Thomas Kesselheim
Anna Heuser

October 20, 2025
Due: October 27, 2025 at 10:00 p.m.

Algorithms and Uncertainty

Winter Term 2025/26

Exercise Set 2

If you would like to submit your solutions for this problem set, please send them via email to aheuser1@uni-bonn.de by Monday evening. Submitting solutions in groups is also possible.

If you would like to present one of your solutions in class, please use the following link to book a presentation slot by Monday evening:
https://terminplaner6.dfn.de/b/3308d4d826c0872fe0291cbacd0d7888-1431302
A short meeting to discuss your solution is mandatory before presenting it in class. To book a time slot for this meeting, please use the following link by Monday evening as well:
https://terminplaner6.dfn.de/b/8b86f85c2af4f4e789083f904474190e-1431312

Exercise 1: (3+4 Points)
Given an instance of Set Cover, let $f = \max_{e \in U} |\{S \in \mathcal{S} \mid e \in S\}|$ denote the *frequency* of the set system.

(a) Consider the unweighted version of Online Set Cover, i.e., $c_S = 1$ for all $S \in \mathcal{S}$, and the following algorithm: Upon arrival of element e, if $\sum_{S:e \in S} x_S = 0$, set $x_S = 1$ for all S with $e \in S$ and $y_e = 1$. Otherwise set $y_e = 0$. Show that this algorithm is f-competitive by using Lemma 3.7.

(b) Now, we generalize the algorithm from (a) to the weighted version. Let $g_e = \max\{0, 1 - \sum_{S:e \in S} x_S\}$ and let S_e be the cheapest set covering e. Upon arrival of element e, for each S that covers e, increase x_S by $\frac{c_{S_e}}{c_S} g_e$ and set $y_e = c_{S_e} g_e$. Show that this algorithm is f-competitive by using Lemma 3.7.

Exercise 2: (5 Points)
Again, given an instance of Set Cover, let $f = \max_{e \in U} |\{S \in \mathcal{S} \mid e \in S\}|$ denote the frequency of the set system.
Use our results from the fourth lecture to devise an online algorithm that is $O(\log f)$-competitive for fractional set cover and prove this. You may assume that f is known beforehand.

Hint: One bound in the analysis from the lecture can be improved for $f < n$. Use it to adapt the algorithm.