Analysis of Algorithms: Assignment 9

Due date: April 15 (Thursday)

Problem 1 (3 points)
Write pseudocode of an algorithm GREEDY-KNAPSACK(W, v, w, n) for the 0-1 Knapsack Problem, and give its running time. The arguments are an weight limit W, array of item values $v[1..n]$, and array of item weights $w[1..n]$. Your algorithm must use the greedy strategy described in class, and return the set of selected items.

Problem 2 (3 points)
Using Figure 17.4(b) in the textbook as a model, draw an optimal-code tree for the following set of characters and their frequencies:

a:1 b:1 c:2 d:3 e:5 f:8 g:13 h:21

Problem 3 (4 points)
Suppose that you drive along some road, and you need to reach its end. Initially, you have a full tank, which holds enough gas to cover a certain distance d.

The road has n gas stations, where you can refill your tank. The distances between gas stations are represented by an array $A[1..n]$, where $A[1]$ is the distance from the start to the first gas station, $A[2]$ is the distance from the first to the second station, $A[3]$ is that from the second to the third station, and so on. The last gas station is located exactly at the end of the road. You wish to make as few stops as possible along the way.

Give an algorithm CHOOSE-STOPS(d, A, n) that identifies all places where you have to refuel, and returns the set of selected gas stations. You may assume that, for each i, $A[i] \leq d$.

Problem 4 (bonus)
This problem is optional; if you solve it, then you get 2 bonus points toward your final grade for the course. You cannot submit this bonus problem after the deadline.

Suppose that the weights of all items in the 0-1 Knapsack Problem are integers, and the weight limit W is also an integer. Design an algorithm that finds a *globally optimal* solution, and give its time complexity in terms of the number of items n and weight limit W.