META ALGORITHM TRACEBACK: extracting the objects that make up a solution from the filled cache

Most DP algorithms compute the solution value or truth, but not the individual components that make up the solution. For instance, in a knapsack, we want the actual items that make up the solution.

Think: What are the details of the problem we need, and how they are represented in the code?
Write: Identity the data structure <DS> to store the solution details. This could be a list of objects or values.

Think about the solution cache. Where is the top-level solution we were asked to solve? Study the solution construction code and identify where the sub-solutions are located in the array, relative to the solution.

Code: Write code that will traverse through the solution array backwards from the top-level solution until a base case is reached. The traceback routine will take the same arguments as the original recursive algorithm. These parameters are the index into the solution cache. Here is the framework algorithm:

```
traceBack(function parameters)
If base case then return //done
//recursive case
Inspect each subsolution in the cache that computed the solution at <function parameters> and use the solution construction code to identify the  subsolution that produced the answer. Store the choice that made the subsolution in <DS>
Call traceback(parameters of the subsolution)
```