Question:

What is the difference between dynamic programming and the ...and-conquer family?

* Why is dynamic programming a kind of algorithm apart?
* Which one of the ...-and-conquer family is closer to dynamic programming?
* Is there some overlap between dynamic programming and any of the the ...-and-conquer family algorithms?

Answers:

The difference between dynamic programming and the and conquer family is commonality. Dynamic algorithms operate by using saved, or cached, information that has already been processed. This is where commonality comes in, as the subproblems overlap; the cached information is reused to make operations more efficient and avoid redundant processing. The and conquer family, on the other hand, does not work for problems that have overlapping sub-problems. Once the problem in question has been split, the sub-problems share no commonalities, so they are solved independently, which is also efficient in this case.

Under the hood, dynamic programming can be considered an optimization strategy that is applied to situations with overlapping subproblems. There is also some minor overlap with the and conquer family if we are using the top down approach. The top down approach is recursive, similar to the divide-and-conquer family, but adds a layer of optimization by caching relevant computations that have already been processed. Simply saying that it solves recursive inefficiencies by adding the ability to cache information to eliminate redundant computations, as mentioned above.

We know that the main difference between the and conquer family and DP is memoization. But if we forget about memorization for a bit, we notice that the divide and conquer family is similar to dynamic programming because it also breaks problems down into subproblems.