META ALGORITHM DYNAMIC PROGRAMMING: turn a recursive algorithm into a dynamic programming algorithm

Review the existing recursive algorithm definition
Write: the name of the function
Review: What are the arguments to the recursive function? These will be the indexes into the cache.
Review: What is the return type of the function? This will be the type that is stored in the cache.
Think: Are all the arguments integers or are they of mixed type? If mixed, maybe a dictionary or hash table may be best. Will all possible solutions be generated, or only a few specific problems? If all, then an array may be best, if only a few, then maybe a dictionary or hash table should be used.
Write the declaration of the cache data structure. This will be defined once, as the first statement of the new DP function:

Think: How can the base cases in the recursive solution be stored in the cache? If the test checks that all of the arguments are a specific value, then this will become a single assignment into the cache. If the test checks only some of the arguments, or tests for ranges of argument values, then loops will be needed that enumerate all possible input argument values that will pass the test. For each problem instance generated, make an assignment into the cache, storing the value returned by the base case.
Code: Write the loops/assignments that fill in the cache.

Review: the recursive calls to understand how the main problem is made simpler and what smaller solutions are needed to compute the main solution.
Think: How do the function arguments change in the recursive calls? Do they always go smaller? What order should the DP loop through the values of the function parameters, such that the smaller problems are always computed before the larger problems?
Write: the code that scans through the cache in the correct order. Here the loops will iterate over the values to the function arguments from smallest value (excluding the base cases) to the largest values. If the function takes more than one argument, nested loops will be needed.

Review the code that calls the smaller problem instances recursively and computes the main solution.
Code: Rewrite this recursive code by replacing function calls with cache access and returns with cache assignments. This code will be the body of the code inside the nested for loops. Note, that the names of the variables in the recursive code will have to be changed to the iterative variables of the DP loops.

Code: add the return statement at the end that returns the solution to the original problem by indexing into the cache