

## Homework Assignment #7

### Due Time/Date

2:20PM Tuesday, November 19, 2024. Late submission will be penalized by 20% for each working day overdue.

### How to Submit

Please write or type your answers on A4 (or similar size) paper. Put your completed homework on the table in front of the instructor's desk before the class on the due date starts. For early or late submissions, please drop them in Yih-Kuen Tsay's mail box on the first floor of Management College Building 2. You may discuss the problems with others, but copying answers is strictly forbidden.

### Problems

There are five problems in this assignment, each accounting for 20 points.

1. (7.23) Describe an efficient implementation of the algorithm discussed in class (as by-product of an inductive proof) for finding an Eulerian circuit in a graph. The algorithm should run in linear time and space. (Hint: try to interweave the discovery of a cycle and that of the separate Eulerian circuits in the connected components with the cycle removed, in the induction step.)
2. (7.28) A **binary de Bruijn sequence** is a (cyclic) sequence of  $2^n$  bits  $a_1a_2 \cdots a_{2^n}$  such that each binary string  $s$  of size  $n$  is represented somewhere in the sequence; that is, there exists a unique index  $i$  such that  $s = a_i a_{i+1} \cdots a_{i+n-1}$  (where the indices are taken modulo  $2^n$ ). For example, the sequence 11010001 is a binary de Bruijn sequence for  $n = 3$ . Let  $G_n = (V, E)$  be a directed graph defined as follows. The vertex set  $V$  corresponds to the set of all binary strings of size  $n-1$  ( $|V| = 2^{n-1}$ ). A vertex corresponding to the string  $a_1a_2 \cdots a_{n-1}$  has an edge leading to a vertex corresponding to the string  $b_1b_2 \cdots b_{n-1}$  if and only if  $a_2a_3 \cdots a_{n-1} = b_1b_2 \cdots b_{n-2}$ . Prove that  $G_n$  is a directed Eulerian graph, and discuss the implications for de Bruijn sequences.
3. In the topological sorting algorithm that we discussed in class for directed acyclic graphs, DFS is used to calculate the indegree of each vertex in the input graph. Please give a detailed description of this calculation in adequate pseudocode. You need to define a main routine which invokes the DFS procedure with suitable preWORK and postWORK.
4. (7.3) Given as input a connected undirected graph  $G$ , a spanning tree  $T$  of  $G$ , and a vertex  $v$ , design an algorithm to determine whether  $T$  is a valid DFS tree of  $G$ .

rooted at  $v$ . In other words, determine whether  $T$  can be the output of DFS under some order of the edges starting with  $v$ . The running time of the algorithm should be  $O(|V| + |E|)$ .

5. (7.38) Given a directed acyclic graph  $G = (V, E)$ , find a simple (directed) path in  $G$  that has the maximum number of edges among all simple paths in  $G$ . The algorithm should run in linear time.