Final

Note

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

Problems

1. We have discussed in class how to rearrange an array into a (max) heap using a bottom-up approach. Please present the approach in a suitable pseudocode.

2. Compute the *next* table as in the KMP algorithm for the string $B[1..12] = abbaaabbaaab$. Please show the details of calculation for $next[11]$ and $next[12]$.

3. Given two strings $A = aaabb$ and $B = acabbb$, compute the minimal cost matrix $C[0..5, 0..6]$ for changing A character by character to B. Show the detail of calculation for the entry $C[5, 6]$.

4. Consider a chain A_1, A_2, A_3, A_4, A_5 of five matrices with dimensions 30×20, 20×30, 30×40, 40×60, and 60×20, respectively. Compute (by immitating an algorithm based on dynamic programming) the minimum number of scalar multiplications needed to evaluate the product $A_1A_2A_3A_4A_5$.

5. 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. The more efficient your algorithm is, the more points you get for this problem. Explain why the algorithm is correct and give an analysis of its time complexity.

6. Give a binary de Bruijn sequence of 2^4 bits. Explain how you can systematically produce the sequence.

7. Consider Dijkstra's algorithm for single-source shortest paths as shown below. You may find in the literature two bounds, namely $O(|V|^2)$ and $O((|E| + |V|) \log |V|)$, for its time complexity. Why is this so?

Algorithm Single_Source_Shortest_Paths (G, v);
begin
    for all vertices w **do**
        $w.mark := \mathit{false}$;
        $w.SP := \infty$;
    $v.SP := 0$;
    while there exists an unmarked vertex **do**
        let w be an unmarked vertex such that $w.SP$ is minimal;
        $w.mark := \mathit{true}$;
        for all edges (w, z) such that z is unmarked **do**
            if $w.SP + \mathit{length}(w, z) < z.SP$ **then**
                $z.SP := w.SP + \mathit{length}(w, z)$
end

8. (a) The time complexity of Dijkstra's algorithm for single-source shortest paths depends on what data structure is used for selecting the next nearest vertex. Two commonly cited bounds are $O(|V|^2)$ and $O((|E| + |V|) \log |V|)$. Which is better? Why?
(b) Dijkstra's algorithm may be repeatedly applied for all-pairs shortest paths. When should it be chosen over Floyd's algorithm? Why?

9. 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 time complexity of your algorithm should be $O(|V| + |E|)$. Explain why your algorithm is correct and analyze its time complexity.

10. A Hamiltonian *cycle* in a directed graph is a simple (directed) *cycle* that includes every vertex of the graph exactly once; a Hamiltonian *path* is a simple *path* that includes each vertex exactly once. The **directed Hamiltonian cycle problem** is to determine whether a given directed graph contains a Hamiltonian cycle, while the **directed Hamiltonian path problem** is to determine whether a given directed graph contains a Hamiltonian path.

    Given that the directed Hamiltonian cycle problem is NP-complete, prove that the directed Hamiltonian path problem is also NP-complete.