CS325 Winter 2017

This set of questions focus on dynamic programming.

1. Knapsack without repetitions. Consider the following knapsack problem:

   The total weight limit $W = 10$ and

Item	Weight	Value
1	6	\$30
2	3	\$14
3	4	\$16
4	2	\$9

   Solve this problem using the dynamic programming algorithm presented in class (Feb 7th). Please show the two dimensional table $L(w, j)$ for $w = 0, 1, ..., W$ and $j = 1, 2, 3, 4$.

2. Give a dynamic programming algorithm for the following task.

   Input: A list of n positive integers $a_1, a_2, \ldots, a_n$ and a number t.
   Question: Does some subset of the a_i's add up to t? (You can use each a_i at most once.)

   The running time should be $O(nt)$.

   Consider the subproblem $L(i, s)$, which returns the answer of "Does a subset of $a_1, ..., a_i$ sum up to s?". Below are some substeps that will help you develop your algorithm.

   a What are the two options we have regarding item i toward answering the subproblem $L(i, s)$?

   b For each of the option, how would it change the subproblem? More specifically, what happens to the target sum and what happens to the set of available integars?

   c Based on the answers to the previous two questions, write a recursive formula that expresses $L(i, s)$ using the solutions to smaller subproblems.

   d Provide pseudocode for the dynamic programming algorithm that builds the solution table $L(i, s)$ and returns the correct answer to the final problem.

   e Modify the pseudocode such that it will not only return the correct "yes", "no" answer, but also return the exact subset if the answer is "yes".

3. 6.2

4. 6.3

5. 6.8