Don't forget to mark the state state in the transition diagrams.

#2–#4 asked for specific processes, not just a final solution. As stated in the problems, little credit was given for a correct end result without evidence of following the intended process. Refer to the slides and examples from class for these processes.

For #3, while ad hoc methods of building NFAs from regular expressions can lead to much smaller and simpler NFAs, the exercise is about becoming familar with the construction from Theorem 3.3. Use that construction!

For #4, methodical condensing of paths in the NFA into single transitions labeled with regular expressions is in the spirit of the NFA-to-RE strategy from class and approaches along these lines received close-to-full credit. For full credit, use the specific strategies (simple cycles, simple paths, etc) from class and show the result of each condensing step in terms of the resulting machine.

The NFA-to-RE strategy from class requires a single final state. For #4b, this needs to be handled first.

#5 asks for a proof, so a construction method (algorithm) is needed — specific example(s) alone are not enough. (Examples can be very helpful for illustrating an algorithm, but don't replace a description of the process that can be applied to any valid input.)

Also remember that while there is only ever a single start state in a DFA, there can be more than one final state — swapping start and final states doesn't result in a legal DFA if the original DFA has multiple final states. Be sure to handle that case.