Final

Note

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

Problems

1. Compute the *next* table as in the KMP algorithm for the string $B[1..11] = abaababaaab$. Please show how $next[10]$ and $next[11]$ are computed from using preceding entries in the table.

2. Given two strings $A = abaab$ and $B = baabbb$, 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. Below is an algorithm skeleton for depth-first search utilizing a stack; assume that the input graph is undirected and connected. Modify the algorithm so that it prints out (the edges of) a DFS tree of the input graph. You should try not to change the overall structure of the original algorithm.

```
Algorithm Simple_Nonrecursive_DFS (G, v);
begin
    push v to Stack;
    while Stack is not empty do
        pop vertex w from Stack;
        if w is unmarked then
            mark w;
            for all edges (w, x) such that x is unmarked do
                push x to Stack
end
```

4. 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.

5. Design an algorithm that, given a weighted directed graph, detects the existence of a negative-weight cycle (the sum of the weights of its edges is negative). Please present your algorithm in an adequate pseudo code and make assumptions wherever necessary. The more efficient your algorithm is, the more points you will be credited for this problem. Explain why your algorithm is correct and give an analysis of its time complexity.

6. 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, recursively compute their minimum-cost spanning trees, and then connect the two spanning trees with an edge between the two subgraphs that has the minimum weight.

7. Prove that if the costs of all edges in a given connected graph are distinct, then the graph has an unique minimum-cost spanning tree.

8. Let $G = (V, E)$ be a directed graph, and let T be a DFS tree of G. Prove that the intersection of the edges of T with the edges of any strongly connected component of G form a subtree of T.

9. Below is an algorithm discussed in class for determining the strongly connected components of a directed graph. Is the algorithm still correct if we replace the line "$v.high := \max(v.high, w.DFS\_Number)$" by "$v.high := \max(v.high, w.high)$"? Why? Please explain.

```
Algorithm Strongly_Connected_Components(G, n);
begin
   for every vertex v of G do
       v.DFS_Number := 0;
       v.component := 0;
   Current_Component := 0;  DFS_N := n;
   while v.DFS_Number = 0 for some v do
       SCC(v)
end

procedure SCC(v);
begin
   v.DFS_Number := DFS_N;
```

```
    DFS_N := DFS_N − 1;
    insert v into Stack;
    v.high := v.DFS_Number;
    for all edges (v, w) do
        if w.DFS_Number = 0 then
            SCC(w);
            v.high := max(v.high, w.high)
        else if w.DFS_Number > v.DFS_Number
                and w.component = 0 then
            v.high := max(v.high, w.DFS_Number)
    if v.high = v.DFS_Number then
        Current_Component := Current_Component + 1;
        repeat
            remove x from the top of Stack;
            x.component := Current_Component
        until x = v
end
```

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 hitting set problem is as follows.

    > Given a collection C of subsets of a set S and a positive integer k, does S contain a hitting set for C of size k or smaller, that is, a subset $S' \subseteq S$ with $|S'| \leq k$ such that S' contains at least one element from each subset in C?

    Prove that the hitting set problem is NP-complete.

    (b) 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.

Appendix

- The vertex cover problem: given an undirected graph $G = (V, E)$ and an integer k, determine whether G has a vertex cover containing $\leq k$ vertices. (A *vertex cover* of G is a set of vertices such that every edge in G is incident to at least one of these vertices.)
  The vertex cover problem is complete.

- The Hamiltonian cycle problem: given an undirected 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 Hamiltonian cycle problem is NP-complete.