Tutorial on FPT algorithms

1. **Feedback Vertex Set in bipartite tournaments**
   A *bipartite tournament* is an orientation of a complete bipartite graph. A *Feedback Vertex Set* of a directed graph G is a set of vertices S such that $G - S$ has no directed cycle.

   (a) Show that if a bipartite tournament has a directed cycle, then it has a directed cycle of length 4.

   (b) Detail an algorithm with running time $4^k n^{O(1)}$ that decides whether a bipartite tournament has a feedback vertex of size at most k. Justify the running time and the correctness of your algorithm.

   Solution:
   1. Let G be a bipartite tournament. Assume G has a directed cycle and let $C = (v_1, v_2, \ldots, v_k, v_1)$ be a smallest directed cycle G. Since G is bipartite, k is even. Assume for contradiction that $k \geq 6$. Since G is a bipartite tournament, there is an arc between v_1 and v_4. If $v_4 \to v_1$, then (v_1, v_2, v_3, v_4) is a directed cycle of length 4, contradicting the minimality of C. If $v_1 \to v_4$, then $(v_1, v_4, \ldots, v_k, v_1)$ is a directed cycle of length $k - 2$, contradicting again the minimality of C.

   2. The algorithm follows the same principle as the algorithm for vertex cover in slides 22.

   We propose the following algorithm ALG for an instance (G, k).

   (a) If G has no directed cycle of length 4, return YES.

   (b) Otherwise, if $k = 0$, return NO.

   (c) Otherwise:
      - find a directed cycle (v_1, v_2, v_3, v_4) (this can be done in $O(n^4)$).
      - Compute, $ALG(G - v_i, k - 1)$ for $i = 1, 2, 3, 4$ and return YES, if one of the $ALG(G - v_i, k - 1)$ return YES.

      Each interior vertex of the search tree has degree 4, and the work done on each node of the search tree is $O(n^4)$, so the total running time is $4^k n^4$. The algorithm is correct because, by question 1 a set of vertices S of a complete bipartite tournament G is a FVS if and only if $G - S$ has no directed cycle of length 4.

2. **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.

Solution: Section 2.2.2 of the textbook parametrized algorithms.

a) Easy

b) Take a directed cycle, the set of all arcs is a FAS, but when you reverse them you don't get a DAG.

c) Let F be a minimal FAS. We want to prove that $G \otimes F$ is acyclic and that, for every $e \in F$, $G \otimes F \setminus e$ has a directed cycle. We first prove that $G \otimes F$ is a DAG. Assume for contradiction that $G \otimes F$ has a directed cycle C. Let $y_1x_1, \ldots, y_\ell x_\ell$ be the arc of $A(C) \cap rev(F)$ (where $rev(F) = \{xy : yx \in F\}$) is the order of their appearance on C, and let $e_i = x_iy_i$ be the reverse of f_i (so e_i are arcs of F). By minimality of F, for $i = 1 \ldots, \ell$, there is a directed cycle containing $e_i = x_iy_i$ in $G \setminus (F \setminus e_i)$, and thus a directed path P_i from y_i to x_i included in $G \setminus F$. By replacing each $f_i = y_ix_i$ by P_i in C, we get a directed cycle of G disjoint from F, contradicting the fact that F is a FAS.

For the minimality, assume $G \otimes (F \setminus f)$ is a DAG for some $f \in F$. By question a, $F \setminus f$ is a FAS, contradicting the minimality of F.

Assume now that F is such that $G \otimes F$ is a DAG and F is minimal with this property. By question a), F is a FAS, and if there exists $F' \subsetneq F$ such that F' if a FAS, then by the first part of the question, $G \otimes F'$ is a DAG, contradicting the minimality of F. So F is a minimal FAS.

d) Let T be a tournament with no directed triangle. Assume for contradiction that T is not a DAG. Let $C = v_1v_2 \ldots v_kv_1$ be a shortest directed cycle of T. By hypothesis $k \geq 4$. Since T is a tournament, there is an arc between v_1 and v_3. If v_1v_3 is an arc, then $v_1v_3\dots v_kv_1$ if a directed cycle, and if v_3v_1, then $v_1v_2v_3v_1$ is a directed cycle. In both cases it contradicts the minimality of C.

e) Following question c), we aim to find a set of arcs F such that $G \otimes F$ is acylic. Moreover, by question d), it is enough to find a set of arcs F such that $G \otimes F$ has no directed triangle.
Observe that if a vertex v is contained in no directed triangle, a set of arcs is a FAS of G if and only if it is a FAS of $G \setminus \{v\}$. This gives us our first rule.

(R1) if a vertex v is not contained in any directed triangle, then delete v from T.

Observe that if an arc e if contained in at least $k+1$ directed triangles, then e must be contained in any FAS of size at most k. This gives us our second rule:

(R2) if an arc e is contained in at least $k+1$ directed triangles, then reverse e and reduce k by 1.

Let (T,k) be an instance of k-Feedback Arc Set in Tournaments and (T',ℓ) be the instance obtained from (T,k) after an exhaustive application of $R1$ and $R2$. We have that (T,k) is a YES-instance if and only if (T',ℓ) is.

Let F ba a FAS of (T',ℓ). We have:

- Since F is a FAS, every directed triangle of T contains an arc in F.
- By $(R2)$: each arc is contained in at most ℓ directed triangles.

So T ahs at most $|F| \times \ell$ directed triangle. Moreover, by $(R1)$, each vertex is contained in a directed triangle. So T has at most $|F| \times (\ell+2)$ vertices.

Hence, if (T,k) has a FAS of size at most k, then (T',ℓ) has a FAS of size at most ℓ and thus has at most $\ell(\ell+2) \leq k(k+2)$ vertices.

3. **Polynomial kernel for the k-Connected Vertex Cover Problem**
In the k-Connected 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 k-Connected Vertex Cover.

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

Solution:

(a) The kernelization consists in applying these two rules as much as possible:

(R1) If a vertex has degree 0, delete it.

(R2) If a vertex has degree at least $k+1$, delete it (and put it in the solution) and decrease k by 1.

Recall that after applying these two reduction rules, VERTEX COVER has a kernel with k^2 edges and k^2+k vertices, that is a YES instance has at most k^2 edges and k^2+k vertices.
The problem is with the first rule that might delete some vertex usefull to make the solution connected. For example: take two vertices a and b both adjacent to $k+1$ vertices and add a vertex c adjacent to a and b. Applying the second rules leads to the deletion of a and b, and then the first rule leads to the deletion of c, while $\{a,b,c\}$ is a solution.

(b) Let G be a YES-instance.
Delete vertices of degree 0 in G, they are useless.
Let X be the set of vertices of degree at least $k+1$. Each vertex of X must belong to any solution, so $|X| \leq k$.
Let Y be the set of vertices of degree 0 in $G-X$, and let $Z = G-(X \cup Y)$. Observe that Y is the set of vertices that would have been deleted by $(R1)$
Z must have a vertex cover of size at most k (actually of size at most $k-|X|$), and vertices in Z has degree at most k. So Z has at most k^2 edges, and since vertices in Z have degree at least 1, $|Z| \leq 2k^2$.
Now, as explained in the previous question, the only purpose for adding a vertex of Y in the solution, is to make it connected. Hence, if u and v are two vertices in Y with the same neighborhood (which is included in X), we can delete one of them. Hence, we keep at most $2^{|X|} \leq 2^k$ vertices of Y: at most one for each subset of X.
All together, and after reductions (delete the vertices of degree 0 in the original graph, then delete vertices in Y that have the same neighborhood), either we return NO, or the graphs has at most $2^k + k + 2k^2$ vertices.

4. **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 $2^{O(k)}n^{O(1)}$.

5. **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 $O(nm+nd(d+1)^d)$.

6. **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 Monte-Carlo algorithm with one-sided error that, given a FEEDBACK VERTEX SET instance (G, k), either reports a FALSE or finds a feedback vertex set in G of size at most k. 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.

7. **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.