Technische Universität München
Fakultät für Informatik
Lehrstuhl für Algorithmen und Komplexität
Prof. Dr. Harald Räcke
Richard Stotz

Summer term 2019
Assignment 10
July 1, 2019

Efficient Algorithms and Data Structures II

Deadline: July 8, 2019, 10:15 am in the ***Efficient Algorithms*** *folder.*

Homework 1

(a) Consider the greedy algorithm for the knapsack problem: Sort the objects by decreasing ratio p_i/w_i and greedily pick items in that order. Show that the approximation ratio of this algorithm is unbounded.

(b) Disappointed by the algorithm from part (a), we improve the algorithm as follows. Order the items as in (a) and find the smallest integer k, so that the total weight of the first k items exceeds the weight bound W. Wlog, the ordered items are named $1, 2, \ldots, n$. Pick items $\{1, \ldots k-1\}$ or just item $\{k\}$, depending on which set yields a higher profit.

Show that the improved greedy algorithm returns a 2-approximation.

Homework 2

The *First-Fit heuristic* for solving the bin packing problem takes each object in turn and places it in the first bin that can accommodate it.

(a) Show that this approach gives a factor 2 approximation algorithm for bin packing.

(b) Give an example on which First-Fit does as bad as $\frac{5}{3} \cdot \mathrm{OPT}$.

Homework 3

Let G be a complete undirected graph in which all edge lengths are either 1 or 2 (clearly, G satisfies the triangle inequality). Give a 4/3 factor algorithm for TSP in this special class of graphs.
Hint: Start by finding a minimum 2-matching in G. A 2-matching is a subset S of edges so that every vertex has exactly 2 edges of S incident at it.

Homework 4

We say that a bin packing algorithm is *monotonic* if the number of bins it uses for packing a subset of the items is at most the number of bins it uses for packing all n items. Show that whereas Next-Fit is monotonic, First-Fit is not.
The Next-Fit heuristic tries to pack the next item only in the most recently started bin. If it does not fit, it is packed in a new bin.

Tutorial Exercise 1

In the maximum k-cut problem, we are given an undirected graph $G = (V, E)$, and non-negative weights $w_{ij} \geq 0$, for all edges $(i, j) \in E$. The goal is to partition the vertex set V into k parts $V_1 \ldots, V_k$ so as to maximize the weights of all edges whose endpoints are in different parts (i.e., $\max. \sum_{(i,j)\in E: i\in V_a, j\in V_b, a\neq b} w_{ij}$).
Give a randomized $\frac{k-1}{k}$ approximation algorithm for the maximum k-cut problem.

Tutorial Exercise 2

Consider the following variant of MAX-CUT: In addition to the graph $G = (V, E)$, we are given two sets of vertex pairs $S_1, S_2 \subseteq V \times V$. The pairs of S_1 need to be separated, the pairs of S_2 need to be on the same side of the cut. Under these constraints, the problem is to find the maximum weight cut.
Give a vector program relaxation for this problem. Apply an adapted form of the MAX-CUT algorithm from the lecture to round its solution to an integral solution. The approximation factor should still be at most $\alpha \geq 0.878$.

> [Hermann Rubin] showed me the Chebyshev type of proof that gives rise to what's now called the Chernoff bound, but it is certainly Rubin's. [...] I am very unhappy about the fact that I did not properly credit Rubin at that time because I thought it was a rather trivial lemma, but many things are only trivial once you know them.
>
> – H. Chernoff