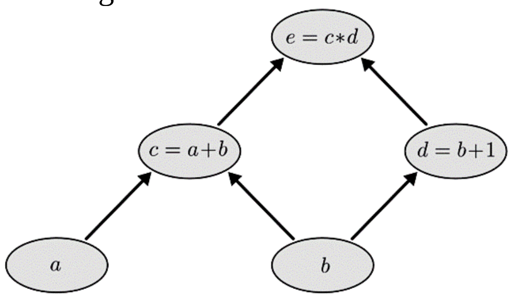
**Habib University - City Campus**
**Course Title:** Algorithms: Design and Analysis
**Instructors' names:** Dr. Shah Jamal Alam and Dr. Ayesha Enayat.
**Class ID:** CS 412 (all sections)
**Examination:** Midterm Exam **Exam Date:** 9-March-2024

**Attempt any 13 points (out of 15). Duration: 90 minutes.**

**Instructions for students:**

- The question paper must be returned at the end of the exam along with the answer script.
- You are allowed to bring one A4 handwritten sheet [no photocopies or typed material will be allowed] and a scientific calculator. No other electronic gadgets are allowed
- Use only a black or blue pen. Use pencils only for sketches [if needed] or rough work.
- You may attempt the questions in any order.
- All answers must be given on the answer script.

**Q.1. [1 point]** Consider the following dependency graph with nodes representing statements from a code snippet. *Linearize* the graph. Show the working.




**Q.2. Choose True/False from the following. Justify your choice in one to two sentences only.**

**A. [0.5 points]** A depth-first search of a directed graph always produces the same number of tree edges (i.e., independent of the order in which the vertices are provided and independent of the order of the adjacency lists).

**B. [0.5 points]** $n! = o(n^n)$.

**C. [0.5 points]** A directed, acyclic graph with $n$ vertices can have at most $^n/_2$ number of strongly connected components.

**D. [0.5 points]** $\log_2 \prod_{i=0}^{n} p(i) = O\left(n \cdot \underbrace{max}_{i} \{\log p(i)\}\right)$

**Q.3. Designing divide-and-conquer algorithms:**

**A. [1.5 points]** Write a pseudocode of an efficient algorithm to compute the powers of a given number $n$ ($n \epsilon \mathbb{N}$). Argue on the runtime complexity of your algorithm.

**B. [1.5 points]** Given a set $P$, of $n$ teams in some sport (say, football), a round-robin tournament is a collection of games in which each team plays each other exactly once. Such round-robin tournaments are often used as the first round for establishing the order of teams (and their seedings) for later single- or double-elimination tournaments. Design an efficient algorithm for constructing a round-robin tournament for the set $P$, of $n$ teams assuming $n$ is in the exact powers of 2.

**C. [1.5 points]** We are given a set $S$, of $n$ integers, and want to determine whether any two (2) of the contained numbers add up to a given input integer, $x$. Design an efficient algorithm to solve this problem with a worst-case time complexity of $O(n \lg n)$.

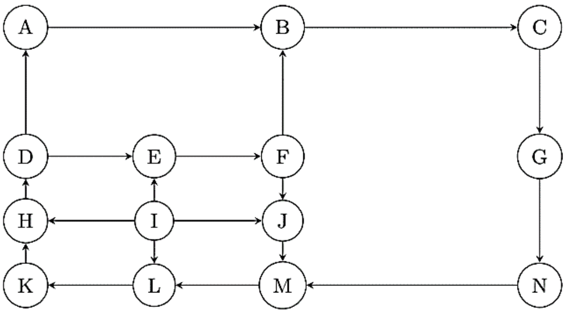
**Q.4. Solve the following recurrences:**

A. **[0.5 points]** $T(n) = 5\,T(n-1) + 6\,T(n-2)$. Give the general solution of the above recurrence.

B. **[1 point]** $g(n) = 7g(^n/_3) + \sqrt[3]{2n} - 5$, with $g(1) = 1$. State the solution in asymptotic bounds.

C. **[0.5 points]** $T(n) = T(^n/_3) + T(^{2n}/_3) + \Theta(n)$. Draw the recurrence tree to get the solution in the upper bound.

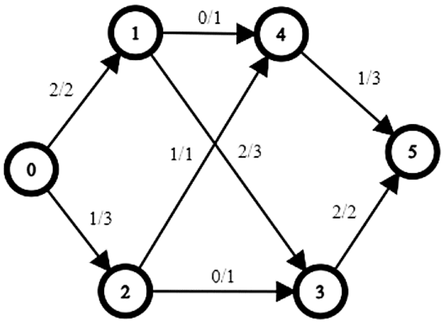
**Q.5. [1 point]** An organization is making a serious effort to implement a flat-hierarchy structure, i.e., any two offices within the organization ultimately report to each other. However, the planning has gone awry in some instances and only certain groups are part of the flat hierarchies. For example, consider the following directed graph, where each labelled node represents an office in the organization and the directed edge represents the relation *reports-to*.




Describe an effective general strategy (state the pseudocode) to identify such groups in a general, directed graph. Argue about the runtime complexity of this strategy.

**Q.6. [1. 5 points]** Suppose that an array contains $n$ numbers, each of which is $-1$, $0$, or $1$. Give a pseudocode of an algorithm that sorts this array in linear time. How does the algorithm's linear time complexity compare to the $\Omega(n \lg n)$ lower bound discussed in the class? Does this imply that the algorithm is more efficient than the lower bound for sorting algorithms that rely on comparisons?

**Q.7. [1 point]** Consider the following directed graph. What is the current flow? Is it also the maximum flow? Justify your answer as formally as possible.




**Q.8. [1 point]** Given that a strongly connected component contains an edge $(u, v)$ such that $u.f > v.f$, what insight does this provide about the order of the traversal of the edge? State your answer in one to two sentences only.

**Q.9. [1 point]** Most graph algorithms that take an adjacency-matrix representation as input require time $\Omega(V^2)$, but there are some exceptions. Explain how to determine whether a directed graph $G$ contains a universal sink [a vertex with an in-degree $|V| - 1$ and out-degree 0] in time $O(V)$, given an adjacency matrix representation for $G$.