Bounded search trees

16 February 2018

1. In the Cluster Vertex Deletion problem, we are given a graph G and an integer k, and the task is to delete at most k vertices from G to obtain a cluster graph (a disjoint union of cliques). Obtain a $3^k n^{O(1)}$-time algorithm for Cluster Vertex Deletion.

2. In the Cluster Editing problem, we are given a graph G and an integer k, and the objective is to check whether we can turn G into a cluster graph (a disjoint union of cliques) by making at most k edge editions, where each edition is adding or deleting one edge. Obtain a $3^k n^{O(1)}$-time algorithm for Cluster Editing.

3. An undirected graph G is called perfect if for every induced subgraph H of G, the size of the largest clique in H is same as the chromatic number of H. In this exercise we consider the Odd Cycle Transversal problem, restricted to perfect graphs. Construct $3^k n^{O(1)}$-time branching algorithm.

4. Let F be a set of graphs. We say that a graph G is F-free if G does not contain any induced subgraph isomorphic to a graph in F; in this context the elements of F are sometimes called forbidden induced subgraphs. For a fixed set F, consider a problem where, given a graph G and an integer k, we ask to turn G into a F-free graph by: (vertex deletion) deleting at most k vertices; (edge deletion) deleting at most k edges; (completion) adding at most k edges; (edition) performing at most k editions, where every edition is adding or deleting one edge. Prove that, if F is finite, then for every of the four aforementioned problems there exists a $2^{O(k)} n^{O(1)}$-time FPT algorithm. (Note that the constants hidden in the O()-notation may depend on the set F.)

5. In the Vertex Cover/OCT problem, we are given an undirected graph G, an integer ℓ, and an odd cycle transversal Z of size at most

k, and the objective is to test whether G has a vertex cover of size at most ℓ. Show that Vertex Cover/OCT admits an algorithm with running time $2^k poly(n)$.

6. In this exercise we consider FPT algorithms for Feedback Arc Set in Tournaments and Feedback Vertex Set in Tournaments. Recall that a tournament is a directed graph, where every pair of vertices is connected by exactly one directed edge (in one of the directions).

   (a) Let G be a digraph that can be made into a tournament by adding at most $k \geq 2$ directed edges. Show that if G has a cycle then it has a directed cycle of length at most $3\sqrt{k}$.

   (b) Show that Feedback Arc Set in Tournaments admits a branching algorithm with running time $(3\sqrt{k})^k n^{O(1)}$.

   (c) Show that Feedback Vertex Set in Tournaments admits a branching algorithm with running time $3^k poly(n)$.

   (d) Observe that, in the Feedback Arc Set in Tournaments problem, we can equivalently think of reversing an edge instead of deleting it. Use this observation to show a branching algorithm for Feedback Arc Set in Tournaments with running time $3^k n^{O(1)}$.

7. A bipartite tournament is an orientation of a complete bipartite graph, meaning its vertex set is a union of two disjoint sets V_1 and V_2 and there is exactly one arc between every pair of vertices u and v such that $u \in V_1$ and $v \in V_2$.

   (a) Show that a bipartite tournament has a directed cycle if and only if it has a directed cycle on 4 vertices.

   (b) Show that Directed Feedback Vertex Set and Directed Feedback Arc Set admit algorithms with running time $4^k n^{O(1)}$ on bipartite tournaments.

8. In the Min-Ones-r-SAT problem, we are given an r-CNF formula ϕ and an integer k, and the objective is to decide whether there exists a satisfying assignment for ϕ with at most k variables set to true. Show that Min-Ones-r-SAT admits an algorithm with running time $f(r;k)n^{O(1)}$ for some computable function f.

9. In the Min-2-SAT problem, we are given a 2-CNF formula ϕ and an integer k, and the objective is to decide whether there exists an

assignment for ϕ that satisfies at most k clauses. Show that MIN-2-SAT can be solved in time $2^k n^{O(1)}$.

10. In the MINIMUM MAXIMAL MATCHING problem, we are given a graph G and an integer k, and the task is to check if G admits an (inclusion-wise) maximal matching with at most k edges.

    (a) Show that if G has a maximal matching of size at most k, then $V(M)$ is a vertex cover of size at most $2k$.

    (b) Let M be a maximal matching in G and let $X \subseteq V(M)$ be a minimal vertex cover in G. Furthermore, let M_1 be a maximum matching of $G[X]$ and M_2 be a maximum matching of $G[V(G) \setminus V(M_1)]$. Show that $M_1 \cup M_2$ is a maximal matching in G of size at most $|M|$.

    (c) Obtain a $4^k n^{O(1)}$-time algorithm for MINIMUM MAXIMAL MATCHING.