Multivariate Algorithmics:
Problem set 5

Dr. Karl Bringmann, Dr. Holger Dell, Cornelius Brand, Marc Roth
http://bit.ly/MulAlg18

Due: Before the lecture on December 18, 2018 Tutorial: December 20, 2018

Problem 5.1 (✐) Let G be a graph with tree-decomposition $\mathcal{T} = (T, \{X_t\}_{t \in V(T)})$. Prove that every clique of G is contained in some bag X_t.

Problem 5.2 (✐) Prove Lemma 7.4 from the lecture:
Given a tree decomposition $\mathcal{T} = (T, \{X_t\}_{t \in V(T)})$ of a graph G of width at most k, one can in time $O(k^2 \cdot \max(|V(T)|, |V(G)|))$ compute a nice tree decomposition of G of width at most k that has at most $O(k|V(G)|)$ nodes.

Problem 5.3 A *cut* of a graph G is a partition (A, B) of the vertices of G and the *size* of a cut (A, B) is the number of edges of G that have one endpoint in A and one endpoint in B. The problem MaxCut asks, given a graph G, to compute the size of the largest cut in G. Prove that MaxCut can be solved in time $f(\mathsf{tw}(G)) \cdot |V(G)|$ for some computable function f by invoking Courcelle's Theorem.

Problem 5.4 An n-vertex graph G is called an α-*edge-expander* if for every set $X \subseteq V(G)$ of size at most $n/2$ there are at least $\alpha \cdot |X|$ edges of G that have exactly one endpoint in X. Prove that the treewidth of an n-vertex d-regular α-edge-expander is $\Omega(n\alpha/d)$.

Problem 5.5 (☠, 2 Points) A *homomorphism* from a graph H to a graph G is a function

$$\varphi : V(H) \to V(G)\,,$$

such that for every edge $\{u, v\}$ of H it holds that $\{\varphi(u), \varphi(v)\}$ is an edge of G.

a) (1 Point) Prove that one can decide whether there exists a graph homomorphism from H to G in time

$$f(|H|) \cdot |V(G)|^{\mathsf{tw}(H)+O(1)}$$

by constructing a dynamic programming algorithm over the tree decomposition of H.

b) (1 Point) Prove that one can decide whether a graph H is a subgraph of a graph G in time

$$f(|H|) \cdot |V(G)|^{\mathsf{tw}(H)+O(1)}\,.$$

c) (1 Bonus Point) Show that there are graphs H for which at least one of the above can be done significantly faster, that is, for every k find a graph H_k of treewidth $\Omega(k)$ such that either a) finding a homomorphism from H_k to G or b) deciding that H_k is a subgraph of G can be done in time

$$f(|H_k|) \cdot |V(G)|^c\,,$$

for some constant c that does not depend on H_k.