The Theory of Computation
Bernard M.E. Moret
Addison-Wesley-Longman, 1998

Detailed Solutions
Chapter 7, Problem 7.27, Part 3

Prove that *Maximum-Leaves Spanning Tree* is NP-complete.

Instance: a graph, $G = (V, E)$, and a bound on the number of leaves of the spanning tree, B.

Question: does G have a spanning tree with at least B leaves?

Input Encoding and Size: An instance of this problem can be described by the two sets, V and E, which costs $\lceil \log_2 |V| \rceil$ for V (no need to name the elements here) and $|E| \cdot 2\lceil \log_2 |V| \rceil$ for E (naming each pair); or it can be described by adjacency lists, one for each vertex, which will cost $(|E| + |V|) \cdot \lceil \log_2(|V| + 1) \rceil$ (each list needs a terminator, hence the +1; the total number of list elements is $|E|$ plus the terminators, $|V|$ of them). In either case, the input size is $\Theta(|E| \cdot \log |V|)$. This representation is best suited to sparse graphs; for dense graphs, we might want an adjacency matrix representation, which would be more concise, taking only $\Theta(|V|^2)$ space. Since the graph might as well be assumed to be connected (otherwise it cannot have any spanning tree whatsoever),[1] the number of edges is bounded above and below by a polynomial function in the number of vertices, $|V| - 1 \leq |E| \leq |V|(|V| - 1)/2$. Thus the input size is bounded above and below by a polynomial function of $|V|$.

Certificate Description and Verification: The problem is in NP: a certificate is just a list of edges that make up the desired spanning tree. Given such a list, call it E', we can verify that the graph (V, E') is a tree and has at least B leaves in time proportional to $|E'| + |V|$, which is safely polynomial in the input size.

---

[1] This does not mean that all legal instances must be connected graphs, although we could certainly define the problem in that way. In the general case, though, we can check whether or not the graph is connected in $\Theta(|E| + |V|)$ time and dismiss such graphs right away. Our transformation will only produce connected graphs anyhow.

Mapping: We prove the problem NP-complete by reducing the known NP-complete problem *X3C* to it. An instance of *X3C* has a set S of $3q$ elements and a collection C of p subsets of S, each of which has exactly 3 elements; it can be encoded as p triples, each requiring $3\lceil \log_2 3q \rceil$ bits, plus the size of S, taking $\lceil \log_2 3q \rceil$ bits, for an input size of $\Theta(p \log q)$. Given such an instance of *X3C*, we construct the graph $G = (V, E)$ for our problem as follows. We let $V = S \cup C \cup \{r\}$; that is, the graph has one vertex for each element of S, one vertex for each 3-subset, and one additional vertex, r. We let $E = E_1 \cup E_2$, where $E_2 = \{(c, x_{c1}), (c, x_{c2}), (c, x_{c3}) \mid c = \{x_{c1}, x_{c2}, x_{c3}\} \in C\}$ and $E_1 = \{(r, c) \mid c \in C\}$. That is, there is an edge from the "root" r to each "3-set vertex" c and an edge from each 3-set vertex to each of its 3 "element vertices." Finally, we set the bound to $2q + p$.

Resource Requirements: Note that the construction can be accomplished in polynomial time, as it can be done in time linear in the size of the generated instance, which is $\Theta((3q + p) \log(3q + p))$. (Since $q \le p \le \binom{3q}{3}$, this is simply some polynomial in p, as is the size of the *X3C* instance.)

Correctness: We claim that the "root" of the constructed graph must be the "root" of the spanning tree. If it were not, all 3-set vertices would have to be reached through other 3-set vertices and their elements; but then we cannot have enough leaves in all. (The "best" case—although one in which there is no solution in either instance—has one element belonging to every 3-set; reaching the other $3q - 1$ elements requires $\lceil (3q - 1)/2 \rceil$ 3-set elements; the resulting maximal number of leaves is $(3q - 1) + (k - \lceil (3q - 1)/2 \rceil) + 1$, which simplifies to $k + 3q - \lceil (3q - 1)/2 \rceil$, which is smaller than $k + 2q$ for all $q > 1$. For $q = 1$, the values are equal, but that number is only reachable with a no-instance, since all 3-sets overlap.) (This whole reasoning could easily be avoided by endowing the "root" with a "tail," as in many similar constructions.) This being established, only one other observation need be made: any second edge from an element to a 3-set is advantageously replaced by an edge from the "root" to the 3-set: such a change does not impair the spanning property, maintains a tree structure, and cannot decrease the number of leaves. The proof now follows easily.

If the *X3C* instance has a solution, i.e., q 3-sets that cover S, then there is a corresponding spanning tree "rooted" at r, with all p edges to the p vertices corresponding to 3-sets, and with 3 edges from each of the q vertices corresponding to the solution 3-sets to the vertices corresponding to their elements. This tree has $3q + (p - q)$ leaves (corresponding to the set elements and to the unused 3-sets), which is exactly our bound. Conversely, if the constructed instance has a solution, our observation above allows us to assume that all p edges from vertex r to the vertices corresponding to 3-sets are present. The spanning tree then must have the structure described in the first part of the proof: any second edge touching a vertex corresponding to an element would cause a cycle and $p - q$ vertices corresponding to 3-sets must be leaves in order to meet the bound on the number of leaves, as the vertices corresponding to elements contribute at most $3q$ leaves.