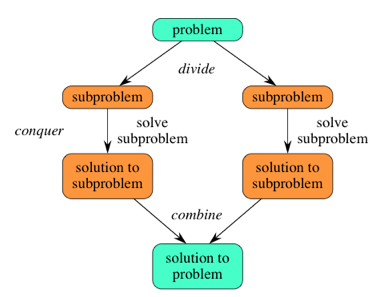
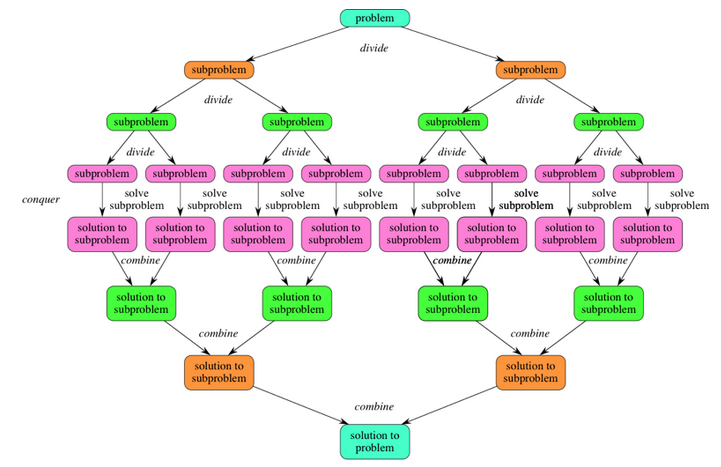
Divide and conquer

A divide and conquer algorithm works by recursively breaking down a problem into two or more sub-problems of the same or related type, until these become simple enough to be solved directly. The solutions to the sub-problems are then combined to give a solution to the original problem. It has three parts:

1. Divide the problem into a number of sub-problems that are smaller instances of the same problem.
2. Conquer the sub-problems by solving them recursively. If they are small enough, solve the sub-problems as base cases.
3. Combine the solutions to the sub-problems into the solution for the original problem.





Dynamic program

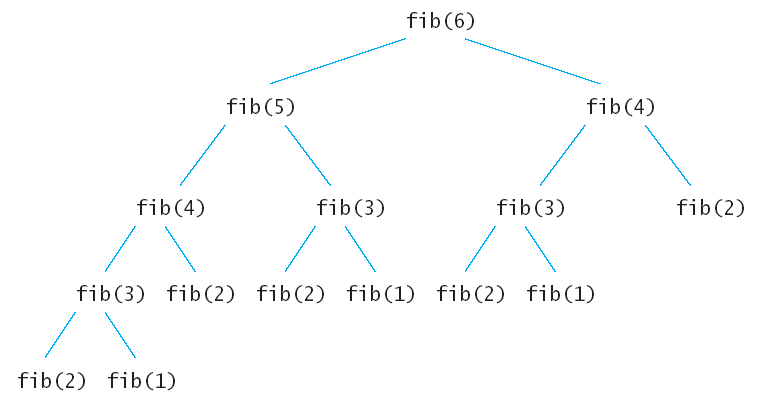
A problem can be broken down into smaller sub-problems, and these smaller sub-problems can still be broken into smaller ones, and there are some over-lapping sub-problems. Results of these smaller sub-problems are remembered and used for similar or overlapping sub-problems.

Dynamic programming is used where we have problems, which can be divided into smaller sub-problems, so that their results can be re-used.

The core idea of Dynamic programming is to avoid repeated work by remembering partial results.

In contrast to greedy algorithms, where local optimization is addressed, dynamic algorithms are motivated for an overall optimization of the problem.

In contrast to divide and conquer algorithms, where solutions are combined to achieve an overall solution, dynamic algorithms use the output of a smaller sub-problem and then try to optimize a bigger sub-problem. Dynamic algorithms use memorization to remember the output of already solved sub-problems. (many values are recalculated in the recursive way)



Every Dynamic Programming problem has a schema to be followed:

1. Show that the problem can be broken down into optimal sub-problems.
2. Recursively define the value of the solution by expressing it in terms of optimal solutions for smaller sub-problems.
3. Compute the value of the optimal solution in bottom-up fashion.