I. There are 4 variables, each with the same domain of 4 values.

You may choose the slots $S_1, \ldots, S_4$ to be the variables. Their domain is then {1, 2, 3, 4}, where the value i designates the painting P_i. Alternatively, you may choose the paintings to be the variables. In the following, we choose the slots to be the variables.

One set of constraints is that no two variables take the same value, i.e.: $S_i \neq S_j$, for all $i,j \in \{1,\ldots,4\}$, $i \neq j$.

The other set of constraints is that no two consecutive slots receive paintings with consecutive painting numbers, i.e., $S_i + 1 \neq S_{i+1}$ and $S_4 + 1 \neq S_1$

II.

1. There are 7 states, which can be colored with either one of three colors each. Hence, there are $3^7 = 2187$ possible assignments.
2. There are only 18 solutions (valid assignments). They can be counted as follows: Fix the color of SA (chosen because it is the most constraining variable). For each color of SA it remains 2 valid colors for each state, except T that still has 3. So let the backtracking algorithm a value for WA. Each color of SA (combined with the color already chosen for WA) leaves only one color for NT, then one color for Q, then one for NSW, then one for V. So, there are 6 valid assignments for the mainland states. In all of them, T can have any of the 3 colors.

III.

1. Let c1 through c5 be the carries of the last column (S + S) through the first (C + R), respectively.

   The variables are the letters A, C, D, E, G, N O, R, and S, and the 5 carries c1 through c5.

   The domain of each letter is {0, 1, …, 9}. The domain of each carry is {0, 1}, since the sum of two single-digit numbers plus a carry cannot total more than 19.

   The constraints are:
   $S + S = 10\,c1 + R$
   $S + D + c1 = 10\,c2 + E$
   $O + A + c2 = 10\,c3 + G$
   $R + O + c3 = 10\,c4 + N$
   $C + R + c4 = 10\,c5 + A$
   $D = c5$
   $D \neq 0$
   No two letters have the same value, $A \neq C$, $A \neq D$, …, $R \neq S$.

2. We apply the unary constraints first, which give D = 1 and c5 = 1.

Then we alternate backtracking and constraint propagation. Since we cannot start with propagating constraints, we first pick the value of one letter, say R. Whenever we need to pick the value of another variable, we select the variable with the fewest remaining values.

So, we first try every possible value of R. All odd values immediately fail the first constraint (no value of S allows R to be odd). Both R = 2 and R = 4 also lead to failures later. R = 6 works and yields S = 3, c1 = 0, c2 = 0, E = 4, O = 2, A = 5, c3 = 0, G = 7, c4 = 0, N = 8, and C = 9.

IV.

1. Red can be eliminated from domains of B and C by forward checking. No further values can be eliminated, by either forward checking or AC3.
2. Both green and blue can be eliminated from domain of D because the only two possible consistent assignments for B and C are blue and green or green and blue; either way, D would be neighboring both a blue and a green country, so D can't be either. This is a ternary constraint, because we must consider B, C, and D at the same time. Forward checking and AC3 can only detect binary constraint violations.
3. Prune a value v_i from domain D_i if for every value v_j in D_j and v_k in D_k such that v_j and v_k are consistent with each other, v_i is inconsistent with either v_j or v_k. Do this for all i, j, k.