CS 430 Introduction to Algorithms **Fall Semester, 2016**

Homework 4 version 1.1

Assigned: Oct. 17 *Due: Nov. 2*

Please respect the following guidelines for writing pseudocode:

1. C instructions are fine. But do not write object-oriented additions. Do not declare or use any class. Declare only procedures (if necessary) and explain in words what each procedure does, and what is the use of each parameter.
2. One instruction per line
3. Match the brackets with a horizontal line
4. Number your lines
5. Write down if your array is indexed $0 \ldots n-1$ or $1 \ldots n$.

Problem 1 Suppose you are given two sets A and B, each containing n positive integers. You can choose to reorder each set however you like. After reordering, let a_i be the i^{th} element of set A, and let b_i be the i^{th} element of set B. You then receive a payoff of $\prod_{i=1}^{n} a_i^{b_i}$. Give an efficient algorithm to find the reorderings that will maximize your payoff. Present the pseudocode. Prove that your algorithm maximizes the payoff, and state its running time.

Problem 2 What is the best way to multiply a chain of matrices with dimensions that are 9×6, 6×3, 3×21, 21×11, 11×5, and 5×50? Show your work.

Problem 3 Give a pseudopolynomial algorithm for KNAPSACK. Strive for running time of $O(nB)$, but make sure running time is polynomial in n and B. The KNAPSACK problem is defined as follows. An instance consists of n items $1, 2, \ldots, n$ where item i has size s_i and profit p_i, and a knapsack size B with $B \geq s_i$ for all $i = 1, 2, \ldots, n$. All the numbers are integers. A feasible solution consists of a subset Q of $\{1, 2, \ldots, n\}$ such that $\sum_{i \in Q} s_i \leq B$. The objective is to maximize the total profit of Q - that is $\sum_{i \in Q} p_i$.

Present the pseudocode, discuss correctness, and analyze the running time.

Problem 4 Problem 15-4 from the textbook ("Printing neatly"). It is Problem 15-2 from the second edition of Cormen. Present the pseudocode, discuss correctness, and analyze the running time. Polynomial time is required.