Final

Note

This is a closed-book exam. Each problem accounts for 10 points, unless otherwise marked.

Problems

1. Let $x_1, x_2, \cdots, x_n$ be a sequence of real numbers (not necessarily positive). Design an $O(n)$ algorithm to find the subsequence $x_i, x_{i+1}, \cdots, x_j$ (of consecutive elements) such that the product of the numbers in it is maximum over all consecutive subsequences. The product of the empty subsequence is defined to be 1. Please present your algorithm in an adequate pseudo code and make assumptions wherever necessary. *Explain why the algorithm is correct.*

2. Given two strings $A = ababb$ and $B = acabbb$, compute the minimal cost matrix $C[0..5, 0..6]$ for changing A character by character to B. Please describe also the detail of calculation for the entry $C[5, 6]$.

3. Give a binary de Bruijn sequence of 2^4 bits, which is a (cyclic) sequence of 2^4 bits $a_1 a_2 \cdots a_{2^4}$ such that each binary sequence of size 4 appears somewhere in the sequence. Explain how you can systematically produce the sequence.

4. 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. Please present your algorithm in an adequate pseudo code and make assumptions wherever necessary. Explain why the algorithm is correct and give an analysis of its time complexity. The more efficient your algorithm is, the more points you get for this problem.

5. What is wrong with the following algorithm for computing the minimum-cost spanning tree of a given weighted undirected graph (assumed to be connected)?

   If the input is just a single-node graph, return the single node. Otherwise, divide the graph into two subgraphs and recursively compute their minimum-cost spanning trees. Finally, connect the two trees with an edge between the two subgraphs that has the minimum weight.

6. Let $G = (V, E)$ be a connected weighted undirected graph and T be a minimum-cost spanning tree (MCST) of G. Suppose that the cost of one edge $\{u, v\}$ in G is *decreased*; $\{u, v\}$ may or may not belong to T. Design an algorithm either to find a new MCST or to determine that T is still an MCST. The more efficient your algorithm is, the more points you will be credited for this problem. Explain why your algorithm is correct and analyze its time complexity.

7. Below is the main procedure for determining the biconnected components of an undirected graph. Is the algorithm still correct if we replace the last second line "$v.high := \max(v.high, w.DFS\_Number)$" by "$v.high := \max(v.high, w.high)$"? Why? Please explain.

procedure BC(v);
begin
  $v.DFS\_Number := DFS\_N$;
  $DFS\_N := DFS\_N - 1$;
  $v.high := v.DFS\_Number$;
  for all edges (v, w) **do**
    if w is not the parent of v **then**
      insert (v, w) into *Stack*;
      if $w.DFS\_Number = 0$ **then**
        $BC(w)$;
        if $w.high \leq v.DFS\_Number$ **then**
          remove all edges from *Stack*
            until (v, w) is reached;
        $v.high := \max(v.high, w.high)$
      else
        $v.high := \max(v.high, w.DFS\_Number)$
end

8. Finding a minimum-size vertex cover for a given graph in general is hard. However, the same problem is easier for trees. Describe an efficient algorithm for finding a minimum-size vertex cover for a given tree. Please present your algorithm in an adequate pseudo code and make assumptions wherever necessary. Explain why your algorithm is correct and analyze its time complexity. The more efficient your

algorithm is, the more points you get for this problem. (Hint: how would you cover an edge that connects a leaf to its parent? Think inductively!)

9. Solve the single-source shortest path problem using the dynamic programming approach (which we have described in class). You need only to define precisely a recurrence relation. Please explain why (or why not) your solution allows edges with a negative weight.

10. Solve one of the following two problems. (Note: If you try to solve both problems, I will randomly pick one of them to grade.)

    (a) The traveling salesman problem is as follows.

    > Given a weighted complete graph $G = (V, E)$ (representing a set of cities and the distances between all pairs of cities) and a number D, does there exist a circuit (traveling-salesman tour) that includes all the vertices (cities) and has a total length $\leq D$?

    Prove that the traveling salesman problem is NP-complete.

    (b) The subgraph isomorphism problem is as follows.

    > Given two graphs $G = (V_1, E_1)$ and $H = (V_2, E_2)$, does G have a subgraph that is isomorphic to H? (Two graphs are isomorphic if there exists a one-one correspondence between the sets of vertices of the two graphs that preserve adjacency.)

    Prove that the subgraph isomorphism problem is NP-complete.

Appendix

- A *vertex cover* of a graph G is a set of vertices such that every edge in G is incident to at least one of these vertices.

- The Hamiltonian cycle problem: given a graph G, does G have a Hamiltonian cycle? (A Hamiltonian cycle in a graph is a cycle that contains each vertex, except the starting vertex of the cycle, exactly once.)
  The problem is NP-complete.