The original order of nested mappings works by bottoming out (reaching $k = 0$), and then working back up by first adding each possible row to each position that has already been deemed safe, then filtering out unsafe positions, and passing it upwards to the next column. The outer mapping is applied to the recursive call, meaning that `(queen-cols (- k 1))` is only computed once for each `k`.

Louis's interchange flatmaps all the safe positions to each row in the current column, so that new positions can be created in preparation for testing. The reason this is slower is because instead of calculating all the current safe positions once, it is calculated `board-size` (n) many times in the inner `map` call, since the outer `lambda` is applied to each row. This contrasts to the other ordering of the nested mappings, where `(queen-cols (- k 1))` is evaluated once and then the `lambda` is applied to each of its positions. This means for each `k`, the recursive call is computed n times, since the `board-size` is always constant. For each of those recursive calls, the recursive call is again repeated for each of the `k - 1` previous columns. The recursive calls form a tree recursive structure with a depth of n (k starts at n, so the first repeated call is $k - 1$ and k decrements to 0, $k - 1 - 0 = n$ because n starts at 1) and n many branches at each node. This compares to the $n + 1$ times `(queen-cols (- k 1))` is called in the original order.

Therefore, if Exercise 2.42 solves the puzzle in time T, Louis's method solves the puzzle in time $\left(\frac{n^n}{n+1}\right)(T)$.