Tutorial on FPT algorithms

1. **Polynomial kernel for Feedback Arc set in tournaments**
   Let G be a directed graph and F be a subset of arcs of T. We denote by $G \otimes F$ the graph obtained from G after reversing each arc of F. F is a **Feedback Arc Set** (FAS for short) if $G - F$ is a directed acylic digrah (DAG for short).

   Given a set of arcs F, you can define $rev(F) = \{yx : xy \in F\}$.

   (a) Prove that if $G \otimes F$ is a DAG, then F is a FAS.

   (b) Give an example of a digraph D with a FAS F such that $D \otimes F$ is not a DAG.

   (c) Let G be a directed graph and F a subset of arcs of G. Show that F is a minimal FAS if and only if F is a minimal set of arcs such that $G \otimes F$ is a DAG.

   A **tournament** is a directed graph T such that for each pair of vertices (u, v), exactly one of the arcs uv and vu is in T. In other words it is an orientation of a complete graph. We are intereste in the k-Feedback Arc Set in Tournaments problem that takes a tournament T and an integer k as input and return YES if and only if T has a FAS of size at most k.

   (d) Prove that a tournament is acyclic if and only it does not contain a directed triangle (that is a directed cycle on 3 vertices).

   (e) Find two simple reduction rules that will permit to get a kernel of size $k(k + 2)$ for the k-Feedback Arc Set in Tournaments problem.
   Hint: The rules look a lot as the two rules saw in class for k-Vertex Cover in graphs.

2. **Polynomial kernel for the k-Connected Vertex Cover Problem**
   In the Connected k-Vertex Cover Problem, we are given a graph G and an integer k, and the objective is to decide if G contains a vertex cover S of size at most k and such that $G[S]$ is connected (where $G[S]$ is the subgraph of G induced by S).

   (a) Give a simple graph for which the kernelization procedure seen in class for k-Vertex-Cover fails for Connected k- Vertex Cover.

   (b) Show that Connected Vertex Cover admits a kernel with at most $2^k + O(k^2)$ vertices.

3. **Triangle Packing**
   Given a graph G and an integer k, the goal is to decide if G contains k vertex-disjoint triangles. Using colour coding, show that the problem can be solved in times $2^{O(k)}n^{O(1)}$.

4. **Randomized algorithm for Feedback Vertex set**
   A feedback vertex set is a set of vertices S such that $G \setminus S$ is a forest.

(a) Let G be a multigraph with minimum degree at least 3 and let S be a feedback vertex set of G. Prove that more than half of the edges are incident to at least one vertex of S.

(b) Prove that there exists a polynomial-time randomized algorithm that, given a FEEDBACK VERTEX SET instance (G, k), either reports a failure or finds a feedback vertex set in G of size at most k. Moreover, if the algorithm is given a yes-instance, it returns a solution with probability at least 4^{-k}.

(c) Prove that there exists a randomized algorithm that, given a FEEDBACK VERTEX SET instance (G, k), in time $4^k \cdot n^{O(1)}$, either reports a failure or finds a feedback vertex set in G of size at most k. Moreover, if the algorithm is given a yes-instance, it returns a solution with constant probability.

5. **Iterative compression for FEEDBACK VERTEX SET in tournaments**
FEEDBACK VERTEX SET in tournament is the following problem:
Input: A tournament T and an integer k
Question: Is there $S \subseteq V(T)$ such that $T \setminus S$ is acyclic?
The goal of the exercise is to use iterative compression to get a $2^k \cdot n^{O(1)}$-time algorithm for FVS in tournaments.

   (a) Design a $3^k \cdot n^{O(1)}$-time algorithm using a simple branching.

   (b) Define the problems FVS COMPRESSION and DISJOINT FVS in tournaments.

   (c) Prove that if we can solve DISJOINT FVS in tournaments in time $n^{O(1)}$, then we can solve FVS in tournaments in time $2^k \cdot n^{O(1)}$.

   (d) Design a polynomial-time algorithm to solve DISJOINT FVS in tournament.

6. **Closest String Problem**
The closest string is an NP-Hard problem which tries to find the geometrical center of a set of input strings.
Given n strings $s_1, \ldots, s_n$ each of length m and an integer d, find a string x of length m such that $d_H(x, s_i) \leq d$ for $i = 1, \ldots, m$, where d_H denotes the Hamming distance.
Find an FPT algorithm for Closest String Problem running in time $O(nm + nd(d+1)^d)$.