Exercise Set 6

Exercise 6.1. Recall the version of KNAPSACK from Exercise 5.3, where items can be used multiple times. Give an FPTAS for this problem.

(4 points)

Exercise 6.2. Give an algorithm for BIN PACKING restricted to instances with a constant number of different item sizes whose running time is polynomially bounded in the number n of items.

Hint: Use dynamic programming.

(4 points)

Exercise 6.3. Prove that for any fixed $\varepsilon > 0$ there exists a polynomial-time algorithm which for any instance $I = (a_1, \ldots, a_n)$ of the BIN PACKING problem finds a packing using the optimum number of bins but possibly violating the capacity constraints by ε, i. e. an $f : \{1, \ldots, n\} \to \{1, \ldots, \text{OPT}(I)\}$ with $\sum_{f(i)=j} a_i \leq 1+\varepsilon$ for all $j \in \{1, \ldots, \text{OPT}(I)\}$.

Hint: Use Exercise 6.2.

(4 points)

Exercise 6.4. Let $A = (a_i)_{1 \leq i \leq p}$ and $B = (b_j)_{1 \leq j \leq q}$ be two inputs of the BIN PACKING problem. We write $A \subseteq B$ if there are indices $1 \leq k_1 < k_2 < \cdots < k_p \leq q$ with $a_i \leq b_{k_i}$ for $1 \leq i \leq p$. An algorithm for the BIN PACKING problem is called monotone if for inputs A and B with $A \subseteq B$ the algorithm needs at least as many bins for B as for A. Prove or disprove:

(a) NEXT FIT is monotone.

(b) FIRST FIT is monotone.

(4 points)

Deadline: Tuesday, May 17$^{\text{th}}$, until 2:15 PM (before the lecture) via eCampus. LaTeX-submissions are highly encouraged, however, you can also submit a scan (e.g. obtained with a mobile phone). Solutions may be submitted in groups of up to 2 people.

The websites for lecture and exercises can be found at:

`http://www.or.uni-bonn.de/lectures/ss22/appr_ss22_ex.html`

In case of any questions feel free to contact me at puhlmann@or.uni-bonn.de.