Deciding on Sub-Problems for Dynamic Programming

I have used the technique of dynamic programming multiple times however today a friend asked me how I go about defining my sub-problems, I realized I had no way of providing an objective formal answer. How do you formally define a sub-problem for a problem that you would solve using dynamic programming?

algorithms dynamic-programming

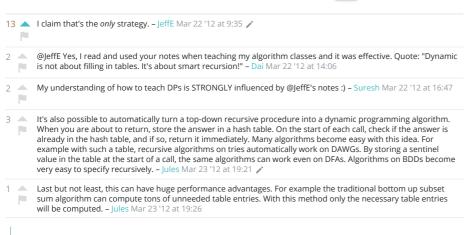




3 Answers

The principle of dynamic programming is to think top-down (i.e recursively) but solve bottom up. So a good strategy for designing a DP is to formulate the problem recursively and generate subproblems that way.





As @Suresh pointed out, once you know that your problem can be solved by DP (i.e. it exhibits optimal substructure and overlapping subproblems), you may think of a divide and conquer recursive solution.

Of course, divide and conquer will be highly ineffecient because every subproblem encountered in the associated recursion tree will be solved again even if it has already been found and solved. This is where DP differs: each time you encounter a subproblem, you solve it and store its solution in a table. Later, when you encounter again that subproblem, you access in $\mathcal{O}(1)$ time its solution instead of solving it again. Since the number of overlapping subproblems is typically bounded by a polynomial, and the time required to solve one subproblem is polynomial as well (otherwise DP can not provide a cost-efficient solution), you achieve in general a polynomial solution.

Therefore, thinking about a divide and conquer solution will provide you with the insight about what a subproblem may be for your particular problem.



can't be partially wrong: either is correct or it is wrong), you can simply show as a counterexample, a problem which does not exhibit both optimal substructure and overlapping subproblems, yet is amenable to a polynomial DP solution. But note that, in this context, that means a solution which is provably better than ordinary divide and conquer. – Massimo Cafaro Mar 22 '12 at 16:23

My experience is finding out a way to "cut down redundant enumerating with help of storing useful value already enumerated". By the way, Dynamic Programming is really popular in ICPC(International Collegiate Programming Contest. Anyone can have his own feeling about DP after practice several ICPC problems.

answered Mar 22 '12 at 9:32
Peng Zhang
133

5