Homework Assignment #4

Due Time/Date

2:20PM Tuesday, October 3, 2023. Late submission will be penalized by 20% for each working day overdue.

How to Submit

Please write or type your answers on A4 (or similar size) paper. Drop your homework by the due time in Yih-Kuen Tsay's mail box on the first floor of Management College Building 2. You may discuss the problems with others, but copying answers is strictly forbidden.

Problems

There are five problems in this assignment, each accounting for 20 points. (Note: problems marked with "(X.XX)" are taken from [Manber 1989] with probable adaptation.)

1. (5.3 adapted) Consider algorithm *Mapping* (see notes/slides). The algorithm starts with a nonempty set A, of course, since a mapping from A to itself is considered. Is it possible that the set S will become empty at the end of the algorithm? Show an example, or prove that it cannot happen.

2. Design an efficient algorithm that, given a sorted array A of n integers and an integer x, determine whether A contains two integers whose sum is exactly x. Please present your algorithm in adequate pseudocode and make assumptions wherever necessary. Give an analysis of its time complexity. The more efficient your algorithm is, the more points you will be credited for this problem.

3. (5.7) Write a program (or modify the code discussed in class) to recover the solution (i.e., enumerate the elements in the solution) to a knapsack problem using the *belong* flag. You should make your algorithm as efficient as possible.

4. (5.17) The Knapsack Problem that we discussed in class is defined as follows. Given a set S of n items, where the ith item has an integer size $S[i]$, and an integer K, find a subset of the items whose sizes sum to exactly K or determine that no such subset exists.

   We have described in class an algorithm to solve the problem. Modify the algorithm to solve a variation of the knapsack problem where each item has an *unlimited* supply. In your algorithm, change the type of $P[i,k].belong$ into integer and use it to record the number of copies of item i needed.

5. (5.23) Write a non-recursive program (in suitable pseudocode) that prints the moves of the solution to the towers of Hanoi puzzle. The three pegs are respectively named A, B, and C, with n (generalizing the original eight) disks of different sizes stacked in decreasing order on peg A.