1 Minimax
Do problem 5.7. You should do your proof by induction on the tree depth.

2 Alpha-beta 1
Describe, in precise terms, general scheme for producing a full tree of arbitrary depth for which alpha-beta will generate the fewest possible nodes (an optimal node ordering). Next describe, in precise terms, a pessimal node ordering.

3 Alpha-beta 2
Do problem 5.18. Again, proof by induction is your friend here.

4 CSPs
Do problem 6.6.

5 Sudoku 1
Show how to formulate Sudoku as a CSP with binary constraints. (This is actually easier than the previous problem if you represent things in a nice way.)

6 Sudoku 2
Implement a simple Sudoku solver using backtracking. Turn in your code and an example of what happens when you run it. If your solver is too slow to solve problems from the starting configuration, give it a board that is almost solved so that you can show at least one example of it working. As usual, you can use any reasonable language, and the code you turn in must be your own work.
7 Sudoku 3

Add the heuristic of choosing the most constrained variable first, then see if your solver is more efficient. To check how constrained a variable is, consider the other assignments already on the row, column and $3 \times 3$ square associated with the variable, rule out that values that are already used, and count up how many are left.