Albert-Ludwigs-Universität, Inst. für Informatik
Prof. Dr. Fabian Kuhn
M. Ahmadi, A. R. Molla, O. Saukh November 12, 2015

Algorithm Theory, Winter Term 2015/16
Problem Set 4

hand in (hard copy or electronically) by 10:15, Thursday November 19, 2015, tutorial session will be on November 23, 2015

Exercise 1: Filling Two Knapsacks (6 points)

In this problem, we consider a variation of the knapsack problem, where we have two instead of only one knapsack. Formally, we have items $1, \dots, n$ and each item i has a positive integer *weight* $w_i \in \mathbb{N}$ and a positive *value* $v_i > 0$. Further, we have two knapsacks of capacities W_1 and W_2. We need to pack the items into the knapsacks such that

- Each item is in at most one of the knapsacks
- For $j \in \{1, 2\}$, the *total weight* of the items in knapsack j is at most W_j.
- The *total value* of the items that are packed in either knapsack is maximized.

(a) (2 points) When first looking at the problem, one could think that it is equivalent to the standard knapsack problem with one knapsack of capacity $W' := W_1 + W_2$. Prove that this is not true by showing that in some cases, the total value that can be packed into one knapsack of capacity $W' = W_1 + W_2$ can be *arbitrarily* larger than the total value that can be packed into two knapsacks of capacities W_1 and W_2.

(b) (4 points) Give a dynamic programming algorithm that *optimally* solves the problem (for integer weights). What is the running time of your algorithm?

Exercise 2: Game Strategy (6 points)

Consider the following two-player game: There is a row of n objects (assuming n is even) with values $v_1, v_2, \dots, v_n$. In the game, the two players make moves alternatingly. In each odd move, the first player either selects the first or the last object of the row and removes it permanently from the row. In even turns, the opponent plays the game in the same way. Each time the player picks some object, it receives its value.

a) (4 points) Devise an dynamic programming algorithm to determine the maximum value that the first player (i.e., the player starting the game) can receive by the end of the game.

b) (2 points) What is the running time for the algorithm?