We begin by noticing two facts related to the graph $\mathcal{H}$ defined in the previous problem. First, if T and T' are neighbors in $\mathcal{H}$, then the number of X-edges in T can differ from the number of X-edges in T' by at most one. Second, the solution given above in fact provides a polynomial-time algorithm to construct a T-T' path in H.

We call a tree *feasible* if it has exactly k X-edges. Our algorithm to search for a feasible tree is as follows. Using a minimum-spanning tree algorithm, we compute a spanning tree T with the minimum possible number a of X-edges. We then compute a spanning tree T with the maximum possible number b of X-edges. If $k < a$ or $k > b$, then there clearly there is no feasible tree. If $k = a$ or $k = b$, then one of T or T' is a feasible tree. Now, if $a < k < b$, we construct a sequence of trees corresponding to a T-T' path in $\mathcal{H}$. Since the number of X-edges changes by at most one on each step of this path, and overall it increases from a to b, one of the trees on this path is a feasible tree, and we return it as our solution.

[1] ex708.930.216