Assignment 11

R-5.1 Let S = {a, b, c, d, e, f, g} be a collection of objects with benefit-weight values as follows: a:(12,4), b:(10,6), c:(8,5), d:(11,7), e:(14,3), f:(7,1), g:(9,6). What is an optimal solution to the fractional knapsack problem for S assuming we have a knapsack that can hold objects with total weight 15? Show your work.

R-5.3 Suppose we are given a set of tasks specified by pairs of the start times and finish times as T = {(1,2),(1,3),(1,4),(2,5),(3,7),(4,9),(5,6),(6,8),(7,9)}. Solve the task scheduling problem for this set of tasks.

R-5-11 Solve Exercise R-5.1 above for the 0-1 Knapsack Problem.

R-5-12 Sally is hosting an Internet auction to sell n widgets. She receives m bids, each of the form "I want k_i widgets for d_i dollars," for i = 1, 2, …, m. Characterize her optimization problem as a knapsack problem. Under what conditions is this a 0-1 versus fractional problem?

After Lesson 11b on memoization, do the following and submit next week:
A. Based only on the characterizing equations (B[k,w]), give a recursive pseudo code algorithm for the 0-1 knapsack problem (do this from the equations and without looking at my solution in the notes), then memoize it so it is efficient. Compare your algorithm to the two given in the lecture notes (iterative dynamic programming version and recursive non-memoized algorithm) in terms of time and space complexity.

C-5.9 How can we modify the dynamic programming algorithm from simply computing the best benefit value for the 0-1 knapsack problem (like A above) to computing the assignment (subset) that gives the maximum benefit? Design a pseudo code algorithm to do the trace back through the 2-dimensional array as we described in the lecture.

B. Suppose we have a set of objects that have different sizes s_1, s_2, …, s_n, and we have some positive upper limit L. Design an efficient pseudo code algorithm to determine the subset of objects that produces the largest sum of sizes that is no greater than L. Hint: dynamic programming similar to 0-1 knapsack problem, except only size/weight and no benefit.