Dr. Ulf-Peter Schroeder

Paderborn, 26. April 2017
Abgabe: Keine!

Übungen zur Vorlesung
Methoden des Algorithmenentwurfs
SS 2017
Blatt 1

Aufgabe 1:
You are consulting for a trucking company that does a large amount of business shipping packages between New York and Boston. The volume is high enough that they have to send a number of trucks each day between the two locations. Trucks have a fixed limit W on the maximum amount of weight they are allowed to carry. Boxes arrive at the New York station one by one, and each package i has a weight w_i. The trucking station is quite small, so at most one truck can be at the station at any time. Company policy requires that boxes are shipped in the order they arrive. At the moment, the company is using a simple greedy algorithm for packing: they pack boxes in the order they arrive, and whenever the next box does not fit, they send the truck on its way. But they wonder if they might be using too many trucks, and they want your opinion on whether the situation can be improved. Here is how they are thinking. Maybe one could decrease the number of trucks needed by sometimes sending off a truck that was less full, and in this way allow the next few trucks to be better packed.
Prove that, for a given set of boxes with specified weights, the greedy algorithm currently in use actually minimizes the number of trucks that are needed.

Aufgabe 2:
Your friend is working as a camp counselor, and he is in charge of organizing activities for a set of junior-high-school-age campers. One of his plans is the following mini-triathalon exercise: each contestant must swim 20 laps of a pool, then bike 10 miles, then run 3 miles. The plan is to send the contestants out in a staggered fashion, via the following rule: the contestants must use the pool one at a time. In other words, first one contestant swims the 20 laps, gets out, and starts biking. As soon as this first person is out of the pool, a second contestant begins swimming the 20 laps; as soon as he or she is out and starts biking, a third contestant begins swimming ... and so on. Each contestant has a projected *swimming time* (the expected time it take him or her to complete the 20 laps), a corresponding projected *biking time* and a corresponding projected *running time*. Your friend wants to decide on a *schedule* for the triathalon: an order in which to sequence the starts of the contestants. Let's say that the *completion time* of a schedule is the earliest time at which all contestants will be finished with all three legs of the triathalon, assuming they each spend exactly their projected swimming, biking, and running times on the three parts.
What's the best order for sending people out, if one wants the whole competition to be over as early as possible? Give an efficient algorithm that produces a schedule whose completion time is as small as possible.

Aufgabe 3:
Let $S = \{1, 2, \ldots, n\}$ be a set of unit-time tasks. To each task i there is associated its deadline d_i and a penalty w_i. The tasks are to be scheduled on a single processor. Whenever a task i does not finish within its deadline, a penalty w_i occurs. The problem is to schedule the tasks in order to minimize the overall penalty. Let I be the set which consists of all subsets A which can be scheduled in a way such that every task of A meets its deadline. Prove that $M = (S, I)$ is a matroid.
Solve the following instance of the scheduling problem via the *Greedy Algorithm for Matroids*:

S	1	2	3	4	5	6	7
d_i	4	2	4	3	1	4	6
w_i	7	6	5	4	3	2	1