Final

Note

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

Problems

1. Let $T(h)$ denote the number of nodes in a smallest AVL tree of height h (smallest in the sense of having the least number of nodes); the height of an empty tree is defined to be 0. Prove that $T(h) = F(h+2) - 1$, where $F(n)$ is the n-th Fibonacci number ($F(1) = 1$, $F(2) = 1$, and, for $n \geq 3$, $F(n) = F(n-1) + F(n-2)$).

2. Compute the *next* table as in the KMP algorithm for the string $B[1..10] = babbabbaba$. Please show how $next[9]$ and $next[10]$ are computed from using preceding entries of the table.

3. Find two application problems in the real world that can be modeled as graph problems and solved by graph algorithms. Please give informal yet precise descriptions of the original problems and then formalize them as graph problems. How many points you will be credited depends on how much the domains of the original problems are different from graphs in nature.

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. Consider the adapted Floyd's algorithm discussed in class for computing the transitive closure of a directed graph. If we exchange the first two lines of the program body, the algorithm becomes as follows (initialization omitted):

   Algorithm Probable_Transitive_Closure (A);
   begin
       for $x := 1$ to n **do**
           for $m := 1$ to n **do**
               for $y := 1$ to n **do**

if $A[x, m]$ and $A[m, y]$ **then** $A[x, y] := true$

end

Is the new algorithm **Probable_Transitive_Closure** correct? Justify your answer.

6. The celebrated Prim's algorithm we discussed in class for finding the minimum-cost spanning tree of a graph is based on the following property:

   > Suppose $G = (V, E)$ is a connected weighted graph. Let V_1 and V_2 be a partition of V and $E(V_1, V_2)$ be the set of edges connecting nodes in V_1 to nodes in V_2. Then, the edge with the minimum weight in $E(V_1, V_2)$ must be in the minimum-cost spanning tree of G.

   Prove the correctness of the above property.

7. 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 *increased*; $\{u, v\}$ 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.

8. A *bridge* in a connected graph is an edge whose removal will break the graph into two separate subgraphs. Design an algorithm for determining if a given graph contains a bridge. 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.

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) A variant of the Hamiltonian path problem is as follows.

    > Given a graph $G = (V, E)$ and $u, v \in V$, does G have a Hamiltonian path from u to v? (A Hamiltonian path in a graph is a simple path that contains each vertex exactly once.)

Prove that this variant of the Hamiltonian path 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

- 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.