CS 1510: Homework 11

John Hofrichter — jmh162@pitt.edu
Kyra F. Lee — kfl5@pitt.edu
Zach Sadler — zps6@pitt.edu

September 25, 2013

Dynamic Programming Problem 15

Algorithm

for l = 0 to W **do**
  maxValue[l] = 0
end for
for l = 0 to W **do**
  for k = 1 to n **do**
    if $0 < l - w_k$ AND $maxValue[l - w_k] + v_k > maxValue[l]$ **then**
      $maxValue[l] = maxValue[l - w_k] + v_k$
    end if
  end for
end for

Explanation

In our algorithm, we take in a set of n objects, each with weight and value. So for an object k, w_k is the weight and v_k is the value. In addition, we take in a maximum weight W, which can also be thought of as a maximum carrying capacity.

Our algorithm initializes an array of size W to zero, then solves the subproblems from smallest weight (0) to the largest weight allowed, W. For each weight l from 0 to W, we attempt to put in each object k from 1 to n. We look back in our array by the object's weight (unless this would yield a negative weight) and compare the current maximum value for weight l with

the maximum value of weight $l - w_k$ plus v_k. If adding this object yields a larger max value for l then we update the maximum value for l.
Thus we solve the problem from bottom up, so that $maxValue[W]$ has our answer, and $maxValue[x]$ contains the maximum possible value for carrying capacity $0 \leq x \leq W$.
Since we have two for loops- one of length W and the other length n- and we do $O(1)$ work inside the inner for loop, our total runtime is polynomial in n and W.