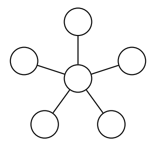
Exercise Sheet 5

Submission due by July 2, 2026

Problem 1: Stars **8 points**

A *5-star* is the following graph:



Given an undirected graph $G = (V, E)$ and a parameter k, we want to decide whether G contains at least k vertex-disjoint induced 5-stars. Use *color-coding* to show that this problem is in FPT.

Problem 2: Longest Cycle **14 points**

In the problem Longest Cycle, we have given a graph $G = (V, E)$ and a parameter k and we have to decide whether G contains a cycle of length *at least* k. Provide an FPT algorithm for this problem.

Hint: Note that there are graphs that do not contain a cycle of length exactly k, but nevertheless contain a cycle of length at least k.

Problem 3: Colors and Kernels **4 + 14 = 18 points**

This exercise builds on lecture 8 (June 22).

Show that the following problems do not admit polynomial kernels (when parameterized by the respective k), unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$.

Part (a) Colorful Graph Motif: Given a graph $G = (V,E)$ and a coloring of its vertices into k colors (not necessarily a proper coloring), does G have a colorful connected subgraph H of size exactly k?

You can use without proof that Colorful Graph Motif is NP-complete.

Part (b) Colorful Left-Right Dominating Set: Given a bipartite graph G with sides of the bipartition L and R (called the *left* and *right* vertices, respectively), and a coloring of L into ℓ colors. Is there a colorful set $X \subseteq L$ of exactly ℓ left vertices that together dominate all right vertices, i.e., $N(X) = R$? The parameter we consider is $k = |R| + \ell$.

You can use without proof that COLORFUL LEFT-RIGHT DOMINATING SET is NP-complete.

Hint: Construct an instance selector that consists of right vertices.

Problem 4: CLOSEST STRING **10 bonus points**

In this task, you again have to implement a program that solves the CLOSEST STRING problem (in a simplified variant). We are given k strings of length n. The goal is to find the smallest D such that there exists a string s^* whose Hamming distance to each input string is at most D.

Solve the problem by means of an ILP as presented in lecture 6. You may introduce further optimizations. In the PDF submission, describe which modifications you made to the ILP formulation from the lecture and any additional insights that helped you solve the problem. Also list for each instance the smallest D for which you found a solution.

Additionally submit the source code and your obtained solutions (in the format described below) as a ZIP archive. You receive a point for every instance you solve.

Input file format: The first line contains k, the number of strings. The next k lines each contain a string. All strings have the same length and consist only of the lowercase letters `a-z`.

Output file format: One line with a string s^* that achieves the minimal Hamming distance to the given strings.

Hint: With the file `validator.py` you can compute the Hamming distance of your solution to the input strings; it reports the maximum over all distances.

Hint: This task requires an ILP solver. Many languages and solvers are available, but `Gurobi` is among the best. To use it, however, you need a license. You can obtain a free academic license with your university e-mail address.