Exercises: Discrete Maths

In the following for a graph $G = (V, E)$, V will represent the set of vertices and E the set of edges, and $|V| = n$ and $|E| = m$. Unless we explicitly state otherwise, a graph will always be undirected.

Exercise 1. Prove that in any connected graph with $n \geq 2$ there exists two vertices having exactly the same degree.

Exercise 2. Given a directed graph G write an algorithm that calculates the out-degree of each vertex assuming that the graph is represented using: (a) Adjacency Lists, (b) Adjacency Matrix. Analyse the complexity of your algorithm in both of the cases.

Exercise 3. Given a graph $G = (V, E)$ and an edge $e \in E$ prove that it is always possible to find a spanning tree T of G that contains e.

Exercise 4. Given a graph $G = (V, E)$ and a vertex $r \in V$, let T be a spanning tree of G obtained by a DFS starting from vertex r. Let $A \subseteq V$ be a set of vertices such that for any two vertices u, v in A, the paths in T, from u to r and from v to r have the same length. Prove that there cannot be two vertices u, v in A such that $\{u, v\} \in E$ (in other words the vertices in A form an independent set in G).

Exercise 5. Prove that BFS and DFS algorithms on a connected graph $G = (V, E)$, produce the same tree if and only if G is a tree.

Exercise 6. A *bipartite* graph $G = (V, E)$ is a graph whose set of vertices V can be partitioned into two sets V_1, V_2, with $V_1 \cup V_2 = V$ and $V_1 \cap V_2 = \emptyset$, such that $\{u, v\} \in E$ implies $u \in V_1, v \in V_2$ or $u \in V_2, v \in V_1$ (in other words V_1 and V_2 form two independent sets).

a) Prove that a graph G is bipartite if and only if it does not contain a cycle of odd length.

b) Describe an algorithm which uses the BFS algorithm to determine whether or not a given connected graph $G = (V, E)$ is bipartite. Explain why your algorithm is correct and describe its running time.

Exercise 7. The *distance* between two vertices in a graph is the length of the shortest path connecting them. The *diameter* of a connected graph G is the maximum distance between all the pairs of vertices of the graph. The *height* of a rooted tree T is the length of a longest path from the root to a leaf (or equivalently the maximum level of any vertex in the tree).

a) Describe an algorithm of complexity $O(nm)$, that given a connected graph $G = (V, E)$ determines its diameter.

b) Prove or give a counterexample to the following claim: "There exists a connected graph whose diameter is equal to 10 and has a BFS tree of height 4".
If a graph G has a BFS tree of height h, what is the largest value for its diameter?

c) Given a connected graph $G = (V, E)$ prove that for any vertex $s \in V$ the height of $DFS(G, s)$ tree is always larger than or equal to the height of $BFS(G, s)$.

Exercise 8. Given a connected graph $G = (V, E)$, describe an algorithm that determines the length of the smallest cycle in G (if the graph has no cycles then the algorithm should output no cycle otherwise it must output the length). Describe the running time of your algorithm.

Exercise 9. Given a directed graph $G = (V, E)$ and a vertex $s \in V$, describe an algorithm of complexity $O(m)$ that determines the number of vertices that are reachable from s and have the maximum distance from s.

Exercise 10. Given a connected graph $G = (V, E)$ and two subsets V_1, V_2 of V, we define the *distance* between V_1 and V_2 as the minimum distance between a vertex in V_1 and a vertex in V_2. In the case V_1 and V_2 are not disjoint their distance is equal to 0. Describe an algorithm of complexity $O(n + m)$ that given G, V_1, V_2 determines the distance between V_1 and V_2.

Exercise 11. Given a directed graph G whose vertices are coloured with red or green, describe an algorithm that determines whether G has a directed cycle whose vertices are all green. Analyse the complexity of your algorithm.