# If you could rename dynamic programming…

If you could rename dynamic programming, what would you call it?

ds.algorithms soft-question terminology ho.history-overview

edited Mar 25 '11 at 10:11

community wiki
3 revs, 3 users 100%
Jack

**protected** by Artem Kaznatcheev ♦ Feb 6 '14 at 15:45

This question is protected to prevent "thanks!", "me too!", or spam answers by new users. To answer it, you must have earned at least 10 reputation on this site (the association bonus does not count).

1 I would say dynamic programming is dynamic programming. That's a separate concept from algorithms. If you would ask "algorithmic applications of dynamic programming", that would make more sense to me. – Yoshio Okamoto Mar 24 '11 at 23:17

1 Of course dp is dp, but programming and dynamic both mean something different today, so when I teach dynamic programming, I wish it had a different name. – Jack Mar 25 '11 at 0:01

4 Sorry, I wasn't clear enough. Are you interested in dynamic programming in general, including the use in control and policy planning, etc., and even stochastic dynamic programming, or just in dynamic programming as a method for algorithm design? The main audience here would know the latter only, but your question is quite general that would includes the former. – Yoshio Okamoto Mar 25 '11 at 1:07

1 Yoshio, I think you should just explain the more general concept resp the differences in an answer as it could enlighten many of us. – Raphael Mar 25 '11 at 1:24

1 @Raphael: Thank you for the suggestion, but I don't think that will be an answer. Probably, it's enough to include references to Wikipedia: en.wikipedia.org/wiki/Dynamic_programming and en.wikipedia.org/wiki/Bellman_equation. – Yoshio Okamoto Mar 25 '11 at 1:54

## 11 Answers

Richard Bellman's autobiography suggests that he chose the term “dynamic programming” to be intentionally distracting.

> The 1950s were not good years for mathematical research. We had a very interesting gentleman in Washington named Wilson. He was secretary of Defense, and he actually had a pathological fear and hatred of the word ‘research’. I'm not using the term lightly; I'm using it precisely. His face would suffuse, he would turn red, and he would get violent if people used the term ‘research’ in his presence. You can imagine how he felt, then, about the term ‘mathematical’. The RAND Corporation was employed by the Air Force, and the Air Force had Wilson as its boss, essentially. Hence, I felt I had to do something to shield Wilson and the Air Force from the fact that I was really doing mathematics inside the RAND Corporation.
>
> What title, what name, could I choose? In the first place I was interested in planning, in decision making, in thinking. But planning, is not a good word for various reasons. I decided therefore to use the word ‘programming’. I wanted to get across the idea that this was dynamic, this was multistage, this was time-varying—I thought, let’s kill two birds with one stone. Let’s take a word that has an absolutely precise meaning, namely ‘dynamic’, in the classical physical sense. It also has a very interesting property as an adjective, and that is it’s impossible to use the word ‘dynamic’ in a pejorative sense. Try thinking of some combination that will possibly give it a pejorative meaning. It’s impossible. Thus, I thought “dynamic programming’ was a good name. It was something not even a Congressman could object to. So I used it as an umbrella for my activities.

(As Russell and Norvig point out in their AI textbook, however, this story must be a creative embellishment of the truth. Bellman first used the phrase "dynamic programming" in 1952, and Charles Erwin Wilson did not become Secretary of Defense until 1953.)

Anyway, Bellman's original motivation suggests **multistage planning**, but at least for algorithmic purposes, I'd prefer something like **frugal bottom-up recursion**, only with fewer syllables.

answered Mar 25 '11 at 3:43

community wiki
Jeffε

9 Of course, what I'd *really* like to do is rename “Bellman's Equation”, or as computer scientists call it, “any recurrence whatsoever”. – Jeffε Mar 25 '11 at 3:45

1 your answer was used here: biostar.stackexchange.com/questions/17954 – Pierre Feb 28 '12 at 14:52

1 In case anyone wants to look up this gentleman, Wilson, he is Charles Erwin Wilson, US Secretary of Defense 1953-57. – David Richerby Feb 6 '14 at 9:14

Probably something that includes the words *table* and *fill*, as this is what happens.

answered Mar 24 '11 at 22:49 — community wiki
Raphael

7 Bah. Dynamic programming isn't about *tables*; it's about smart recursion. – Jeffε Mar 25 '11 at 3:13

3 I feel it's better to separate two aspects: "Extracting a recursive structure from the problem, and writing down as a recurrence" (modeling) and "Solving the obtained recurrence in a bottom-up way" (algorithms). I feel dynamic programming (in algorithms) refers to both, and this is also the case for linear programming, integer programming, semidefinite programming, etc. – Yoshio Okamoto Mar 26 '11 at 2:34

1 Tables are *sometimes* used. Dynamic programming over trees (for example: maximum independent set) doesn't normally use a table [=array] to memoize the recurrence; it uses a tree, or in some cases a tree of arrays, or an array of trees, or in some cases a Cartesian product of trees. Similarly for dynamic programming over dags or dynamic programming over tree decompositions for graphs of bounded treewidth. – Jeffε Mar 26 '11 at 3:15

1 JeffE, a name for a technique hardly ever covers all applications. Take "diagonalisation", for example; in advanced applications, there is no diagonale anywhere in sight (or at least not the exclusive area of action). But I am not too much in love with "table filling", anyway, while I think it could be reasonable name for beginners. – Raphael Mar 26 '11 at 10:17

3 My 2 cents on this topic. Dynamic programming has two distinct aspects to design algorithms. First is to understand the mechanics of dp as in: smart recursion plus memoization leading to a faster algorithm. The second aspect is how to **recognize** that a problem may admit a smart recursion and come up with it. Both are important to teach. For the latter, a common case is divide and conquer with limited interaction and in my view worthwhile to explicitly highlight. – Chandra Chekuri Feb 11 '14 at 15:50

|

**Memoization** is a fairly common variant.

answered Mar 24 '11 at 23:00 — community wiki
Suresh Venkat

8 Memoization is used, but does not characterize dp. – Jack Mar 25 '11 at 0:04

20 Exactly. Memoization is dynamic programming by accident. – Jeffε Mar 25 '11 at 3:12

@professor erickson - very well said. i cannot stop laughing. – Akash Kumar Mar 25 '11 at 11:51

7 Still, if we get to choose a new name for DP, then using memoization would be quite fitting for it. E.g. "edit distance can be computed in poly-time using memoization". – Noam Mar 26 '11 at 15:40

Dynamic programming is a special case of memoization, plus explicitly directing control flow instead of following the natural applicative evaluation order. It's a shame it's usually taught the way it is, as it puts too much emphasis on filling out a table, instead of the recursive specification. It comes across as magical. – Neil Toronto Feb 7 '14 at 3:42

This paper (paywalled doi) calls *problems* that can be attacked using DP "decomposable".

edited Feb 7 '14 at 3:07 — community wiki
2 revs, 2 users 67%
Yuval Filmus

After my recent lecture on dynamic programming in algorithm design I had asked students to suggest a new name for this technique. While I was amused by "Tough programming," I wanted something that might make the technique more memorable. After the discussion here, I may propose two names, one for top-down and one for bottom-up:
Multiway-Divide and Memoized-Conquer (aka Divide^M & Conquer^M), and
Merge all subproblems (aka Merge_all)

answered Mar 25 '11 at 22:09 — community wiki
Jack

1 I do not think that creating a strong association with usual divide & conquer (as in Mergesort) is a good idea. There, subproblems are solved independently and only two results are merged. In DP, the checked subproblems are not independent and all combinations are checked. (both in rough terms). Therefore I think a name should highlight the difference rather than creating a sense of similarity. – Raphael Mar 26 '11 at 10:15

@Raphael, I share the concern about the dangers of creating a strong association, but disagree with your statement of the differences. In DP it is crucial that the subproblems *are* "solved independently" even though solutions to subsubproblems are shared. I usually point that if some oracle told you the right division, it would be divide and conquer, but since I don't speak Greek (unlike my PhD advisor), I have to try all possible divisions. – Jack Mar 26 '11 at 12:28

2 Yes, subproblems are *conceptually* independent. However, I was relating to them using the same partial results, as you point out. This separates DP from D&C, since it is e.g. possible to parallelise the latter without computing subproblems multiple times (or communicating the results) as opposed to the former. Your analogy is nice but does not warrant the

name in this case as finding the correct divisions is an essential part of the problem. I suppose you could say that DP is a generalisation of D&C based on that reasoning, though. – Raphael Mar 26 '11 at 12:46

@Raphael, sorry to be pedantic, but since this is a question about teaching, I want the notion of subproblem independence to be clear, and just saying "conceptually independent" is not precise. When you try a division, the subproblems you produce may not have dependencies. Your other comments focus on the steps of DP algorithms; I prefer to focus first on precisely defining the problem to be solved. My favorite examples: min weight triangulation and min convex decomposition of simple polygons (cs.unc.edu/~snoeyink/demos/convdecomp/MWTDemo.html) – Jack Mar 26 '11 at 21:36

It is ok to be pendantic. But consider the didactic import of the choice of words: you tell the students that subproblems are independent and then give them an algorithm that does *not* separate their computation, but in fact relies heavily on "interleaved" computation. I think that can be very confusing. Therefore, you really have to separate conceptual/mathematical/optimisation/... and computational independence at some point. – Raphael Mar 26 '11 at 22:38

|

Recursive view or Recursive horizon

answered Mar 25 '11 at 23:13 — community wiki Mark

Lazy horizon? You delay calculating a lot of results until they are needed. DP need not be recursive. – Chad Brewbaker Feb 6 '14 at 15:53

To go with *divide-and-conquer*, I would say *splice-and-combine.*

I usually use both words, *splice* and *combine* while teaching/explaining DP; but not used splice-and-combine explicitly. Sometimes I have used *overlapping-divide-and-conquer* to contrast the two paradigms.

answered Mar 26 '11 at 3:56 — community wiki V Vinay

There are two important aspects of DP: (1) defining the subproblems (i.e., setting up a "table", which could be a multidimensional array indexed maybe by integers, vertices, subsets of vertices etc.) and (2) recursively solving the subproblems. I propose "tabular/tabulated recursion" as a name that refers to both aspects.

answered Mar 27 '11 at 7:45 — community wiki Daniel Marx

6 tabular recursion does have a nice feel to it. – Suresh Venkat Mar 27 '11 at 7:47

I discussed this with some colleagues recently, and after a heated discussion we came up with **tabular call caching**.

answered Apr 8 '11 at 18:29 — community wiki Kevin Wortman

3 that sounds like something you do at an outsourced call center ;) – Suresh Venkat Apr 8 '11 at 21:49

I'd suggest name **Inductive Programming** -- as a kind of smth of bridge-like from our times to old good times of Euler, Kepler et al. Or maybe even **Reverse Inductive Programming**. And yes, for me DP is strongly associated with the Induction, in old good sense of the notion. Memoization, cacheing, tables etc are just elements of technique, not of the core of the approach to crack things.

edited Apr 12 '11 at 14:18 — community wiki 2 revs trg787

my main problem with DP is the P. so I'm not a fan of this idea.. – Suresh Venkat Apr 12 '11 at 17:07

I'd name it "Forward recursion with memoization".

answered Feb 6 '14 at 8:23 — community wiki Benjamin