University of Freiburg
Dept. of Computer Science
Prof. Dr. F. Kuhn
S. Faour, M. Fuchs, A. Malyusz

Algorithm Theory
Exercise Sheet 9

Due: Friday, 22nd of December 2023, 10:00 am

Exercise 1: Dracula's path *(8 Points)*

Count Dracula, eternal in his quest for the arcane, sought a peculiar path through the Midnight Matrix—a path shrouded in darkness. The challenge was to find the shortest simple path with even edges in an undirected graph G, from a starting point, marked by the chilling presence of a coffin s, to a destination enveloped in ancient cryptic powers t. Help Dracula by finding a polynomial algorithm for this problem.
Hint:Try to construct a new graph. Create a matching in it and try to grab the even shortest path between s and t as an alternating path in this new graph.

Exercise 2: Bonus Points Problem *(10 Points)*

In the mysterious land of Graphylvania, where mathematical enigmas echoed through the haunted valleys and enchanted vertices held the key to ancient secrets, a peculiar game unfolded. Two formidable players, known as the Graph Masters, engaged in a strategic duel over a finite graph denoted as G. The rules of the game were as intricate as the web of shadows that cloaked Graphylvania. The Graph Masters, taking turns with a solemn rhythm, selected previously untouched vertices, ensuring that each chosen point was adjacent to the one preceding it. The starting vertex is choosen by the Graph Master that play's second. The eerie echoes of their moves resonated through the cryptic edges of G. As the game progressed, a cryptic truth emerged—the last player capable of choosing a vertex would be crowned the ultimate Graph Master. The stakes were high, and the winner would gain not only mathematical glory but also access to the ancient powers hidden within the graph. Rumors whispered through the haunted halls of Graphylvania that a secret lay dormant within the very essence of the graph G. A spectral mathematician known for his insatiable thirst for mathematical truth, Count Graphula, an eternal observer of the mathematical realm, sought to uncover the secrets of the game. Help Count Graphula in finding an if and only if theorem where for every graph you can tell who has the winning strategy.

Exercise 3: Probabilistic Algorithms *(4+8 Points)*

(a) In the dark, brooding realm of Transylvania, where legends of the undead and supernatural mysteries whispered through the centuries, a mathematical puzzle unfurled amidst the ancient castle of the legendary vampire, Count Dracula.

Within the shadowy corridors of Dracula's castle, a collection of N mystical points manifested—each point in the plane, a manifestation of arcane energies that resonated with the supernatural forces permeating the very stones of the castle. Dracula, eternal in his quest for knowledge, sought to unravel the enigma concealed within these points.

The challenge presented itself in the form of an ancient riddle—a line must be drawn through these mystical points, ensuring that it traversed at least $\frac{N}{4}$ of the total points. It was known that such a line exists. As the night fell and the moon cast an eerie glow over the castle, Dracula, intrigued by the mathematical puzzle, summoned the denizens of the night to aid him. You are one of these creatures. Create an algorithm which with high probability (at least 0.999) finds this mystical line.

(b) In class, we looked at the following simple contention resolution problem. There are n processes that need to access a shared resource. Time is divided into time slots and in each time slot, a process i can access the resource if and only if i is the only process trying to access the resource. We have shown that if each process independently tries to access the resource with probability $1/n$ in each time slot, in time $O(n \log n)$, all processes can access the resource at least once with high probability. The goal of the exercise is to improve the algorithm and to get an $O(n)$ time algorithm under the following assumptions.

- As in the lecture, all the processes know n (the number of processes). In the algorithm of the lecture, this is needed because the probability $1/n$ for accessing the resource depends on n. As in the lecture, we also assume that all processes start together in the first time slot.
- If a process tries to access the resource in a time slot, the process afterwards knows whether the access was successful or not. Also, we assume that a process only needs to succeed once, i.e., once a process has been successful, it stops trying to access the resource.

The goal of this exercise is to give and analyze a randomized algorithm which guarantees that for some given constant $c > 0$ with probability at least $1 - 1/n^c$, during the first $O(n)$ time slots, each of the n processes can access the resource at least once.

(a) (2 points) Let us first assume that in each time slot at most $n/\ln n$ processes (among n processes) need to access the resource. Adapt the algorithm of the lecture such that all processes succeed in accessing the channel in $O(n)$ rounds with probability at least $1 - 1/n^{c+1}$.

(b) (1 point) Let us now assume that we are given an algorithm which guarantees that after $T(n)$ time slots, the number of processes which have not yet succeeded is at most $n/\ln n$ with probability at least $1 - 1/n^{c+1}$. What is the probability that all n processes succeed when combining this algorithm with the adapted algorithm of the lecture from question (a). Define the appropriate probability events to analyze this probability.

(c) (5 points) It remains to give an algorithm to which manages to get rid of all except $n/\ln n$ of the processes with probability at least $1 - 1/n^{c+1}$. Show that this can be achieved by an algorithm which runs in multiple stages. You can use the following hint.

Hint: *You can make use of the following fact. Consider a time interval consisting of at least $e^2 k$ time slots. During the time interval, there are at most k processes trying to access the resource and in each time slot, each of the at most k processes tries to access the resource with probability $1/k$. Then, with probability at least $1 - e^{-k}$, at the end of the interval, at most $k/2$ of the processes have not succeeded to access the resource.*