Algorithms on Trees

Problems that are NP-complete in general graphs can become rather easy in special graph classes. Often it happens in practice that the input to a graph problem is a tree. (For example, many networks are hierarchically structured.) Most problems on trees can be solved by bottom-up dynamic programming. We illustrate the principle by (once more) the Weighted Vertex Cover problem which is also equivalent to the Weighted Independent Set problem. A remark is that the unweighted problem version can even be solved by a greedy algorithm on trees.

In the given tree we distinguish an arbitrary node $r$ as the root. All edges are oriented away from the root. This defines a directed tree $T$. For every node, let $T_v$ denote the subtree with root $v$, consisting of $v$ and all nodes reachable from $v$ via directed edges. We denote the weight of a node $v$ by $w(v)$. For every $v$ we define $OPT(v, 1)$ and $OPT(v, 0)$ as the weight of a minimum vertex cover in $T_v$ with $v$ and without $v$, respectively. What we want is the minimum of $OPT(r, 1)$ and $OPT(r, 0)$.

These values are computed as follows. If $v$ is a leaf, we immediately have $OPT(v, 1) = w(v)$ and $OPT(v, 0) = 0$. Now let $v$ be an inner node, and $v_1, \ldots, v_d$ the children of $v$. If $v$ is not in the vertex cover, we have to take all children, hence $OPT(v, 0) = \sum_{i=1}^{d} OPT(v_i, 1)$. If $v$ is in the vertex cover, we can independently decide for any child to take it or not, and the minimum value is optimal. Hence we have $OPT(v, 1) = w(v) + \sum_{i=1}^{d} \min(OPT(v_i, 1), OPT(v_i, 0))$.

That’s all! The running time is $O(n)$, since every node is involved in only constantly many calculations for its parent node.

In graphs that are not exactly trees but enjoy a tree-like structure we can apply similar dynamic programming schemes. The theory of treewidth
has had a prominent role in graph algorithms since the 1990’s. However it is technically difficult – too much for a broad course. If you have a deeper interest: Book sections 10.4-10.5 give a first introduction to treewidth.

**Extra Topic (not exam-relevant)**

A geometric approach to data anonymization, based on greedy approximate Set Cover and sort of center selection, is presented. Instead of providing notes we refer to the original article: