

Programming Assignment 3

Winter 2014 (Due March 9, 2014)

Be sure to read the Programming Style document when turning in your assignment.

The 0/1 Knapsack problem (presented in class and in Chapter 16 pg. 425-426) is a classic problem in Computer Science that has applications not only in Operations Research, but in Electric Power Management, VLSI layout, Pattern Layout in Manufacturing, Process Scheduling and Computer Networking. The idea is to load a box of fixed capacity with containers of known weights and profits such that the profit is maximized and the box is not overloaded.

The 0/1 Knapsack problem is NP-Complete so we do not expect a low-cost solution to the problem. An exact solution is possible through Dynamic Programming and an approximate solution technique is presented in Chapter 16 by sorting the items in nonincreasing profit/weight density. You must use your implementation of Heapsort, Mergesort, or Insertion Sort (choose and justify your choice as the best one) for this sorting algorithm with the appropriate assert statements embedded.

In this project you will implement the dynamic programming solution to 0/1 Knapsack and a greedy solution to 0/1 Knapsack and compare the quality of their solutions and the run-time cost of their solutions. In short, this assignment requires you to implement

1. Greedy solution to the 0/1 Knapsack
2. Implement the recurrence given in class for a dynamic programming solution to the 0-1 knapsack
3. Implement Exercise 16.2-2 as an iterative integer dynamic programming solution.

To ensure a reasonable comparison, make sure that the capacity of the box is large enough ($>2n$). As always, compare the experimental runtime with the $O()$ projections and determine n_0 for each. You must work on your own and this must be your own work.