Bounded Search Trees

Find branching rules for a variety of different problems that show the respective problem is in FPT. The exact running-time isn't important at this stage.

Cluster Editing

Problem statement

Given a graph *G* and a parameter *k*, can *G* be turned into a cluster graph using at most *k* operations?
An operation is the insertion or deletion of a single edge.
A cluster graph is a disjoint union of cliques.

Solution

A graph is a disjoint union of cliques iff it contains no induced P_3.
Assume we have an induced P_3 on vertices *u, v, w*. Then at least one of the following three operations must be performed:

1. insert the edge {u,w},
2. delete the edge {u,v}, or
3. delete the edge {v,w}.

Thus we obtain a branching of width 3, and in every branch the parameter *k* is reduced by one.

Independent Set

Problem statement

Given a *d*-degenerate graph *G* and a parameter $k + d$, does *G* contain an independent set of size *k*?

A graph is *d*-degenerate if every subgraph has a vertex of degree at most *d*.

Solution

Pick a vertex v whose degree is at most d. We branch on the decision which neighbor of v belongs to the independent set.
There are up to d possibilities for choosing a neighbor, plus the additional option of not choosing any neighbor (in which case we can safely pick v itself). Consequently, we obtain at most $d + 1$ branches. In each branch one vertex is selected for the independent set, thereby decreasing k by one.

For the new instance in each branch we delete the chosen vertex together with all of its neighbors (they cannot be part of the independent set any longer).

Closest String

Problem statement

Given k strings and a parameter d, does there exist a string whose Hamming distance to each of the k strings is at most d?

Solution

Let the given strings be $s_1, \dots, s_k$ and start with the candidate $s = s_1$. We can apply up to d modifications to s in order to make its distance to every other s_i at most d. If the Hamming distance between s and some s_i exceeds $2d$, we can abort and answer NO (no string can be within distance d of both s and s_i). If s already has distance at most d to every s_i, we abort and answer YES.

Otherwise, let s_i be a string whose distance from the current s is δ with $d < \delta \leq 2d$. To get a feasible solution, we change at least one of the δ positions where s and s_i differ. Consequently, we branch on which of those δ positions to modify. This yields a branching of width at most $\delta \leq 2d$. In each branch the number of remaining allowed modifications decreases by one.

Note that after a few steps the remaining budget of modifications may be smaller than the original d. If $x < d$ modifications are left, the target distance is still d. Hence we know that $d < \delta \leq x + d$ (and not $x < \delta \leq 2x$).

Odd Cycle Transversal

Problem statement

Given a perfect graph and a parameter *k*, is there a set of *k* vertices that hits (has non-empty intersection with) every odd cycle?

A graph is *perfect* if for every induced subgraph the size of a maximum clique ω equals the chromatic number χ.

Solution

Hitting every odd cycle is equivalent to making the graph bipartite after the selected vertices are removed.
If the current graph is already bipartite we are done.
Otherwise the chromatic number (and therefore also ω) is larger than 2, which implies the existence of a triangle. From that triangle we must delete at least one vertex. Hence we obtain a branching of width *3*, and the parameter *k* is reduced by one in every branch.

Feedback Vertex Set in Tournaments

Problem statement

Given a tournament graph and a parameter *k*, can we delete *k* vertices so that the resulting graph is acyclic?

A tournament graph is a directed graph with exactly one edge for each vertex pair.

Solution

If the tournament graph *G* is not already acyclic, we show that it must contain a directed triangle. Let *C* be a shortest directed cycle in *G*. If *C* is no triangle, there exist two non-consecutive vertices *u, v* on *C*. Since *G* is a tournament graph, exactly one of the edges *uv* or *vu* is present; without loss of generality assume *uv* exists. This edge *uv* together with the path along *C* from *v* to *u* yields a shorter directed cycle, contradicting the minimality of *C*. Hence a directed triangle exists.

At least one vertex of this triangle must be removed. This yields a branching of width 3, decreasing k by one in each branch. Note that the subgraphs that arise after deletions are again tournament graphs.

Feedback Arc Set in Tournaments

Problem statement

Given a tournament graph and a parameter k, can we delete k edges so that the resulting graph is acyclic?

A tournament graph is a directed graph with exactly one edge for each vertex pair.

Solution

We can use the same idea as for **Feedback Vertex Set in Tournaments**, but we must deal with the fact that deleting edges destroys the tournament property. There are two ways to deal with this:

1. Argue that a short directed cycle (of length depending on k) still exists after each deletion, or
2. Circumvent the issue with the following statement.

Lemma.
There exists a feedback arc set of size at most k iff there is a set of edges A with $|A| \leq k$ such that reversing every edge in A makes the graph acyclic.

Hence, when looking for a minumum feedback arc set, we can also reverse edges instead of deleting them. The remaining argument is analogous to **Feedback Vertex Set in Tournaments**.

How to proof the lemma:
The direction "reversing makes acyclic" ⇒ "deleting makes acyclic" is obvious. For the other direction, let A be a minimum feedback arc set of size at most k. We show that reversing the edges in A makes the graph acyclic. Let G' be the graph obtained after reversal and assume that G' contains a directed cycle. Let C be such a directed cycle in G' containing the smallest possible number of reversed edges. Note that C must contain at least one reversed edge (otherwise it would already be a cycle in $G - A$). Pick such an edge $uv \in A$ with vu belonging to C. Since A is a minimum feedback arc set, $G - A + uv$ is cyclic. Thus, there is a directed path

form *v* to *u*. Replacing *vu* in *C* by that path (which does not contain reversed edges) yields a directed cycle with fewer reversed edges than *C*, contradicting the minimality of *C*.