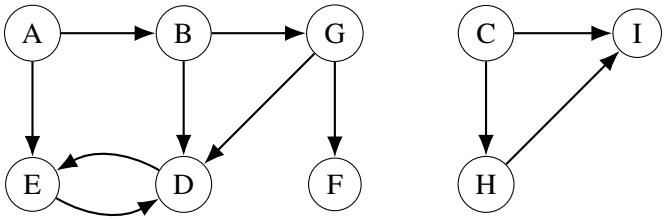
CMPSC 465 Data Structures & Algorithms
Fall 2025 Ke Chen and Yana Safonova

Worksheet 4

Monday, Sep 29, 2025

1. **Graph Basics.** Run DFS on the following graph, visit nodes alphabetically (e.g. given a choice between nodes D and F, visit D first).



   (a) List the nodes in the order you visit them (so each node should appear in the ordering exactly once).

   (b) List each node with its pre- and post-number. The numbering starts from 1 and ends at 18.

   (c) Label each edge as **T**ree, **B**ack, **F**orward or **C**ross.

2. **Award Ceremony.** Your job is to prepare a lineup of n awardees at an award ceremony. You are given a list of m constraints of the form: awardee i wants to receive his award before awardee j. Design an algorithm to either give such a lineup that satisfies all constraints, or return that it is not possible. Your algorithm should run in $O(m+n)$ time.

3. **Bipartite Graph.** You are given an undirected graph $G = (V,E)$. Design an algorithm to determine if G is bipartite, i.e., its vertices can be colored with two colors such that every edge has endpoints of different colors. If it is bipartite, return such a coloring, otherwise return that it is not possible. Your algorithm should run in $O(|V|+|E|)$ time..

4. **Pouring Water.** We have three containers whose sizes are 10 pints, 7 pints, and 4 pints, respectively. The 7-pint and 4-pint containers start out full of water, but the 10-pint container is initially empty. We are allowed one type of operation: pouring the contents of one container into another, stopping only when the source container is empty or the destination container is full. We want to know if there is a sequence of pourings that leaves exactly 2 pints in the 7- or 4-pint container.

   1. Model this as a graph problem: give a precise definition of the graph involved and state the specific question about this graph that needs to be answered.
   2. What algorithm should be applied to solve the problem?
   3. Find the answer by applying the algorithm.

5. **DFS.** You are given a binary tree $T = (V,E)$ (in adjacency list format), along with a designated root node $r \in V$. Recall that u is said to be an ancestor of v in the rooted tree, if the path from r to v in T

passes through u. You wish to preprocess the tree so that queries of the form "is u an ancestor of v?" can be answered in constant time. The preprocessing itself should take linear time. How can this be done?

6. **DFS.** You are given a binary tree $T = (V, E)$ with a designated root node. In addition, there is an array $x[\cdot]$ with a value for each node in V. Define a new array $z[\cdot]$ as follows:
for each $u \in V$,
$z[u] =$ the maximum of the x-values associated with u's descendants.
Give a linear-time algorithm that calculates the entire z-array.