COMP4500/7500 Advanced Algorithms & Data Structures
Tutorial Exercises 4 (2014/2)*

School of Information Technology and Electrical Engineering, University of Queensland

August 18, 2014

This material aims to familiarise you with graph representations and algorithms, A good treatment of elementary graph algorithms may be found in CLRS Chapter 22 [3rd, 2nd]; CLR Chapter 23 [1st].

1. (See CLRS Exercise 22.1-6, p593 [3rd], p530 [2nd], CLR Exercise 23.1-6, p468 [1st])
   When an adjacency-matrix representation is used, most graph algorithms require time $\Omega(|V|^2)$, but there are some exceptions. Show how to determine whether a directed graph contains a ***universal sink*** — a vertex with in-degree $|V| - 1$ and out-degree 0 — in time $\Theta(|V|)$, given an adjacency-matrix representation for G.

   In the adjacency-matrix representation of G, an entry $G.\mathit{edge}(u, v)$ being TRUE corresponds to an edge from u to v. A vertex v is a universal sink if row v is all FALSE (out-degree 0) and column v is all TRUE except for the entry in row v. There can be at most one universal sink in a graph.

2. (CLRS Exercise 22.2-4; CLR Exercise 23.2-4)
   Argue that in a breadth-first search, the value of $v.d$ assigned to a vertex v is independent of the order in which the vertices in each adjacency list are given.

3. (CLRS Exercise 22.4-3; CLR Exercise 23.4-3)
   Give an algorithm that determines whether or not a given ***undirected*** graph $G = (V, E)$ contains a cycle. Your algorithm should run in $O(|V|)$ time, independent of $|E|$.

4. (CLRS Exercise 22.3-11; CLR Exercise 23.3-9)
   Show that a depth-first search of an undirected graph G can be used to identify the connected components of G, and that the depth-first forest contains as many trees as G has connected components. More precisely, show how to modify depth-first search so that each vertex v is assigned an integer label $v.\mathit{comp}$ between 1 and k, where k is the number of connected components of G, such that $u.\mathit{comp} = v.\mathit{comp}$ if and only if u and v are in the same connected component.

5. (CLRS Exercise 22.2-7; CLR Exercise 23.2-7)

   The *diameter* of a tree $T = (V, E)$ is given by

   $$\max_{u,v \in V} \delta(u, v),$$

   that is, the diameter is the largest of all shortest-path distances in the tree. Give an efficient algorithm to compute the diameter of a tree, and analyse the running time of your algorithm. Assume that the tree is represented as an undirected graph and that you are given an algorithm that performs a breadth-first search to find the distance from a single node r to all other nodes.