Sparsity — tutorial 6

Tree-depth, tree-width and low tree-depth colorings

Problem 1. Prove that a class $\mathcal{C}$ of graphs has bounded tree-depth if and only if there is a number k such that no graph $G \in \mathcal{C}$ contains a path of length k.

Problem 2. Let G be an n-vertex graph with m edges. Show that we can compute in time $\mathcal{O}(n+m)$ a tree-depth decomposition of depth at most $2^{\mathsf{td}(G)}$.

Definition 1. A *tree-decomposition* of a graph G is a pair $\mathcal{T} := (T, (B_t)_{t \in V(T)})$ consisting of a tree T and a family $(B_t)_{t \in V(T)}$ of sets $B_t \subseteq V(G)$, such that

1. for all $v \in V(G)$ the set
$$B^{-1}(v) := \{t \in V(T) \ : \ v \in B_t\}$$
is non-empty and induces a subtree of T and
2. for every edge $e \in E(G)$ there is $t \in V(T)$ with $e \subseteq B_t$.

The *width* of $\mathcal{T}$ is $\max\{|B_t| - 1 \ : \ t \in V(T)\}$. The *tree-width* of G, denoted $\mathsf{tw}(G)$, is defined as the minimal width of any tree-decomposition of G.

Problem 3. Let G be an n-vertex graph. Prove that $\mathsf{tw}(G) \leqslant \mathsf{td}(G) \leqslant (\mathsf{tw}(G)+1) \cdot \log n$.

Problem 4. Let $H \preceq G$. Prove that $\mathsf{td}(H) \leqslant \mathsf{td}(G)$ and $\mathsf{tw}(H) \leqslant \mathsf{tw}(G)$.

Problem 5. Let G be an n-vertex graph. Prove that $\mathrm{col}_n(G) = \mathsf{tw}(G) + 1$.

Problem 6. Let $\mathcal{C}$ be a class of graphs of tree-width at most k. Prove that it can be tested in linear time whether $G \in \mathcal{C}$ is 3-colorable.

Problem 7. Let $\mathcal{C}$ be a class of graphs of tree-width at most k and let H be graph. Prove that for every n-vertex graph $G \in \mathcal{C}$ it can be tested in time $f(|H|, k) \cdot n$ whether G contains a subgraph isomorphic to H.