CHAPTER 2

Tree-Decompositions of Graphs

Tree-decompositions of graphs play an important role in graph structure theory, in the theory of algorithms and in practical computation.

2.1. Introduction to tree-Width

Definition 2.1.1. Let G be a graph. A *tree-decomposition* of G is a pair (T, W), where T is a tree and $W = (W_t : t \in V(T))$ is such that

(i) $\bigcup_{t \in V(T)} W_t = V(G)$, and every edge of G has both ends in some W_t, and

(ii) if $t, t', t'' \in V(T)$ and t' lies on the path from t to t'' in T, then $W_t \cap W_{t''} \subseteq W_{t'}$.

The *width* of (T, W) is $\max\{|W_t| - 1 : t \in V(T)\}$, and the *tree-width* of G is the minimum width of a tree-decomposition of G.

Exercise 2.1.2 ([**10**])**.** Let (T, W) be a tree-decomposition of G, let H be a connected subgraph of G, and let $t, t'' \in V(T)$ be such that $V(H) \cap W_t \neq \emptyset \neq V(H) \cap W_{t''}$. Prove that if $t' \in V(T)$ lies on the path of T between t and t'', then $V(H) \cap W_{t'} \neq \emptyset$.

Exercise 2.1.3 ([**10**])**.** Let (T, W) be a tree-decomposition of G, and let $e \in E(T)$ with ends t_1, t_2. Then $T \backslash e$ has exactly two components T_1, T_2 say. Prove that there is no edge between $\bigcup_{t \in V(T_1)} W_t - (W_{t_1} \cap W_{t_2})$ and $\bigcup_{t \in V(T_2)} W_t - (W_{t_1} \cap W_{t_2})$.

Exercise 2.1.4. Let (T, W) be a tree-decomposition of a graph G, and let H be a complete subgraph of G. Prove that there exists $t \in V(T)$ such that $V(H) \subseteq W_t$. Deduce that the complete graph K_n has tree-width $n - 1$.

Exercise 2.1.5. Let (T, W) be a tree-decomposition of G, and let e be an edge of G with ends u and v. Let $w \notin V(G)$ and for $t \in V(T)$ let $W'_t = (W_t - \{u, v\}) \cup \{w\}$. Prove that $W/e := (W'_t : t \in V(T))$ is a tree-decomposition of G/e, the graph obtained from G by contracting e. Deduce that if G has an H minor, then the tree-width of H is at most the tree-width of G.

Exercise 2.1.6. Prove that a simple graph G has tree-width at most 1 if and only if G is a forest.

Exercise 2.1.7. Prove that a graph G has tree-width at most 2 if and only if G is series-parallel.

Exercise 2.1.8 ([**8**])**.** A graph G is chordal if and only if G has a tree-decomposition (T, W) such that every W_t is a clique.

Exercise 2.1.9. If G is a graph and $k \geq 0$ is an integer, then G has tree-width at most k if and only if G is a subgraph of a chordal graph with no subgraph isomorphic to K_{k+2}.

Exercise 2.1.10. Prove that every graph G of tree-width t has a set $X \subseteq V(G)$ of size at most t such that every component of $G \backslash X$ has at most $|V(G)|/2$ vertices.

Exercise 2.1.11. Prove that every graph G of tree-width t on n vertices has a separation (A, B) of order at most t such that both $A - B$ and $B - A$ have at most $2n/3$ vertices.

Exercise 2.1.12. Let $n \geq 2$ be an integer. Prove that the $n \times n$ grid (the adjacency graph of the $n \times n$ chessboard) has tree-width n.

Exercise 2.1.13 ([**13**])**.** For every planar graph H there exists an integer n such that the $n \times n$ grid has a minor isomorphic to H.

Theorem 2.1.14. *For every planar graph H there exists an integer k such that if a graph G has tree-width at least k, then it has an H minor.*

Exercise 2.1.15. If H is non-planar then no such integer exists.

Remark 2.1.16. We will prove Theorem 2.1.14 later. The above theorem was first proved in [**11**]. A simpler proof with a better bound on k was given in [**13**]; the bound there is $k = 20^{2^{n^5}}$, where $n = 2|V(H)| + 4|E(H)|$. A yet simpler proof with only a marginally worse bound was given in [**6**]. It was an open problem for a long time whether the bound can be improved to one that is polynomial in k. An affirmative answer was recently given by Chekuri and Chuzhoy.

Exercise 2.1.17. Let H be the $n \times n$ grid, and let k be as in the above theorem. Prove that $k \geq \Omega(n^2 \log n)$.

Hint. Random graphs.

Definition 2.1.18. Consider the following cops-and-robbers game. The robber stands on a vertex of the graph and can at any time run at great speed to any other vertex along a path of the graph. He is not permitted to run through a cop, however. There are k cops, each of whom at any time either stands on a vertex or is in a helicopter (that is, is temporarily removed from the game). The objective of the player controlling the movement of the cops is to land a cop via helicopter on the vertex occupied by the robber, and the robber's objective is to elude capture. (The point of the helicopters is that cops are not constrained to move along paths of the graph—they move from vertex to vertex arbitrarily.) The robber can see the helicopter approaching its landing spot and may run to a new vertex before the helicopter actually lands. Thus, for the cops to capture the robber they will need to first occupy all vertices adjacent to the vertex where the capture is to take place, because otherwise the robber will be able to run to a different vertex and not be captured.

There are two forms of this game of interest. In the first, the robber is invisible, and so to capture him the cops must methodically search the whole graph. (An equivalent problem is that of clearing the vertices of some plague which infects along edges.) In the second form of the game the cops can see the robber at all times—the difficulty is just to corner him somewhere.

Exercise 2.1.19. Prove that if G has tree-width at most k, then $k + 1$ cops can capture a visible robber in G.