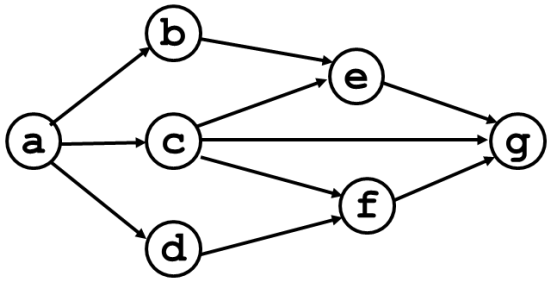
Department of Computer Science and Engineering
National Sun Yat-sen University
Advanced Programming and Practice - Final Exam., June 22, 2017

1. Explain each of the following terms. (20%)
   (a) Eulerian cycle
   (b) lexicographic order
   (c) branch and bound
   (d) longest common subsequence problem
   (e) matrix-chain multiplication problem

2. (a) Please give two answers for the *topological order* in the following acyclic digraph. (4%)
   (b) Design an algorithm to find one *topological order* of a given acyclic digraph. (9%)



3. Suppose we obtain the recurrence formula of time complexity for solving a problem with a *divide-and-conquer* method as follows:
   $T(n)=b$, if $n \leq 2$
   $T(n)=2T(n/2)+cn$, if $n > 2$,
   where n is the input size of the problem, b and c are constants. Please derive the time complexity and represent it with O notation. (12%)

4. In the 0/1 *knapsack* problem, we are given n objects, each with its weight w_i and profit p_i, for $1 \leq i \leq n$, and the capacity M. Please derive the *dynamic programming* formula for solving the problem. (10%)

5. (a) What data structures should be used in the *depth-first search*, *breadth-first search* and *best-first search* methods, respectively? (9%)
   (b) Explain the *hill climbing* method for searching the solution of a given problem. (6%)

6. In the *activity selection* problem, we are given n activities, $A = \{1, 2, \ldots, n\}$, each with a start time s_i (positive integer) and a finish time f_i (positive integer), $s_i \leq f_i$, where activity i occupies time interval $[s_i, f_i]$. Activity i and activity j are

compatible if $s_i \geq f_j$ or $s_j \geq f_i$. The problem is to select a maximum-size set of mutually compatible activities. Please design an algorithm for solving this problem and analyze the time complexity. (15%)

7. In the *range minimum query* (RMQ) problem, we are given a list of integers $T=<t_1, t_2, \ldots, t_n>$, where the elements in T are not sorted. The problem asks to construct a data structure in the preprocessing stage such that the minimum of a range can be answered efficiently in the query stage. For example, suppose $T=<5, 3, 10, 7, 8, 5, 9, 4>$, $n=8$. The answer of RMQ for the range [2, 5] is 3, that is equivalent to find the minimum of 3, 10, 7, 8 (t_2, t_3, t_4, t_5). As more examples, the answers of RMQ for [1, 5], [3, 5] and [5, 7] are 3, 7 and 5, respectively. Please design an algorithm for solving this problem such that the preprocessing stage requires $O(n)$ time and each query requires $O(\log n)$ time. You have to analyze the time complexity of your algorithm. (15%)