Thomas Kesselheim
Rebecca Lehming

October 28, 2024
Due: November 04, 2024 at 10:00 p.m.

Algorithms and Uncertainty

Winter Term 2024/25

Exercise Set 4

If you want to hand in your solutions for this problem set, please send them via email to rlehming@uni-bonn.de *by Monday evening. Of course, submitting solutions in groups is also possible.*

If you would like to present one of the solutions in class, please also send an email to rlehming@uni-bonn.de *stating* ***which task*** *you would like to present in* ***which of the tutorials****. Deadline for the email is Monday, 10:00 pm. Please note that the tasks will be allocated on a first-come-first-serve basis, so sending this email earlier than Monday evening is highly recommended.*

Exercise 1: (1+4 Points)

Consider the following randomized algorithm for Online Bipartite Matching:
Whenever a vertex $r \in R$ is revealed, let L_r be the set of currently unmatched neighbors of r. Then choose any $l \in L_r$ uniformly at random and match r to l.

(a) Explain the difference between this algorithm and the Ranking Algorithm from Lecture 7.

(b) We are given an instance of Online Bipartite Matching with n offline nodes $\ell_1, \dots, \ell_n$ and n online nodes that appear in order $r_1, \dots, r_n$. For every $i \in [n]$, r_i is connected to ℓ_i. Additionally for every $i \in [\frac{n}{2}]$, r_i is connected to every node in $\{\ell_{\frac{n}{2}+1}, \dots, \ell_n\}$. Show that the algorithm achieves an expected competitive ratio of at most $\frac{1}{2} + \frac{O(\log n)}{n}$ on this instance.

Exercise 2: (3+4 Points)

(a) Suppose a tourist visits Bonn and wants to try out all the different restaurants in the city. So every evening she uniformly at random picks one of the n restaurants. Show that the expected number of days she needs for visiting every restaurant at least once is $\Theta(n \log n)$.
Hint: You can use that, when performing a sequence of independent trials with success rate p, the expected number of trials you need until your first success is $\frac{1}{p}$.

(b) Now we have a set of memory items P of size n and a cache that can store up to k items. We assume the cache to be full at the beginning. We need to answer a request sequence $\sigma = \sigma_1, \sigma_2, \dots, \sigma_m$ where $\sigma_i \in P$ for all i. Each time we want to access an item $x \in P$ one of the following happens:

- If x is not in the cache we remove one element from the cache and add x instead. This induces a cost of 1.
- If x is in the cache we do nothing with a cost of 0.

We define cost of σ to be sum of costs of all request $\sigma_1, \ldots, \sigma_m$. The goal is to minimize the total cost.
Use Yao's principle and your result in (a) to show that every randomized algorithm for this problem is $\Omega(\log k)$ competitive.