SELECTED SOLUTIONS

7.1 **(a)** FALSE; **(b)** TRUE.

7.2 **(a)** TRUE; **(b)** TRUE.

7.14 Let $A \in \text{NP}$. Construct NTM M to decide A in nondeterministic polynomial time.

$M =$ "On input w:

1. Nondeterministically divide w into pieces $w = x_1 x_2 \cdots x_k$.
2. For each x_i, nondeterministically guess the certificates that show $x_i \in A$.
3. Verify all certificates if possible, then *accept*.
   Otherwise if verification fails, *reject*."

7.21 We give a polynomial time mapping reduction from *CLIQUE* to *HALF-CLIQUE*. The input to the reduction is a pair $\langle G, k \rangle$ and the reduction produces the graph $\langle H \rangle$ as output where H is as follows. If G has m nodes and $k = m/2$ then $H = G$. If $k < m/2$ then H is the graph that is obtained from G by adding j nodes, each connected to every one of the original nodes, where $j = m - 2k$. Thus H has $m + j = 2m - 2k$ nodes. Observe that G has a k-clique iff H has a clique of size $k + j = m - k$ and so $\langle G, k \rangle \in$ *CLIQUE* iff $\langle H \rangle \in$ *HALF-CLIQUE*. If $k > 2m$ then H is the graph that is obtained by adding j nodes to G without any additional edges, where $j = 2k - m$. Thus H has $m + j = 2k$ nodes. Thus, G has a k-clique iff H has a clique of size k and so $\langle G, k_k \rangle \in$ *CLIQUE* iff $\langle H \rangle \in$ *HALF-CLIQUE*. We also need to show *HALF-CLIQUE* $\in$ NP. The certificate is simply the clique.

7.33 If we assume P $=$ NP then *CLIQUE* $\in$ P and so we can test whether G contains a clique of size k in polynomial time, for any value of k. By testing whether G contains a clique of each size, from 1 to the number of nodes in G, we can determine the size t of a maximum clique in G in polynomial time. Once we know t, we can find a clique with t nodes as follows. For each node x of G, remove x and calculate the resulting maximum clique size. If resulting size decreases, replace x and continue with the next node. If the resulting size is still t, keep x permanently removed and continue with the next node. When all nodes have been considered in this way, the remaining nodes are a t-clique.

7.42 Consider a TM running on an input. A ***crossing sequence*** is the sequence of states that the machine enters at a given cell as the computation progresses. For a particular TM, if all crossing sequences on all inputs are of a fixed length k or less, an NFA simulating that TM may be obtained. The NFA needs only enough states to record the current crossing sequence plus a start state. It proceeds by guessing the sequence of crossing sequences that would occur if the TM were processing its input, while checking that neighboring crossing sequences are consistent. We assume that the TM has been modified so that it accepts at the right-hand end of the input location. The NFA's accept states correspond to crossing sequences that contain the TM's accept state and that are consistent with the TM's computation on the initially blank portion of the tape following the input. Thus, a TM with crossing sequences of bounded length decides a regular language.

Next we show that a TM with unbounded crossing sequence length cannot run in $o(n \log n)$ time. Assume to the contrary that we have TM M that runs for $o(n \log n)$ time and that has arbitrarily long crossing sequences. Chose $\alpha > 0$ to be very small. On all inputs of length $n \geq n_0$, for some sufficiently large n_0, TM M runs in time $\alpha n \log_2 n$. For all these inputs, at least half of the crossing sequences must have length $2\alpha \log n$ or less, to avoid exceeding the overall running time. Call these the *short* crossing sequences. The number of distinct crossing sequences of length l is $|Q|^l$, where M has $|Q|$ states. By choosing α small enough, we can force lots of repetition among the short crossing sequences because $|Q|^{2\alpha \log_2 n}$ is much less than $n/2$. For each such k, consider the strings that have a crossing sequence of length k or more. Chose k so that all such strings have length n_0 or more, and let w be the shortest such string. Say that the long crossing sequence occurs at position i in w. Three positions a, b, and c must exist that have identical crossing sequences. Remove the substring from positions a though $b-1$ or from b through $c-1$ from w to avoid removing position i. Because positions with identical crossing sequences were overlaid, M operates on the new string as it did on w, except for the portion removed. Hence the crossing sequence of length k or more still occurs. But w was selected to be the shortest string having such a crossing sequence, a contradiction.