

Dynamic Programming is an algorithmic technique, developed by Richard Bellman. It solves problems by recursively breaking them down into simpler subproblems, and storing the solutions to each of these sub-problems in an array (or similar data structure) so each sub-problem is only calculated once.

Observe the following equation as an example:

$$5 + 4 + 3 + 2$$

It can be broken down into:

$$5 + 4 \text{ and } 3 + 2$$

When we solve those two smaller equations, it becomes easier to resolve the initial equation:

$$5 + 4 = 9$$

$$3 + 2 = 5$$

$$9 + 5 = 14$$

This simple example demonstrates the principles of Dynamic Programming. The importance of this technique becomes clearer as we add more and more numbers to our equation example:

$$10 + 10 + 5 + 4 + 3 + 2 + 10 + 10$$

On the first occurrence of '10 + 10', we solve it: '10 + 10 = 20'. However, when we come across its second occurrence, we do not work it out again, because we already solved it earlier and we know its solution is 20.

The same thought process occurs in Dynamic Programming. It stores the solution to the problem so we do not need to recalculate it every time we encounter it. This act is called Memoisation.