path, we know that there exist edges (u_i, u_{i+1}) for any $i = 1, \ldots, n-1$ in the sequence. Assume that there is a different topological ordering T_2 which places some u_j before u_i for $j > i$. This implies that in the graph G, there exists a path from u_j to u_i. But we already know that there exists a path from u_i to u_j (a subpath of the Hamiltonian path), contradicting the fact that G is a DAG. (Alternatively, one can view it as follows: if there is a different topological order, then some node u_j will have an edge "back" to some node u_i with $i < j$, and we would have a cycle).

Now we prove the reverse direction. Assume that for graph G, there exists a *unique* topological ordering $T_1 = u_1, u_2, \ldots, u_n$. We claim that there must be an edge between any pair of consecutive nodes u_i, u_{i+1} for $i = 1, \ldots n-1$. Indeed, if that was not the case, then we could swap u_i and u_{i+1} in the ordering without affecting the relative order of any other pair of nodes, and the resulting ordering would be a topological ordering. Therefore, a unique topological ordering implies a Hamiltonian path.