Exercise Sheet 3

Submission due by June 4th, 2026

Organizational Info

- You have *three weeks* (including the holidays) to work on the exercise problems.
- Please work in pairs or groups of three if possible.
- Submit your solutions as a *well-formatted* PDF via ILIAS.

Problem 1: Odd Subgraph — $4 + 8 = 12$ points

Given a graph G and a parameter k. The problem Odd Subgraph asks whether there is a subgraph of G with k edges in which all vertices have odd degree. Note that a subgraph does not necessarily have to be vertex-induced.

Part (a) Show that if the maximum degree in G is at most k and G contains at least $k \cdot (2k - 1)$ edges, then there is a matching in G with k edges.

Part (b) Show that for Odd Subgraph, a kernel of size $O(k^2)$ is computable in polynomial time.

Hint: How do you solve the cases not covered by part (a)?

Problem 2: Edge Clique Cover — 16 points

The parameterized problem Edge Clique Cover is defined as follows. Given a graph G and a parameter k. The goal is to find a set of at most k cliques such that every edge of G is contained in at least one of the cliques.

Provide reduction rules that compute a kernel with at most 2^k vertices for Edge Clique Cover. Argue why these reduction rules are safe and explain why the resulting kernel has at most 2^k vertices.

Hint: Two vertices u and v are called *true twins* if $N(u) \cup \{u\} = N(v) \cup \{v\}$, where $N(u)$ denotes the neighbors of u. Can you get rid of such true twins?

Problem 3: Vertex Cover on bipartite graphs — $4 + 4 + 4 = 12$ points

The *Application of Lenstra* exercise from the previous version was moved to sheet 4 and replaced with this new exercise.

Part (a) Show that the LP relaxation of the ILP for Vertex Cover has an optimal integer solution if the graph is bipartite (i.e. that there is an optimal solution in which all variables are set to 0 or 1).

Part (b) *This subexercise builds on lecture 6 (June 1).*

Use the duality theorem to prove Kőnig's theorem. Kőnig's theorem states that in a bipartite graph the number of vertices in a minimum vertex cover equals the number of edges in a maximum matching.

You can use (without proof) that the LP relaxation of the ILP for Maximum Matching has an optimal integer solution if the graph is bipartite.

Part (c) Prove that we can find a minimum vertex cover in a bipartite graph in time $\mathcal{O}(m\sqrt{n})$.

You should use part (b) and the fact that a maximum matching can be computed in time $\mathcal{O}(m\sqrt{n})$ on bipartite graphs (you don't need to prove this).

Problem 4: Closest String — 10 bonus points

In this task, you should implement a program (using any programming language you prefer) that solves the Closest String problem. Given is a set of strings. The task is to find a minimum k and a string s such that s has a Hamming distance of at most k to each given string.

Use the algorithm from Active Session 1. However, you are welcome to implement further rules and make optimizations.

Describe in the PDF submission which procedures you have implemented and what else you have optimized. Also, in the PDF submission, state for each instance the smallest k for which you found a solution. Additionally, submit the source code and your found solutions (in the format described below) as a ZIP file. You can get 2 points for each solved instance.

Input format: The first line contains n, the number of strings. The next n lines contain one string each. All of these strings have the same length.

Output format: One line with a string that has the minimum Hamming distance to the given strings.

Note: Using the file `validator.py`, you can determine the Hamming distance of your solution to the given strings. It outputs the maximum Hamming-Distance over all given strings.