Lindsey Bieda and Joe Frambach
Dynamic Programming Problems
9.28.2011

13. Our goal is now to consider the Knapsack problem, and develop a method for computing the actual items to be taken in $O(L)$ space and $O(nL)$ time.

(a) Consider the following problem. The input is the same as for the knapsack problem, a collection of n items $I_1, \ldots, I_n$ with weights $w_1, \ldots, w_n$, and values $v_1, \ldots, v_n$, and a weight limit L. The output is in two parts. First you want to compute the maximum value of a subset S of the n items that has weight at most L, as well as the weight of this subset. Let us call this value and weight v_a and w_a. Secondly for this subset S you want to compute the weight and value of the items in $\{I_1, \ldots, I_{n/2}\}$ that are in S. Let use call this value and weight v_b and w_b. So your output will be two weights and two values. Give an algorithm for this problem that uses space $O(L)$ and time $O(nL)$.

(b) Explain how to use the algorithm from the previous subproblem to get a divide and conquer algorithm for finding the items in the Knapsack problem a and uses space $O(L)$ and time $O(nL)$.

16. Give an algorithm for the following problem whose running time is polynomial in $n + W$:
Input: positive integers $w_1, \ldots, w_n$, $v_1, \ldots, v_n$ and W.
Output: The maximum possible value of $\sum_{i=1}^{n} x_i v_i$ subject to $\sum i = 1^n x_i w_i \leq W$ and each x_i is a nonnegative integer.

17. Give an algorithm for the following problem whose running time is polynomial in $n + L$, where $L = \max\ (\sum_{i=1}^{n} v_i^3, \prod_{i=1}^{n} v_i)$.
Input: positive integers $v_1, \ldots, v_n$
Output: A subset S of the integers such that $\sum_{v_i \in S} v_i^3 = \prod_{v_i \in S} v_i$.